

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:

The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

University Libraries
Carnegie Mellon University
Pittsburgh PA 15213-3890

**An Intelligent Predictive Control Approach
to the High-Speed Cross-Country
Autonomous Navigation Problem**

Alonzo Kelly

CMU-RI-TR-95-33^

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213

September 29, 1995

© 1995 by Alonzo Kelly. All rights reserved.

This research was sponsored by ARPA under contracts "Perception for Outdoor Navigation" (contract number DACA 76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-RO59, monitored by TACOM).



Robotics

Thesis

An Intelligent Predictive Control Approach to the High-Speed Cross-Country Autonomous Navigation Problem

Alonzo Kelly

Submitted in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in the field of Robotics

ACCEPTED:

Anthony Stentz
Anthony Stentz Thesis Committee Chair

10-31-95
Date

Matthew T. Mason
Matthew T. Mason Program Chair

November 1, 1995
Date

Raj Reddy
Raj Reddy Dean

Nov 3, 1995
Date

APPROVED:

Paul Christiano
Paul Christiano Provost

7 November 1995
Date

Abstract

Autonomous robot vehicles promise many ultimate civilian, military, and space applications. Off-road autonomous vehicles must engage the world exactly as they find it without relying on having it engineered to suit them. For this reason, off-road autonomous navigation is one of the most difficult automation challenges. Previous work in the area has been disappointing from the perspective of the speeds attained, and the inability of systems to travel long distances autonomously. Indeed, no system has travelled an autonomous mile or exceeded 3 m/s speeds. To date, no off-road system has approached the capabilities needed to address real applications.

This thesis examines and proposes a solution to the problem of high speed autonomous navigation of outdoor vehicles. As a systems-level effort, aspects of perception, path planning, position estimation, and to a lesser extent, strategic planning and motion control are considered. The emphasis of the work has been to assess the fundamental requirements of the problem, and to validate the conclusions of this assessment through the demonstration of an improved ability to achieve a real cross-country mission on several vehicle testbeds.

Results indicate that cross-country navigation systems of unprecedented capability are possible if they are designed to optimally utilize limited computing resources. A system of unprecedented performance has been constructed and extensively tested.

The essential argument of the thesis is one of architecture. An intermediate intelligent predictive control layer is introduced between the typical high-level strategic or artificial intelligence layer and the typical low-level servo control layer. This new layer, the tactical layer, incorporates some deliberation, and some environmental mapping as do deliberative AI planners, yet it also emphasizes the real-time aspects of the problem as do minimalist reactive architectures.

The contribution of the work is a codified systems theory that permits future design efforts to benefit from the experience and a fieldworthy prototype system that provides a baseline capability for continued research. Specific results include an analysis of the complexity of range image perception for autonomous vehicles and an associated computational image stabilization algorithm which permits highest vehicle speeds.

The problem of local autonomous mobility has been formulated entirely in an optimal control context. In this context, the concepts of actuation space and hazard space replace the configuration space that is more typical of AI planners. The resulting high fidelity models stabilize coordinated control of a high speed vehicle for both obstacle avoidance and goal seeking purposes.

List of Figures

	page
Figure 1	Response Ratio.....18
Figure 2	Throughput Ratio.....20
Figure 3	Acuity Ratios.....22
Figure 4	Fidelity Ratios.....24
Figure 5	Normalized Vehicle.....26
Figure 6	System Control Loops.....29
Figure 7	Empirical Braking Distance.....32
Figure 8	Stopping Distance.....33
Figure 9	Steering Limits.....37
Figure 10	Stopping Region.....38
Figure 11	Impulse Turning Trajectories.....39
Figure 12	Impulse Turning Distance.....40
Figure 13	Reverse Turning Trajectories.....41
Figure 14	Lookahead Zones.....45
Figure 15	Planner Lookahead.....46
Figure 16	Vertical Field of View.....48
Figure 17	Minimum Range.....50
Figure 18	Maximum Range.....51
Figure 19	Horizontal Field of View.....53
Figure 20	Tunnel Vision Problem.....55
Figure 21	Vertical Field of View.....56
Figure 22	Occlusions.....56
Figure 23	Nondimensional Vertical Field of View.....57
Figure 24	Sweep Rate.....58
Figure 25	Stabilization Problem.....59
Figure 26	Hill Occlusion.....61
Figure 27	Hole Occlusion.....62
Figure 28	Lateral Occlusion Problem.....63
Figure 29	Practical Sampling.....67
Figure 30	Perception Ratio.....68
Figure 31	Differential Imaging Kinematics.....70
Figure 32	Small Incidence Angle Assumption.....76
Figure 33	Correspondence Problem.....77
Figure 34	Stereo Range Resolution.....81
Figure 35	Effect of Integral Disparity.....83
Figure 36	Subpixel Disparity Estimation.....84
Figure 37	Geometric Decorrelation.....86
Figure 38	Typical Scanning Pattern.....97
Figure 39	Typical Spot Pattern.....97
Figure 40	Scanning Density Expression.....98
Figure 41	Scanning Density.....99
Figure 42	Imaging Density.....100

Figure 43	Geometric Efficiency.....	101
Figure 44	Latency Problem.....	104
Figure 45	Response Regimes.....	105
Figure 46	Transience in the Reverse Turn.....	107
Figure 47	Transient Steering Response.....	108
Figure 48	Bicycle Model.....	111
Figure 49	Constant Speed Reverse Turn.....	113
Figure 50	Constant Speed Partial Reverse Turn.....	114
Figure 51	Ackerman Steer Configuration Space.....	115
Figure 52	Feedforward.....	120
Figure 53	Throughput Problem.....	123
Figure 54	Throughput for Constant Flux, Constant Scan.....	127
Figure 55	Throughput for Adaptive Sweep, Constant Scan.....	130
Figure 56	Throughput for Adaptive Sweep, Adaptive Scan.....	132
Figure 57	Area Consumption.....	133
Figure 58	Throughput for Adaptive Sweep, Uniform Scan.....	134
Figure 59	Throughput for All Algorithms.....	135
Figure 60	Computational Spiral Effect.....	137
Figure 61	Standard Model.....	155
Figure 62	System Data Flow.....	165
Figure 63	Architecture.....	167
Figure 64	I/O Filter.....	169
Figure 65	Target Vehicles.....	173
Figure 66	Computational Image Stabilization.....	184
Figure 67	Range Window Computation.....	187
Figure 68	Adaptive Sweep Implementation.....	188
Figure 69	Adaptive Sweep/Scan Implementation.....	189
Figure 70	Stereo Triangulation.....	190
Figure 71	Essential Difficulty.....	191
Figure 72	Spurious Disparities.....	192
Figure 73	Adaptive Scan in Stereo Vision.....	197
Figure 74	Wrappable Map Indexing.....	199
Figure 75	Wrappable Terrain Map.....	200
Figure 76	Motion Distortion.....	201
Figure 77	Coordinate Transformation Data Flow.....	203
Figure 78	Map Management Data Flow.....	205
Figure 79	Stereo Vision Data Flow.....	207
Figure 80	Adaptive Rangefinder Perception.....	212
Figure 81	Adaptive Horizontal Baseline Stereo.....	214
Figure 82	Vertical Baseline Stereo.....	215
Figure 83	Adaptive Vertical Baseline Stereo.....	216
Figure 84	Hierarchical Planning with Arbitration.....	225
Figure 85	Local Navigation as an Optimal Control Problem.....	226
Figure 86	Multivariable Linear System Block Diagram.....	227
Figure 87	State Space Simulator.....	229
Figure 88	Optimal Control Through Actuation Space Search.....	230

Figure 89	Forward Modelling in Local Navigation.....	231
Figure 90	Basic Pure Pursuit.....	233
Figure 91	Feedforward Simulator.....	235
Figure 92	Reference Points.....	236
Figure 93	Propulsion Plant Model.....	238
Figure 94	Propulsion Servo.....	238
Figure 95	Steering Servo.....	239
Figure 96	Hazard Signals.....	242
Figure 97	Wheel Collision.....	244
Figure 98	Body Collision.....	244
Figure 99	Static Stability.....	245
Figure 100	Adaptive Pure Pursuit.....	247
Figure 101	Feedforward Pure Pursuit.....	248
Figure 102	Latency Effects.....	251
Figure 103	Head Servo.....	253
Figure 104	Merit Servo.....	255
Figure 105	Controller Data Flow.....	256
Figure 106	Command Generator Data Flow.....	257
Figure 107	Tactical Controller Data Flow.....	257
Figure 108	Global Path Tracker Data Flow.....	259
Figure 109	Arbiter Data Flow.....	260
Figure 110	System Model Data Flow.....	261
Figure 111	FIFO Queue Daemons Data Flow.....	262
Figure 112	Goal Arbitration.....	264
Figure 113	Obstacle AvoidanceTest Run.....	265
Figure 114	Performance of Feedforward Pure Pursuit.....	266
Figure 115	Performance of Speed Planning.....	267
Figure 116	State Space Kalman Filter.....	276
Figure 117	Position Estimator Data Flow.....	314
Figure 118	System Model Data Flow.....	315
Figure 119	Sensor Model Data Flow.....	316
Figure 120	Kalman Filter Data Flow.....	317
Figure 121	Kalman Filter Test Results.....	325
Figure 122	Crosstrack Aiding Problem.....	327
Figure 123	Terrain Map.....	335
Figure 124	State Space Vehicle Model.....	336
Figure 125	Hazard Detection and Avoidance.....	337
Figure 126	Adaptive Perception.....	338
Figure 127	Azimuth Scanner.....	392
Figure 128	Polygonal Reflection.....	393
Figure 129	Nodding Reflection.....	393
Figure 130	Azimuth Kinematics.....	394
Figure 131	Azimuth Range Image.....	398
Figure 132	Azimuth Field of View.....	399
Figure 133	Azimuth Scanning Pattern.....	400
Figure 134	Azimuth Spot Pattern.....	401

Figure 135	Elevation Scanner.....	402
Figure 136	Nodding Reflection.....	403
Figure 137	Polygonal Reflection.....	403
Figure 138	Elevation Kinematics.....	404
Figure 139	Elevation Range Image.....	408
Figure 140	Elevation Field of View.....	409
Figure 141	Elevation Scanning Pattern.....	410
Figure 142	Elevation Spot Pattern.....	411
Figure 143	Perspective Homogeneous Transform.....	412
Figure 144	Stereo Triangulation.....	413
Figure 145	Perspective Kinematics.....	414
Figure 146	Perspective Range Image.....	419
Figure 147	Disparity Image.....	419
Figure 148	Disparity Curve of a Column.....	420
Figure 149	Perspective Field of View.....	422
Figure 150	Perspective Scanning Pattern.....	423
Figure 151	Perspective Spot Pattern.....	424

List of Tables

	page
Table 1: Braking Distance.....	32
Table 2: Throughput Estimates.....	135
Table 3: Elementary Nondimensionals.....	140
Table 4: Configuration Nondimensionals.....	142
Table 5: Response Nondimensionals.....	143
Table 6: Throughput Nondimensionals.....	146
Table 7: Acuity Nondimensionals.....	148
Table 8: Fidelity Nondimensionals.....	150
Table 9: ERIM Range Sensor Parameters.....	175
Table 10: Configuration Parameters.....	175
Table 11: Run-Time (SPARC 20).....	176
Table 12: ERIM Range Sensor Parameters.....	209
Table 13: Camera Parameters (Vertical Baseline Case).....	209
Table 14: Configuration Parameters.....	210
Table 15: Adaptive Perception Performance.....	212
Table 16: Nonadaptive Perception Performance.....	213
Table 19: 3-Link Planar Manipulator.....	364
Table 20: Frame Parameters for Wheeled Vehicles.....	380

List of Symbols

Lowercase Alphabetics

a acceleration
a_{max} maximum acceleration
b stereo baseline
b/h normalized baseline
B, B_d (dynamic) braking coefficient
b_k kinematic braking coefficient
c undercarriage clearance
c normalized undercarriage clearance
d disparity
f frequency, focal length
 $\dot{\theta}^{att}$ attitude measurement rate
f_{cells} map cell throughput
f_{cpu} CPU speed
f_{comm} communications bandwidth
f_{heading} heading measurement rate
f_{images} frame rate
f_{pixels} sensor throughput
g acceleration due to gravity
h sensor height
h correlation window halfheight
h (x) measurement function
h perception ratio
i x-axis unit vector
j y-axis unit vector
I [i, j] left image intensity
n/2 oversampling factor
p probability density
r wheel radius
r normalized wheel radius
r position vector
r [i, j] right image intensity
s arc length, distance travelled
s_B stopping distance

s_T turning stopping distance
s_{IT} impulse turning distance
s_{error} kinematic steering model error
t time
t_k, t (kinematic) turning coefficient
t_d dynamic turning coefficient
t_t transient turning coefficient
u command vector
u_d disturbance input vector
u image plane coordinate
v image plane coordinate
w correlation window halfwidth
x state vector
x_i (t) candidate response trajectory
x_{goal} (t) goal response trajectory
x crossrange coordinate
XQ initial crossrange coordinate
X_j left image plane x coordinate
x_r right image plane x coordinate
y output vector, hazard vector
y downrange coordinate
y₀ initial downrange coordinate
y_j left image plane y coordinate
y_r right image plane y coordinate
y_{command}, y_{cmd} command signal
y_{error} error signal
y_{response}, y_{res} response signal
z vertical coordinate
z_Q initial vertical coordinate
z_f front elevation
z_r rear elevation

List of Symbols

Uppercase Alphabetics

A system dynamics matrix
 A_L longitudinal aspect ratio
 A_r wheel fraction
 A_c undercarriage tangent
 C covariance matrix
 C_{err} geometric decorrelation
 $C[d]$ correlation curve
 $C_I[i, j, d]$ Intensity correlation score table
 $C_s[i, j, d]$ Sign correlation score table
 $D_a[i, j, d]$ SAD score table
 $D_s[i, j, d]$ SSD score table
 \hat{F} Force
 F system dynamics matrix
 H measurment matrix
 H_{enc} encoder measurement matrix
 H_{dop} Doppler measurement matrix
 H_{com} compass measurement matrix
 H_ρ AHRS measurement matrix
 J Jacobian matrix
 K_p proportional gain
 K_M mirror gain
 K Kalman gain matrix
 $L[i, j]$ normalized left image intensity
 L vehicle wheelbase
 \bar{L} normalized wheelbase
 M vehicle mass
 M merit signal
 N statistical normalization operator
 \hat{P} propulsive force
 $R[i, j]$ normalized right image intensity
 R range
 R rotation matrix
 R_{max} maximum range

R_{min} minimum range
 \bar{R} range ratio
 $R_{xx}(t)$ autocorrelation function
 $R_{xy}(t_1, t_2)$ cross-correlation function
 R_L planner lookahead distance
 S scatter matrix
 $S_{xx}(j\omega)$..power spectral density
 $S_{xy}(j\omega)$..cross power spectral density
 T time
 T homogeneous transform matrix
 T_{act} actuator delay
 T_{cont} control reaction time
 T_{cyc} software cycle time
 T_{dyn} dynamics reaction time
 T_{hw} hardware reaction time
 T_{images} ...frame period
 T_{lat} frame buffer latency
 T_{look} temporal planning horizon
 T_{perc} perception reaction time
 T_{plan} planning reaction time
 T_{react} system reaction time
 T_{sens} sensing reaction time
 T_{sw} processing reaction time
 T_{turn} transient turning reaction time
 T_{veh} vehicle reaction time
 U correlation window width
 V vehicle speed
 V_{pixel} pixel footprint speed
 V_{enc} encoder speed measurement
 V_{dop} Doppler speed measurement
 V correlation window height
 W vehicle width
 \vec{W} weight
 X_L, Y_L left camera relative coordinate
 X_R, Y_R right camera relative coordinate

List of Symbols

Ycommon camera relative coordinate

Lowercase Greek Alphabetics

αsteer angle

$\dot{\alpha}$steer angle rate

$\dot{\alpha}_{\max}$maximum steer angle rate

$\dot{\alpha}_z$angular velocity (z component)

δmap resolution

δ_nnormalized map resolution

δnormalized disparity

δ_udisparity gradient - u component

δ_vdisparity gradient - v component

$\delta(x)$Dirac delta generalized function

Aobstacle spacing

$T|_s$perceptual software efficiency

$T|_G$geometric efficiency

Kcurvature

K_0initial curvature

$[L$coefficient of friction

JLstatistical mean

vcoefficient of lateral acceleration

$\backslash/$yaw, pixel azimuth, vehicle yaw

\backslash/Qinitial yaw

$\backslash/_{\text{comp}}$compass heading measurement

\backslash/jvehicle yaw rate

$V|_L$planner lookahead angle

$A\backslash|_L$incremental lookahead angle

pradius of curvature

p_Kkinematic steering limit

p_Ddynamic steering limit

p_Ran unnamed nondimensional

p_{min}minimum radius of curvature

p_{cyc}throughput ratio

p_{dx}fidelity ratio

$p_{\alpha y}$minimum acuity ratio

p_{dz}maximum acuity ratio

\bar{p}_{react}reactive ratio

p_tturning fidelity ratio

p_{ahrs}AHRS attitude measurement

pattitude vector

Garea density

astatistical standard deviation

G_ffront elevation uncertainty

G_rrear elevation uncertainty

a_xx coordinate state uncertainty

a_yy coordinate state uncertainty

G_zz coordinate state uncertainty

G_gpitch state uncertainty

G_{ϕ}roll state uncertainty

G_{ψ}yaw (heading) state uncertainty

G_yspeed state uncertainty

$G_{\dot{\psi}}$yawrate state uncertainty

G_{enc}encoder measurement uncertainty

G_{dop}Doppler measurement uncertainty

G_{comp}compass measurement uncertainty

G_{pitch}pitch measurement uncertainty

a_{roll}roll measurement uncertainty

G_{yaw}yaw measurement uncertainty

G_{steer}steering measurement uncertainty

G_ccommunications load

G_jimaging density

G_sscanning density

G_pprocessor load

Ttime constant

tnormalized time constant

T_Bbraking reaction time

T_jturning reaction time

θpitch, beam elevation, terrain slope

θ_{\max}maximum beam depression

e_{\max}maximum allowable body pitch

List of Symbols

θ_{min} minimum beam depression
 $\dot{\theta}$ vertical sweep rate
 ϕ roll
 $\dot{\omega}$ angular velocity

Uppercase Greek Alphabetics

Ψ sensor flux
 Φ transition matrix
 Γ angular velocity transform matrix
 Ξ observability matrix

Increments and Differentials

$dx, \Delta x$ crossrange incremental distance
 $dy, \Delta y$ downrange incremental distance
 $dz, \Delta z$ vertical incremental distance
 Δd disparity resolution
 Δf_{cpu} change in CPU speed
 Δh height increment, resolution
 Δt time delay
 $\Delta R, \Delta R_L$..incremental lookahead distance
 $\Delta \bar{R}$ normalized range difference
 $\Delta \bar{R}_L$ normalized increm. lookahead dist.
 ΔT_{react} change in system reaction time
 $\Delta \alpha$ steer angle increment
 $\Delta \theta$ pitch error

Functions and Operators

$E(x)$ expected value
 $N(\mu, \sigma)$..Gaussian distribution
 $x \oplus y$ logical equivalence
 $x \otimes y$ outer product
 $L(f(x))$.Functional over f(x)
 $\vec{\nabla}$ gradient operator
 ∇^2 Laplacian operator
 $rem\left(\frac{x}{y}\right)$...remainder
 $\lfloor x \rfloor$ floor function, or greatest integer
 $sgn(x)$ sign of x
 $signof(x)$ sign of x

List of Acronyms

AHRS	Attitude and Heading Reference System
AI	Artificial Intelligence
ALV	Autonomous Land Vehicle
• ALVINN	Autonomous Land Vehicle in a Neural Net
AM	Amplitude Modulated
ARPA	Advanced Research Projects Agency
A-Space	Actuation Space
CCD	Charge Coupled Device
CEM	Cartesian Elevation Map
CMU	Carnegie Mellon University
C-Space	Configuration Space
DH	Denavitt-Hartenberg
DR	Dead Reckoning
EKF	Extended Kalman Filter
ERIM	Environmental Research Institute of Michigan (name of a laser rangefinder)
FIFO	First In, First Out (a type of queue data structure)
FM	Frequency Modulated
GDOP	Geometric Dilution of Precision
GPS	Global Positioning System
HFOV	Horizontal Field of View
HIFOV	Horizontal Instantaneous Field of View
HMMWV	Highly Mobile Multi-Wheeled Vehicle
H-Space	Hazard Space
IFOV	Instantaneous Field of View
INS	Inertial Navigation System
JPL	Jet Propulsion Laboratory
MIMO	Multi-Input, Multi-Output

List of Acronyms (continued)

NAVLAB	Navigation Laboratory
RANGER	Real Time Autonomous Navigator with a Geometric Engine
RATLER	Robotic All Terrain Lunar Exploration Rover
RPY	Roll-Pitch-Yaw
RISC	Reduced Instruction Set Computer
SPARC	Scalable Processor Architecture
SSD	Sum of Squared Differences
TOF	Time of Flight
UGV	Unmanned Ground Vehicle
VFOV	Vertical Field of View
VIFOV	Vertical Instantaneous Field of View

Table of Contents

PART I: Introduction		
Section 1:	Problem Statement.....	1
Section 2:	Requirements.....	2
2.1	Continuous, High Speed.....	2
2.2	Rough Terrain.....	2
2.3	Ackerman Steer Vehicle.....	2
2.4	Robustness.....	2
Section 3:	Previous Work.....	3
3.1	Differences from Important Historical Systems.....	3
3.2	Summary of Previous Work.....	6
Section 4:	Approach.....	7
4.1	Principles.....	7
4.2	Perspective.....	7
Section 5:	Acknowledgments.....	8
5.1	Analysis.....	8
5.2	Perception.....	9
5.3	Planning/Control.....	9
5.4	Position Estimation.....	9
Section 6:	Notational Conventions.....	10
Section 7:	Thesis Overview.....	11
7.1	Analysis.....	11
7.2	Mobility.....	11
7.3	Perception.....	12
7.4	Planning.....	12
7.5	Position Estimation.....	12
7.6	Conclusions.....	13
Section 8:	How to Read This Thesis.....	14
8.1	Summary Level.....	14
8.2	Intermediate Level.....	14
8.3	Detail Level.....	14

PART II: Analysis

Chapter 1:	Elements.....	16
Section 1:	Guaranteed Safety Policy.....	16
1.1	Four Dimensions of Safety.....	16
1.2	Safety.....	17
1.3	Response.....	18
1.4	Throughput.....	20
1.5	Acuity.....	22
1.6	Fidelity.....	24
Section 2:	Configuration - Normalized Vehicle.....	26
Chapter 2:	Response.....	27
Section 1:	Reaction Time.....	27
1.1	Reaction Time Components.....	27
1.2	Nondimensional Response.....	28
1.3	System Reaction Time.....	29
1.4	Braking Reaction Time.....	30
1.5	Turning Reaction Time.....	30
1.6	Reaction Distance.....	30
Section 2:	Maneuverability.....	31
2.1	Braking.....	31
2.2	Empirical Braking.....	32
2.3	Panic Stop Maneuver.....	33
2.4	Nondimensional Braking.....	34
2.5	Braking Regimes.....	35
2.6	Turning.....	36
2.7	Steering Limits.....	37
2.8	Turning Stop Maneuver.....	38
2.9	Impulse Turn Maneuver.....	39
2.10	Impulse Turning Distance.....	40
2.11	Reverse Turn.....	41
2.12	Stability Problem.....	41
2.13	Nondimensional Turning.....	42
2.14	Turning Regimes.....	42
2.15	Turning Stop Maneuver.....	43
2.16	Impulse Turn Maneuver.....	44
2.17	Impulse Turning Regimes.....	44
Section 3:	Lookahead.....	45
3.1	Adaptive Regard.....	45
3.2	Pointing Rules.....	46
3.3	Adaptive Lookahead.....	47

3.4	Nondimensional Lookahead.....	47
3.5	Nondimensional Pointing Rules.....	48
Chapter 3:	Throughput.....	49
Section 1:	Depth of Field.....	49
1.1	Minimum Sensor Range.....	50
1.2	Maximum Sensor Range.....	51
1.3	Nondimensional Maximum and Minimum Range.....	52
1.4	Myopia Problem.....	52
Section 2:	Horizontal Field of View (HFOV).....	53
2.1	Nondimensional Horizontal Field of View.....	54
2.2	Tunnel Vision Problem.....	55
Section 3:	Vertical Field of View (VFOV).....	56
3.1	Nondimensional Vertical Field of View.....	57
Section 4:	Sweep Rate.....	58
4.1	Stabilization Problem.....	59
4.2	Nondimensional Sweep Rate.....	60
4.3	Adaptive Sweep.....	60
Section 5:	Occlusion.....	61
5.1	Hill Occlusion.....	61
5.2	Hole Occlusion.....	62
5.3	Occlusion Problem and Unknown Hazard Assumption.....	63
5.4	Lateral Occlusion.....	63
Section 6:	Perceptual Bandwidth.....	64
6.1	Sensor Flux.....	64
6.2	Sensor Throughput.....	64
6.3	Sweep Rate.....	64
6.4	Processor Load.....	64
6.5	Perceptual Software Efficiency.....	65
6.6	Computational Bandwidth.....	65
6.7	Communications Bandwidth.....	65
Chapter 4:	Acuity.....	67
Section 1:	Range Image Acuity.....	67
1.1	Sampling Theorem.....	67
1.2	Terrain Smoothness Assumption.....	68
1.3	Impact of Imaging Geometry on Acuity.....	68
1.4	Nomenclature.....	69
1.5	Sampling Problem.....	69
1.6	Differential Imaging Kinematics.....	70
1.7	Pixel Footprint Area and Density Nonuniformity.....	71
1.8	Pixel Footprint Aspect Ratio.....	72
1.9	Minimum Sensor Acuity in Image Space.....	73

1.10	Maximum Sensor Acuity in Image Space.....	74
1.11	Kinematic Maximum Range and the Myopia Problem.....	75
1.12	Maximum Angular Resolution.....	75
1.13	Acuity Problem.....	75
1.14	Small Incidence Angle Assumption.....	76
Section 2:	Brief Introduction to Stereo Vision.....	77
2.1	Correspondence Problem.....	77
2.2	Disparity and Epipolar Geometry.....	77
2.3	Area-Based Stereo.....	78
2.4	Fundamental Inconsistency of Area-Based Stereo.....	80
Section 3:	Range Acuity in Stereo Vision.....	81
3.1	Range Resolution.....	81
3.2	Need for Subpixel Disparity Estimation.....	82
3.3	Subpixel Disparity Estimation.....	84
Section 4:	Angular Acuity in Stereo Vision.....	85
4.1	Disparity Gradient.....	85
4.2	Geometric Decorrelation.....	86
4.3	Halo Effect.....	87
Section 5:	Perceptual Uncertainty.....	90
5.1	Motivation.....	90
5.2	Mechanism.....	90
5.3	Obstacle Height.....	91
5.4	Terrain Gradient.....	92
Section 6:	Positioning Bandwidth.....	93
6.1	Heading and Positioning Bandwidth.....	93
6.2	Attitude and Positioning Bandwidth.....	94
6.3	Motion Distortion Problem.....	95
6.4	Need for Simultaneous Digitization in Stereo Vision.....	95
Section 7:	Geometric Efficiency.....	97
7.1	Imaging Geometry.....	97
7.2	Scanning Density.....	98
7.3	Imaging Density.....	100
7.4	Geometric Efficiency.....	101
7.5	Adaptive Scan.....	102
7.6	Acuity Nondimensionals.....	102
Chapter 5:	Fidelity.....	103
Section 1:	Modelling Dynamics and Delays.....	103
1.1	Latency Problem.....	104
1.2	Minimum Significant Delay.....	104
1.3	Dynamic Systems.....	105
1.4	Response Regimes.....	105
1.5	Characteristic Times and Low Latency Assumption.....	106

1.6	Normalized Time Constant.....	106
1.7	Low Dynamics Assumption.....	106
1.8	Transience in Turning.....	107
1.9	Heading Response.....	108
1.10	Nondimensional Transient Turning.....	109
1.11	Command Following Problem.....	109
Section 2:	Dynamics of Steering.....	111
2.1	The Bicycle Model.....	111
2.2	Fresnel Integrals.....	112
2.3	Dynamics of the Constant Speed Reverse Turn.....	113
2.4	The Clothoid Generation Problem.....	114
2.5	Configuration Space.....	115
Section 3:	Positioning Fidelity.....	116
3.1	Absolute Attitude Accuracy Requirement.....	116
3.2	Rigid Terrain Assumption.....	116
3.3	Rigid Suspension Assumption.....	116
3.4	Image Registration Problem.....	117
3.5	Linear Relative Accuracy Requirement.....	117
3.6	Angular Relative Accuracy Requirement.....	117
Section 4:	Perceptual Fidelity.....	118
4.1	Incidence Sensitivity Problem.....	118
4.2	Attitude Sensitivity Problem.....	118
Chapter 6:	Interactions.....	119
Section 1:	Rationale for a Feedforward Approach.....	119
1.1	Command Following Problem.....	119
1.2	Modelling Problem.....	119
1.3	Dynamics Feedforward.....	120
1.4	Impact of Feedforward Dynamics on Computational Complexity.....	120
1.5	Impact of Dynamics on Planner/Controller Hierarchy.....	121
1.6	A Real Time Control View of High-Speed Autonomy.....	121
Section 2:	Rationale for Adaptive Perception.....	123
2.1	Throughput Problem.....	123
2.2	The Illusion.....	124
2.3	Adaptive Perception.....	124
2.4	Assumptions of the Analysis.....	125
2.5	Common Throughput Expression.....	125
2.6	Basic Mechanism.....	126
2.7	Complexity of Constant Flux Range Image Processing.....	127
2.8	Complexity of Adaptive Sweep Range Image Processing.....	128
2.9	Complexity of Adaptive Sweep/Scan Range Image Processing.....	131
2.10	Complexity of Adaptive Sweep, Uniform Scan Image Processing.....	133
2.11	The Fundamental Speed/Resolution Trade-off.....	135
Section 3:	Rationale for A Real-Time Approach.....	137

3.1	Computational Spiral Effect.....	137
3.2	Impact of the Spiral Effect.....	138
Chapter 7:	Summary and Conclusions.....	140
Section 1:	Elements.....	140
1.1	Summary of Elementary Nondimensionals.....	140
1.2	Derivative Issues.....	140
1.3	Derivative Adaptive Rules.....	141
Section 2:	Configuration.....	142
2.1	Summary of Configuration Nondimensionals.....	142
Section 3:	Response.....	143
3.1	Summary of the Response Nondimensionals.....	143
3.2	Maneuverability and Response Time.....	144
3.3	Braking.....	144
3.4	Turning.....	144
3.5	Lookahead.....	144
Section 4:	Throughput.....	146
4.1	Summary of Throughput Nondimensionals.....	146
4.2	Depth of Field.....	146
4.3	Horizontal Field of View.....	146
4.4	Vertical Field of View.....	146
Section 5:	Acuity.....	148
5.1	Summary of the Acuity Nondimensionals.....	148
5.2	Range Image Acuity.....	148
5.3	Range Acuity in Stereo Vision.....	149
5.4	Angular Acuity in Stereo Vision.....	149
5.5	Positioning Bandwidth.....	149
5.6	Geometric Efficiency.....	149
Section 6:	Fidelity.....	150
6.1	Summary of the Fidelity Nondimensionals.....	150
6.2	Modelling Dynamics and Delays.....	150
6.3	Dynamics of Turning.....	151
6.4	Positioning Fidelity.....	151
Section 7:	Interactions.....	152
7.1	Rationale For a Feedforward Approach.....	152
7.2	Rationale For Adaptive Perception.....	152
7.3	Rationale For a Real-Time Approach.....	153

PART III: Intelligent Autonomous Mobility

Chapter 1: Concept	155
Section 1: Standard Architectural Model	155
1.1 Policy Layer	156
1.2 Strategic Layer	157
1.3 Control Layer	157
1.4 Tactical Layer	158
Section 2: Bandwidth - Conceptualization Gap	159
2.1 Fast Execution Monitoring and Path Replanning	159
2.2 Smart Control	159
Section 3: Tactical Control Layer	160
3.1 Command Following Problem	160
3.2 Tactical Control layer	160
3.3 Hierarchical Time Spectrum	161
3.4 Uncertainty	161
Chapter 2: Design	162
Section 1: The Nature of High-Speed Autonomy	162
Section 2: Intelligent Predictive Control.....	163
2.1 Deliberative Elements	163
2.2 Reactive Elements	163
Chapter 3: Implementation	165
Section 1: Real-Time Kernel	165
1.1 Position Estimator	166
1.2 Map Manager	166
1.3 Vehicle	166
1.4 Controller	166
1.5 Perception	166
Section 2: Rationale - Bandwidth Requirements	167
2.1 Image Input Stream	167
2.2 Vehicle State Input Stream	167
2.3 Vehicle Command Output Stream	167
Section 3: Test Environment Layer.....	168
3.1 I/O Filter	169
3.2 General Rationale for Physical I/O	170
3.3 Main Event Loop	171
3.4 Development Support Tools	171
Section 4: Computational Hardware	172
4.1 Engineering Workstations	172

4.2	Implementation in Special Purpose Hardware	172
4.3	Implementation in Parallel Hardware	172
Chapter 4:	Results	173
Section 1:	Vehicles Automated	173
Section 2:	Test Run Excursion and Speed	174
Section 3:	Real-Time Performance	175
Chapter 5:	Summary and Conclusions	177
Section 1:	Concept	177
1.1	Standard Architectural Model	177
1.2	Bandwidth - Conceptualization Gap	178
1.3	Tactical Control Layer	178
Section 2:	Design	179
2.1	The Nature of High-Speed Autonomy	179
2.2	Intelligent Predictive Control	179
Section 3:	Implementation	180
3.1	Real-Time Kernel	180
3.2	Rationale - Bandwidth Requirements	180
3.3	Test Environment Layer	180
Section 4:	Results.....	181
4.1	Vehicles Automated	181
4.2	Test Run Excursion and Speed	181
4.3	Real-Time Performance	181

PART IV: Perception		
Chapter 1:	Concept.....	182
Section 1:	Adaptive Lookahead.....	183
Section 2:	Adaptive Sweep.....	183
2.1	Selection Problem.....	183
2.2	Computational Range Image Stabilization.....	183
2.3	Computational Stereo Range Image Stabilization.....	185
Section 3:	Adaptive Scan.....	185
Chapter 2:	Design.....	186
Section 1:	Adaptive Perception From Range Images.....	186
1.1	Range Window Computation - First Phase.....	186
1.2	Range Window Processing - Second Phase.....	187
Section 2:	Adaptive Stereo Perception.....	190
2.1	Basics of Stereo Perception.....	190
2.2	Stereo Triangulation.....	190
2.3	Width of Disparity Window.....	191
2.4	Essential Difficulty.....	191
Section 3:	Refinements to Correlation-Based Stereo.....	193
3.1	Optimal Shape of the Correlation Window	193
3.2	Complexity of Stereo Vision.....	193
3.3	Adaptive Scan in Stereo Vision.....	196
3.4	Adaptive Sweep.....	198
Section 4:	Terrain Mapping.....	199
4.1	Wrappable Map.....	199
4.2	Motion Distortion Removal.....	200
4.3	Delayed Interpolation.....	201
4.4	Image Registration.....	202
4.5	Elevation Uncertainty.....	202
Chapter 3:	Implementation.....	203
Section 1:	Coordinate Transformation.....	203
1.1	Ambiguity Removal.....	204
1.2	Median Filter.....	204
1.3	Range Window Filter.....	204
1.4	Range Window Computation.....	204
1.5	Image to Sensor.....	204
1.6	Sensor to Vehicle.....	204
1.7	Vehicle to World.....	204
1.8	Pose Selector.....	204

Section 2:	Map Manager.....	205
2.1	World to Map.....	205
2.2	Map to World.....	205
2.3	Registration.....	205
2.4	Fusion/Accumulation.....	206
2.5	Write Map.....	206
2.6	Read Map.....	206
2.7	Age Filter.....	206
Section 3:	Stereo Vision.....	207
3.1	Scale and Normalize.....	207
3.2	Rectify.....	207
3.3	Correlation.....	207
3.4	Disparity.....	208
3.5	Cleanup.....	208
3.6	Triangulation.....	208
Chapter 4:	Results.....	209
Section 1:	Parameter Configuration.....	209
Section 2:	Range Images.....	211
Section 3:	Stereo Vision.....	214
3.1	Horizontal Baseline.....	214
3.2	Vertical Baseline.....	215
Chapter 5:	Summary and Conclusions.....	217
Section 1:	Concept.....	217
1.1	Adaptive Lookahead.....	218
1.2	Adaptive Sweep.....	218
1.3	Adaptive Scan.....	218
Section 2:	Design.....	218
2.1	Adaptive Perception From Range Images.....	218
2.2	Adaptive Stereo Perception.....	219
2.3	Refinements to Correlation-Based Stereo.....	219
2.4	Terrain Mapping.....	220
Section 3:	Implementation.....	222
3.1	Coordinate Transformation.....	222
3.2	Map Manager.....	222
3.3	Stereo Vision.....	222
Section 4:	Results.....	222
4.1	Range Images.....	222
4.2	Stereo Vision.....	222

PARTV: Planning/Control		
Chapter 1: Concept.....224		
Section 1:	Optimal Control.....	225
1.1	Hierarchical Planning.....	225
1.2	Optimal Control.....	226
1.3	State Space Model.....	227
1.4	Actuation Space Search.....	228
1.5	Feedforward Optimal Control.....	229
1.6	Forward Modelling.....	230
Section 2:	Obstacle Avoidance.....	232
Section 3:	Goal-Seeking.....	233
Chapter 2: Design.....234		
Section 1:	Optimal Control.....	234
1.1	Goal Arbitration.....	234
1.2	State Space Model.....	235
1.3	Discrete Time Nonlinear Model.....	240
Section 2:	Obstacle Avoidance.....	242
2.1	Temporal State Interpolation.....	242
2.2	Hazard Representation.....	242
2.3	Hazard Arbitration.....	243
2.4	Hazard Estimation.....	243
2.5	Matrix Representation.....	245
2.6	Hazard Uncertainty.....	246
Section 3:	Goal-Seeking.....	247
3.1	Rough Terrain Pure Pursuit.....	247
3.2	Feedforward Pure Pursuit.....	248
Section 4:	Adaptive Regard.....	249
4.1	Planning Window.....	249
4.2	Real-Time Latency Modelling.....	250
Section 5:	Adaptive Regulators.....	253
5.1	Sensor Stabilization and Pointing.....	253
5.2	Speed Planning.....	254
Chapter 3: Implementation.....256		
Section 1:	Controller.....	256
1.1	Command Generator.....	257
1.2	Tactical Controller.....	257
1.3	Strategic Controller.....	259
1.4	Arbiter.....	260

Section 2:	Vehicle.....	261
2.1	System Model.....	261
2.2	FIFO Queue Daemons.....	262
2.3	Vehicle Daemons.....	263
Chapter 4:	Results.....	264
Section 1:	Optimal Control.....	264
Section 2:	Obstacle Avoidance.....	265
Section 3:	Goal-Seeking.....	266
Section 4:	Adaptive Regulation.....	267
Chapter 5:	Summary and Conclusions.....	268
Section 1:	Concept.....	268
1.1	Optimal Control.....	268
1.2	Hierarchical Planning.....	269
Section 2:	Design.....	270
2.1	Optimal Control.....	270
2.2	Obstacle Avoidance.....	271
2.3	Goal-Seeking.....	272
2.4	Adaptive Regard.....	272
2.5	Adaptive Regulators.....	273
Section 3:	Implementation.....	273
3.1	Controller.....	273
3.2	Vehicle.....	273
Section 4:	Results.....	273
4.1	Optimal Control.....	273
4.2	Obstacle Avoidance.....	273
4.3	Goal-Seeking.....	274
4.4	Adaptive Regulation.....	274

PART VI: Position Estimation

Chapter 1:	Concept.....	275
Section 1:	State Space Kalman Filter.....	276
1.1	Positioning Requirements.....	276
1.2	Design Decisions and Assumptions.....	277
Section 2:	AHRS Dead Reckoning.....	278
2.1	AHRS Components.....	278
2.2	Other Sensors.....	278
2.3	Extended Kalman Filter.....	278
2.4	Unforced Kalman Filter.....	278
Section 3:	Advantages.....	279
3.1	Redundant Measurements.....	279
3.2	Integration of Dead Reckoning and Triangulation.....	279
3.3	Modelling Frequency Response.....	279
3.4	Computational Inertial Force Compensation.....	279
3.5	Nonlinear Error Propagation.....	279
3.6	Noise Immunity.....	279
Chapter 2:	Design.....	280
Section 1:	AHRS Dead Reckoning System Model.....	280
1.1	Nav Frame System Model.....	280
1.2	Observability in the Nav Frame.....	281
1.3	Body Frame System Model.....	282
Section 2:	AHRS Dead Reckoning Measurement Model.....	285
2.1	Transmission Encoder.....	285
2.2	Doppler Groundspeed Radar.....	285
2.3	Compass.....	286
2.4	AHRS.....	286
2.5	Steering Wheel Encoder.....	286
2.6	Wheel Encoders.....	287
2.7	Complete DR Measurement Matrix.....	289
Section 3:	AHRS Dead Reckoning Uncertainty Model.....	290
3.1	State Uncertainty.....	290
3.2	Measurement Uncertainty.....	294
Section 4:	Aided AHRS Dead Reckoning.....	297
4.1	Fixes in the Navigation Frame.....	297
4.2	Fixes in a Positioner Frame.....	298
Section 5:	Roadfollower Aiding.....	299
5.1	Measurement Model.....	299
5.2	Predictive Measurement.....	302

5.3	Advantages.....	302
Section 6:	Terrain-Aided AHRS Dead Reckoning.....	303
6.1	Terrain Aids.....	303
6.2	Measurement Uncertainty.....	304
Section 7:	Absolute Landmark Recognition.....	305
7.1	Absolute Landmarks.....	305
7.2	Feature-Based Scheme.....	306
7.3	Uncertain Absolute Landmarks.....	307
7.4	Iconic Scheme.....	308
Section 8:	Differential Landmark Recognition.....	309
8.1	Differential Landmarks.....	309
8.2	Feature-Based Scheme.....	309
8.3	Iconic Scheme.....	309
8.4	Examples of Differential Aids.....	310
Section 9:	Simultaneous Mapping and Position Estimation.....	311
9.1	State Vector Augmentation.....	311
9.2	Measurement Model.....	311
9.3	Measurement Jacobian.....	311
Section 10:	Real Time Identification.....	312
10.1	State Vector Augmentation.....	312
10.2	Measurement Model.....	312
10.3	Measurement Jacobian.....	313
Chapter 3:	Implementation.....	314
Section 1:	Position Estimator.....	314
1.1	System Model.....	315
1.2	Sensor Model.....	315
1.3	Kalman Filter.....	315
Section 2:	System Model.....	315
2.1	Transition Matrix.....	315
2.2	State Update.....	315
2.3	System Jacobian.....	316
2.4	Covariance Update.....	316
Section 3:	Sensor Model.....	316
3.1	Prediction.....	316
3.2	Gradient.....	316
3.3	Covariance.....	316
Section 4:	Kalman Filter.....	317
4.1	Kalman Gain.....	317
4.2	Residual.....	317
4.3	Covariance Update.....	317
4.4	State Update.....	317

Section 5:	Calibration and Tuning.....	318
5.1	Relative Navigation.....	318
5.2	Zups.....	319
5.3	State Vector Augmentation.....	319
Section 6:	Initialization and Reconfiguration.....	320
6.1	Initialization.....	320
6.2	Reconfiguration.....	320
Section 7:	Bandwidth and Efficiency.....	321
7.1	Asynchronous Implementation.....	321
7.2	Efficiency.....	321
Section 8:	Outlier Rejection.....	323
8.1	Angular Addition.....	323
8.2	Reasonableness Checks.....	323
8.3	Legitimate State Discontinuities.....	323
Chapter 4:	Results.....	324
Section 1:	AHRS Dead Reckoning.....	324
Section 2:	Crosstrack Aiding.....	327
Chapter 5:	Summary and Conclusions.....	328
Section 1:	Concept.....	328
Section 2:	State Space Kalman Filter.....	328
2.1	Design.....	329
2.2	AHRS Dead Reckoning System Model.....	329
2.3	AHRS Dead Reckoning Measurement Model.....	330
2.4	AHRS Dead Reckoning Uncertainty Model.....	330
2.5	Aided AHRS Dead Reckoning.....	330
2.6	Terrain-Aided AHRS Dead Reckoning.....	331
2.7	Simultaneous Mapping and Position Estimation.....	331
Section 3:	Real Time Identification.....	331
3.1	Implementation.....	331
3.2	Position Estimator.....	331
3.3	System Model.....	331
3.4	Sensor Model.....	331
3.5	Kalman Filter.....	332
3.6	Calibration and Tuning.....	332
3.7	Initialization and Reconfiguration.....	332
3.8	Bandwidth and Efficiency.....	332
Section 4:	Outlier Rejection.....	332
	Results.....	333
4.1	AHRS Dead Reckoning.....	333
4.2	Crosstrack Aiding.....	333

PART VII: Summary and Conclusions

Section 1:	Problem Statement.....	334
Section 2:	RANGER Navigator.....	335
2.1	Operational Modes.....	335
2.2	Goal-Seeking.....	335
2.3	World Model.....	335
2.4	Vehicle Model.....	336
2.5	Hazard Assessment.....	336
2.6	Arbitration.....	337
2.7	Sensors.....	337
2.8	Adaptive Perception.....	337
2.9	Position Estimation.....	338
2.10	Implementation.....	338
2.11	Architecture.....	338
Section 3:	Perspectives on Intelligence.....	339
3.1	On Memory.....	339
3.2	On Deliberation.....	339
3.3	On Reaction.....	339
Section 4:	Fundamental Arguments.....	340
4.1	Guaranteed Safety.....	340
4.2	Response.....	340
4.3	Throughput.....	340
4.4	Acuity.....	340
4.5	Fidelity.....	341
Section 5:	Key Assumptions.....	342
5.1	Continuity Assumption.....	342
5.2	Terrain Smoothness Assumption.....	342
5.3	Obstacle Sparsity Assumption.....	342
5.4	Small Incidence Angle Assumption.....	342
5.5	2-1/2D World Assumption.....	342
5.6	Static Environment Assumption.....	343
Section 6:	Conclusions.....	344

PART VIII: Contributions

Section 1:	General Claims	346
1.1	Codification	346
1.2	Classification	346
1.3	Demonstration	346
Section 2:	Specific Claims	347
2.1	Perspective	347
2.2	Approach	347
2.3	Generality	347
Section 3:	Limitations / Future Work.....	348
3.1	Strategic Planning	348
3.2	Stereo Vision	348
3.3	Identification	348
3.4	Coordinated Control	348
3.5	Map Matching	348
3.6	Kalman Filter	348

PART IX: Appendices

Chapter 1:	Kinematic Transforms	350
Section 1:	Notational Conventions	351
Section 2:	Homogeneous Coordinates and Transforms	352
2.1	Points	352
2.2	Operators	352
2.3	Homogeneous Coordinates	352
2.4	Homogeneous Transforms	353
2.5	Homogeneous Transforms as Operators	354
2.6	Example - Operating on a Point	355
2.7	Homogeneous Transforms as Coordinate Frames	355
2.8	Example - Interpreting an Operator as a Frame	356
2.9	The Coordinate Frame	357
2.10	Homogeneous Transforms as Coordinate Transforms	357
2.11	Example - Transforming the Coordinates of a Point	359
2.12	Inverse of a Homogeneous Transform	359
2.13	A Duality Theorem	360
Section 3:	Forward Kinematics.....	361
3.1	Nonlinear Mapping	361
3.2	Mechanism Models	361
3.3	Modelling a Mechanism	362
3.4	Denavit Hartenberg Convention for Mechanisms	362
3.5	Example: 3-Link Planar Manipulator	364
Section 4:	Inverse Kinematics.....	366
4.1	Existence and Uniqueness	366
4.2	Technique	366
4.3	Example: 3 Link Planar Manipulator	367
4.4	Standard Forms	370
Section 5:	Differential Kinematics.....	372
5.1	Derivatives of Fundamental Operators	372
5.2	The Mechanism Jacobian	372
5.3	Singularity	373
5.4	Example: 3-Link Planar Manipulator	374
5.5	Jacobian Determinant	374
5.6	Jacobian Tensor	375
Section 6:	Vehicle Kinematics	376
6.1	Axis Conventions	376
6.2	Frame Assignment	376
6.3	The RPY Transform	378
6.4	Frame Parameters for Wheeled Vehicles	380
6.5	Inverse Kinematics for the RPY Transform	380

6.6	Angular Velocity.....	382
Section 7:	Actuator Kinematics.....	384
7.1	The Bicycle Model.....	384
Chapter 2:	Imaging Kinematics.....	385
Section 1:	Optical Kinematics.....	385
1.1	The Reflection Operator.....	385
1.2	The Vector Projection Operator.....	386
1.3	The Plane Projection Operator.....	386
1.4	Rate Relationships.....	387
1.5	Mirror Gain.....	388
1.6	Composition of Translation and Reflection.....	389
1.7	Composition of Rotation and Reflection.....	390
1.8	Intersection of a Line and a Plane.....	391
Section 2:	Kinematics of the Azimuth Scanner.....	392
2.1	Forward Kinematics.....	392
2.2	Forward Imaging Jacobian.....	394
2.3	Projection Table.....	395
2.4	Inverse Kinematics.....	396
2.5	Inverse Imaging Jacobian.....	396
2.6	Analytic Range Image of Flat Terrain.....	397
2.7	Field of View.....	398
2.8	Resolution.....	399
2.9	Azimuth Scanning Pattern.....	400
2.10	Azimuth Spot Pattern.....	401
Section 3:	Kinematics of the Elevation Scanner.....	402
3.1	Forward Kinematics.....	402
3.2	Forward Imaging Jacobian.....	404
3.3	Projection Table.....	405
3.4	Inverse Kinematics.....	406
3.5	Inverse Imaging Jacobian.....	406
3.6	Analytic Range Image of Flat Terrain.....	407
3.7	Field of View.....	408
3.8	Resolution.....	409
3.9	Elevation Scanning Pattern.....	410
3.10	Elevation Spot Pattern.....	411
Section 4:	Kinematics of the Stereo Range Image.....	412
4.1	Perspective Projection.....	412
4.2	Stereo Triangulation.....	413
4.3	Forward Kinematics.....	414
4.4	Forward Imaging Jacobian.....	414
4.5	Projection Table.....	415
4.6	Inverse Kinematics.....	417
4.7	Inverse Imaging Jacobian.....	417

4.8	Analytic Range Image of Flat Terrain.....	418
4.9	Disparity Image of Flat Terrain.....	419
4.10	Disparity Gradient of Flat Terrain.....	420
4.11	Field of View.....	421
4.12	Resolution.....	422
4.13	Perspective Scanning Pattern.....	423
4.14	Perspective Spot Pattern.....	424
Chapter 3:	The State Space Kalman Filter.....	425
Section 1:	Random Processes.....	425
1.1	Random Constant.....	425
1.2	Bias, Stationarity, Ergodicity and Whiteness.....	426
1.3	Correlation Functions.....	426
1.4	Power Spectral Density.....	427
1.5	White Noise.....	428
1.6	Kalman Filter Noise Model.....	428
1.7	Scalar Uncertainty Propagation.....	430
1.8	Combined Observations of a Random Constant.....	430
Section 2:	Fundamentals of the Discrete Kalman Filter.....	431
2.1	The State Model of a Random Process.....	432
2.2	A Word on the Transition Matrix.....	433
2.3	Low Dynamics Assumption.....	433
2.4	Discrete Filter Equations.....	434
2.5	Time and Updates.....	435
2.6	Measurement Model.....	436
2.7	Observability.....	436
2.8	Uncertainty Transformation.....	437
2.9	Sequential Measurement Processing.....	437
2.10	The Uncertainty Matrices.....	437
2.11	Connection to Navigation Theory.....	438
2.12	Connection to Smith and Cheeseman.....	438
Section 3:	Linearization of Nonlinear Problems.....	439
3.1	Linearized Kalman Filter.....	439
3.2	Extended Kalman Filter.....	440
3.3	Taylor Remainder Theorem.....	442
3.4	Forms of Linearization.....	442
3.5	State Transition for Nonlinear Problems.....	444
3.6	System Jacobian for Nonlinear Problems.....	444
3.7	Uncertainty Propagation for Nonlinear Problems.....	444
3.8	Other Kalman Filter Algorithms.....	445
3.9	The Measurement Conceptualization in the EKF.....	445
Section 4:	State Vector Augmentation.....	446
4.1	Principle.....	446
4.2	Correlation Models.....	447

PART I Introduction

Autonomous vehicles promise many ultimate applications in tasks which are dangerous, difficult, tedious, or error-prone when performed by humans. Civilian applications include smart automobiles, automated delivery systems, autonomous farming, and autonomous mining. Military applications include battlefield reconnaissance, troop resupply, and medical evacuation of injured personnel. Space applications include planetary exploration, autonomous launch vehicles, and vehicle inspection robots. Regardless of the application, missions which require motions which exceed the reach of a typical manipulator can be addressed by the mobile robot.

The problem of autonomous off-road navigation is one of the most difficult terrestrial autonomous navigation problems. Earlier solutions to the cross-country navigation problem have encountered performance limitations that were not well understood.

This thesis has studied the nature of typical performance limits, expanded the envelope in a significant way, and demonstrated the higher capability on a real vehicle. It presents the principles behind and the development of an integrated autonomous cross-country navigation system incorporating perception, control, and position estimation called RANGER¹. RANGER is a software technology which provides basic high performance intelligent mobility of an autonomous vehicle. Using the NAVLAB II testbed at Carnegie Mellon University, it has been developed to directly address the requirements of a rudimentary off-road mission.

Section 1: Problem Statement

The thesis applies a structured systems analysis and design methodology to the problem. The problem that is addressed can be stated in two points:

- **assess the fundamental requirements of range image based, high-speed, autonomous cross-country navigation**
- **demonstrate an improved ability to achieve a real cross-country mission**

This real cross-country mission is a simple **piloting mission** - to get from A to B as fast as possible without damaging the vehicle. The **strategic planning problem** - the problem of generating the path to follow - is outside the scope of the work. The path to be followed is imparted to the system as an externally-supplied continuous path or sequence of waypoints.

1. An acronym for Real Time Autonomous Navigator with a Geometric Engine.

Section 2: Requirements

An attempt has been made to develop an autonomous vehicle control system which achieves:

- the highest possible speed of continuous motion
- on the roughest possible terrain
- for the longest possible excursion

Within the simple navigation context of the work, the following requirements are the highest level objectives.

2.1 Continuous, High Speed

Speeds which approach the speeds at which humans can drive the same terrain (say, 20 mph) are targeted. Ideally, the system would adjust its speed based on the terrain roughness - driving faster when the terrain permits.

2.2 Rough Terrain

Rugged, sparsely-vegetated, mostly dry terrain is the target environment within which the vehicle must navigate safely. A flat world assumption is not considered valid² in the target terrain nor is the typically useful assumption that obstacles can be represented as discrete points in a two-dimensional space.

2.3 Ackerman Steer Vehicle

While the Ackerman steering configuration is not ideal for planning purposes, most large vehicles intended for human use employ this configuration, so the restriction to this class of vehicles is not as limiting as it may appear. In any case, the maneuverability of such vehicles is dramatically different from other alternatives, particularly at speed, and the thesis will show that efficiency demands that a special case solution for this class of vehicles be adopted. Momentum and the nonholonomic constraint limit maneuverability of all high-speed vehicles, so many of the results of the work apply generally.

2.4 Robustness

The system is intended for relatively long traverses of suitable terrain. While robustness is difficult to quantify, it is the intention that reliability be sufficient for routine autonomous excursions of tens of kilometers over suitably chosen terrain.

2. Although this assumption is not valid, it will be used in many places as a useful theoretical approximation.

Section 3: Previous Work

This section summarizes the capabilities of some of the most significant historical off-road systems. The list is not exhaustive, nor is it a complete discussion of each system provided. Unfortunately, little has been published to date on some systems, and available publications do not address some of the issues investigated here.

3.1 Differences from Important Historical Systems

RANGER differs from all historical precedents in two ways. First, it explicitly evaluates vehicle safety in terms of timing, speed, resolution, and accuracy in real-time and employs many adaptive mechanisms which optimize performance. In particular, explicit computation of stopping distance and required minimum throughput is performed and these considerations and others form the basis of an elegant adaptive perception mechanism which makes unprecedented vehicle speeds possible. Second, the system uses relatively high-fidelity feedforward models of system dynamics and terrain following and is configured as a tightly coupled controller.

3.1.1 Hughes ALV

The Hughes Autonomous Land Vehicle [13] was one of the first off-road mobile robots ever developed. The system generalized the notion of an obstacle to include any vehicle configuration that is unsafe. This allowed navigation over terrain of any surface shape. Based on the ERIM rangefinder and a highly terrainable vehicle, it achieved speeds up to 3.5 km/hr, over distances exceeding 500 meters, and running times averaging 20 minutes.

The speed of this vehicle was limited by many factors including the complexity of the perception and planning processing, and the speed of communication with off-board processing. Robustness was limited by low-fidelity modelling of steering constraints, poor terrainability and maneuverability of the vehicle on wet muddy soil, inability to detect small but dangerous obstacles (steel fence spikes), constrained excursion due to line of sight radio communications, local planning minima and bugs in the high-level planner.

The system discussed here differs from the ALV in its real-time minimalist approach to the problem. The adaptive perception algorithms are considerably faster than the Connection Machine implementation of the ALV algorithms. It also discards the costly energy minimization interpolation scheme of the ALV in favor of a temporal interpolation of the hazard vector signal. The system also differs in its high-speed design incorporating a differential equation state space model.

3.1.2 JPL's Robby

The JPL rover [82] was the first system to drive autonomously across outdoor terrain using dense stereo data, rather than sonar or laser rangefinder data. This system was demonstrated in a 100 meter cross-country demo in September 1990. It has achieved an average speed of 100 meters in 4 hours (7.0 mm/sec) in a start/stop mode. The total elapsed time in this case is not very meaningful, since it includes downtime while the system was being debugged and numerous stops to evaluate performance³. The vehicle speed was mechanically limited to 4 cm/sec and off-board

3. Larry Matthies, personal communication.

communication throughput was limited. Cycle times were on the order of 30 sees for planning and 10 sees for perception. Passive stereo was used in the perception system. Later runs were able to achieve speeds of 4 cm/sec.

The system discussed here differs from Robby in that it attempts both wide excursion and high speed. It is difficult to assess the degree to which computer evolution alone has made this possible.

3.1.3 JPL's Rocky Series

The Rocky series of Microrovers [78] are small-scale prototypes intended as testbeds for simple behavior-based control systems. They are intended to produce a flight rover for the MESUR pathfinder mission of 1996. These robots are tested in the context of operations in the vicinity of a planetary lander, so large traverse distance or high speed are not being pursued.

The system discussed here differs from the Rocky series in its deliberative approach to autonomy and its attempt to achieve wide excursions at higher speeds. It borrows the minimalist configuration principle in an attempt to optimize real-time performance.

3.1.4 CMU's FastNav

The FastNav project concentrated on achieving high vehicle speeds and robust obstacle detection on fairly benign terrain. The Cyclone single scanline laser rangefinder was used as the basis of the perception subsystem. It was used to check a global path for discrete obstacles, such as trees, rocks, and ditches, and stop accordingly. Obstacle detection was based on detecting deviations from a flat world. Path tracking used a position estimation system based on inertial navigation, dead reckoning, and the GPS satellite navigation system.

FastNav [74] made use of simple tire/soil interaction and vehicle dynamics models to navigate on locally flat terrain with sparse obstacles. Path tracking was demonstrated at 30 km/hr and obstacle detection was demonstrated at 3 m/s.

Some of the reliability problems that were experienced include high sweep rates of the Cyclone scanner due to vehicle pitching, and poor accuracy of the inertial navigation system. Early problems with instability of the tracking algorithm were fixed by incorporating a first order model of the steering dynamics into a feedforward compensator. The longest autonomous run was 1 kilometer⁴.

RANGER borrows heavily from the FastNav project in three ways. First, steering dynamics feedforward is generalized to a complete state space rough terrain model. Second, the mere fact that such speeds were possible pointed directly to a problem with the way in which nonadaptive image processing was a major cause of poor real-time performance. The line scanner has been an important existence proof that the bare throughput requirement at these speeds was far less severe than it appears to be. Third, the pure pursuit path tracker is an adaptation of the FastNav tracker for rough terrain.

RANGER also differs from FastNav in that it is a cross-country system, not a dirt road system. Full 3D models are employed and the considerable difficulties of rough terrain drive many aspects of the design.

4. Much of this is personal communication with Sanjiv Singh and Jay West. Existing documentation does not cover the issues mentioned completely.

3.1.5 CMU UGV Systems

From September 1990 to the present, CMU researchers have developed two preliminary versions of autonomous cross-country navigators. Both navigators are based on the Highly Mobile Multi-Wheeled Vehicle (HMMWV). The performance of both prototypes has been promising when compared with that of previous systems. Both systems are based on the ERIM laser rangefinder.

The first system [10] is similar in concept to the ALV. It generates a complete geometric description of the terrain in cartesian coordinates by transforming successive range images. It evaluates candidate vehicle trajectories by simulating the vehicle state forward in time in order to predict collisions.

This system was the first to model dynamic constraints on vehicle state in path planning. Path generation was based on an alternative to the spanning arcs approach of the ALV for path generation which is theoretically superior in cluttered environments.

Reliability problems for this system were associated with the poor fidelity of elevation maps, cycle time overruns, and path generation and planning algorithm failures. Nevertheless, this system had achieved its purpose of advancing the state of the art and serving as an experimental testbed.

The most impressive achievement consists of a test run of five kilometers comprised of some thirty orbits of the same circular path on substantially flat terrain with a total of less than ten obstacle avoidance maneuvers. Speed has never exceeded 3 meters per second, and only half of this speed has been achieved while avoiding obstacles.

RANGER borrows heavily from this system because it is basically the next prototype in this series. It differs from its predecessor in its adaptive perception algorithms, the efficiency of its admissability module, its universal 3D models, and it favors a controls feedforward approach over the cluttered environment search-based planner.

More recently, a second group [53] has developed a system called SMARTY which processes range images to recover regions of constant terrain gradient. This system has achieved runs over distances of 840 meters, at speeds of 2-3 m/s while avoiding natural obstacles, and it has probably exceeded these results as of this writing.

RANGER differs from SMARTY in that it can be considered to be more of a deliberative system on the spectrum of such systems because it remembers both vehicle state and environmental state from cycle to cycle at a higher level of detail than SMARTY does. RANGER is also theoretically superior to SMARTY on rough terrain at the cost of the extra processing and slower reaction time required to navigate in a fully 3D world model.

3.1.6 ALVINN

ALVINN is a road following system based on neural networks [68]. Although it solves a different problem, ALVINN provides a second precedent for the use of feedforward for the purposes of imparting stability at high speed.

3.1.7 VITA

The Daimler-Benz group has developed impressive road followers based on a control systems approach to road following. Although a different problem is solved by these systems, the control approach has been an important precedent.

3.2 Summary of Previous Work

Very few precedents exist in the area of autonomous cross-country navigation and no metrics of performance have yet been introduced to measure progress. For those systems discussed, comparisons have been made based on subjective assessments of reliability and the most rudimentary of metrics: speed, excursion, and run time.

The CMU full geometry system executed orbits of a circle, and its speed had never exceeded 3 meters/second, so the autonomous mile remains an unachieved goal among cross-country navigators. On this basis, a system which merely navigated for a mile at 10 mph (4.5 m/s) would be unprecedented. The primary reasons for this limitation include lack of robust obstacle avoidance and limited computational and perceptual throughput.

In general, systems which are targeted toward rough terrain have been slow and unreliable. Those that have been able to assume specific terrain roughness limitations have enjoyed higher speeds and reliability at the cost of the scope limitation caused by those assumptions.

Section 4: Approach

The design will commence with the system requirements and proceed to elaborate, through a top-down analysis, the form and function of all subsystems required to achieve the mission. The emphasis of the design will be on *achieving the mission*, and given the current state of the technology, this has led to an unprecedented emphasis on designed-in system robustness.

The approach is viewed as a *systems approach*. The entire problem of basic intelligent mobility in the target environment is examined. Subsystem interactions are quantified. Problem constraints are exploited. Assumptions are enforced consistently throughout. The conjecture of the thesis is that this approach has led to an optimal system from the perspective of utilization of computing resources.

4.1 Principles

The approach is based on the following principles:

- Achieving high performance in an engineered system is a different matter from achieving basic functionality. Performance is a quantitative attribute, and it is achieved through a quantitative optimization process.
- The reliability of an engineered system emerges from the reliability of its components. Truly reliable systems are designed to be that way. Reliability is achieved by directly addressing the reliability issue as part of the problem.
- High speed autonomous navigation is a problem with stringent real-time constraints. Guaranteed performance and reliability against those constraints is achieved through the techniques of real-time systems analysis and design.

4.2 Perspective

As currently addressed, the **local** high speed autonomous navigation problem is devoid of difficult cognitive content. This is because it can be characterized as that of:

- simulating the vehicle state forward in time
- utilizing a near perfect description of the terrain (at some gross resolution)
- evaluating the relative merits of candidate trajectories
- choosing to execute any, or some optimal, safe trajectory

The question of assessing vehicle safety is a matter completely described by simple physical mechanics. That is, for known terrain geometry, computation of vehicle orientation and obstacle collision is a relatively simple matter in principle - although inevitable uncertainties complicate the process somewhat.

This thesis will address the problem of designing an optimum system which meets the requirements of a mission and simultaneously respects the *broad* limitations of contemporary componentry. It is considered both likely and appropriate that results will include an evaluation of how contemporary components can be improved in the most cost-effective manner. In this sense, the work is much more theoretical than a quantitative engineering design.

Section 5: Acknowledgments

This work has benefitted from the assistance of many individuals who should be named here. One has to write a thesis as if the work was entirely one's own. Yet, none of us work in a vacuum and the overall contribution arises from the synergy of the team and the free exchange of ideas in an academic setting. Tony Stentz, Martial Hebert, Barry Brumitt and R. Coulter made significant creative contributions to this work and they deserve a large measure of credit for the groundbreaking work which lead to it. Several years ago, Tony's group, then consisting of Tony, the author, Barry, and 'R' implemented our first joint attack on this problem, and it was the experience of that endeavour which made this work possible.

Martial personally embodies the experience of CMU in this business of making vehicles go and his influence on this work has been valuable and deep. Larry Matthies provided a second vehicle upon which to continue the work at JPL which was an excellent forcing function for the software engineering of the code. Reid Simmons and Lalit Katragadda brought the code up on the RATLER vehicle, identified many problems, and made many insightful comments on the design.

5.1 Analysis

Barry Brumitt and the author investigated the impact of steering delays on the planning and path generation problems, respectively, prior to this work. Based on this earlier work, and the precedent of FastNav, the idea of generalizing steering dynamics feedforward to a complete state space model ultimately emerged. Barry did initial work on the extent of the dynamically feasible subset of C space. The C space planner constructed by Barry Brumitt and Tony Stentz provided the initial impetus for the use of a feedforward simulator.

The notion of relating pixel size to subtended length on a vertical surface has been around for some time on the ARPA UGV program. Its origin is unknown to the author. Wherever it came from, it is the key element which permits an analysis of computational complexity.

R. Coulter first investigated the braking performance of the CMU HMMWV in a quantitative manner. This led naturally to the question of quantifying steering performance.

Omead Amidi was probably the first person to suggest that a tightly coupled real-time implementation of the high level perceive-think-act loop would be ultimately necessary at high speed.

Dong Hun Shin and Sanjiv Singh were the first to recognize the significance of steering delay and the first to implement a feedforward solution on the FastNav vehicle. Shin implemented the first pure pursuit path tracker for the HMMWV vehicle to be used in off-road work. Shin identified the difficulty of the clothoid generation problem with the Fresnel integral. Dean Pomerleau later provided a second precedent for steering feedforward in the ALVINN system.

Many of the ideas contained here can be considered to be a natural evolution of the ideas of Tony Stentz which formed the basis of the first full geometry off-road navigator at CMU. This software system was implemented by Barry Brumitt, R. Coulter, the author, Bill Burky and George Mueller under the direction of Tony Stentz.

Tony first pointed out the difficulty of treating negative obstacles and started the development of adaptive regard by relating lookahead to the vehicle stopping distance. Tony also pointed out the basis for kinematic requirements on the vertical field of view.

Martial Hebert pointed out the sampling problem, the attitude sensitivity problem, and the significance of the pixel aspect ratio on throughput. Many conversations with Martial suggested the need for adaptive perception.

5.2 Perception

None of this work would have been possible without the guidance of Martial Hebert. Martial taught me terrain mapping, laser rangefinders, and guided the development of adaptive stereo. Many conversations with Martial led me into the search for a different way of doing range image perception. Martial Hebert and Bill Ross taught me the basics of correlation-based stereo.

Larry Matthies of the Jet Propulsion Laboratory educated me on the physics of remote sensing and the mathematics of pattern classification. Collaborations with Todd Litwin at JPL provided an opportunity to integrate the navigation system with Larry's stereo system.

5.3 Planning/Control

R. Coulter was the first to suggest that an optimal control approach to propulsion was worth investigating. R. educated me on the need for gravity compensation in propulsion control.

The admissability modules evolved from earlier versions that were designed and constructed by R. Coulter and Tony Stentz, and later refined by George Mueller. Earlier still, the Hughes ALV pioneered the field itself with a similar approach.

Larry Matthies and Todd Litwin of the Jet Propulsion Labs provided the opportunity to integrate the system with a wide field of view stereo ranging system which provided an opportunity to engineer the system to accept a generalized suite of environmental sensor inputs.

Larry educated me on his stereo algorithm. Todd collaborated with me on the software engineering of the code. Indirectly, Don Gennery told me the relationship between least squares suspension models and the minimum potential energy principle. Tam Nguyen explained the JPL propulsion controller to me.

5.4 Position Estimation

This work was partially inspired by a task given to me by Martial Hebert in the Introduction to Mobile Robots course at CMU when I served as the class teaching assistant. Ben Motazed conceived the basic idea of a general Kalman filter that could be used for position estimation and to damp relative crosstrack error for autonomous vehicle convoys.

My interest in navigation theory has its roots in a collaborative effort with R. Coulter where we attempted to uncover such a theory. My own interest in vehicle calibration forced the investigation of state vector augmentation. All of these elements come together in this filter.

Section 6: Notational Conventions

Throughout the document the 3 X 3 matrix R_a^b denotes a rotation matrix which transforms a displacement from its expression in coordinate system 'a' to its expression in coordinate system 'b'. The 4 X 4 matrix T_a^b denotes the homogeneous transform which transforms a vector from its expression in coordinate system 'a' to its expression in coordinate system 'b'. The 4 X 4 matrix P_a^b denotes a nonlinear projection operator *represented* as a homogeneous transform. In such notation, the vector normalization step is implicit in the transform. The matrix H_a^b denotes the Jacobian of the transformation from system 'a' to system 'b'.

The entire document will be necessarily loose about the specification of derivatives. If x and y are scalars, \bar{x} and \bar{y} are vectors, and X and Y are matrices, then all of the following derivatives can be defined:

$\frac{\partial y}{\partial x}$	a partial derivative	$\frac{\partial y}{\partial \bar{x}}$	a gradient vector
$\frac{\partial \bar{y}}{\partial \bar{x}}$	a vector partial derivative	$\frac{\partial}{\partial \bar{x}} \bar{y}$	a Jacobian matrix
$\frac{\partial Y}{\partial X}$	a matrix partial derivative	$\frac{\partial Y}{\partial \bar{X}}$	a Jacobian tensor

Generalization to higher order tensors are obvious.

Bolded nonitalic **text** indicates keywords that repeat in the document and appear in the index.

Section 7: Thesis Overview

A document this size cannot be read in a single sitting so this section is provided to attempt to tie the sections of the document together. The overall structure of the document is an analysis of requirements of off-road mobility followed by a design of the three major software subsystems of an autonomous navigator.

7.1 Analysis

In Part II: Analysis the requirements of the problem are analysed from the following perspective. First, the basic requirement of intelligent mobility is to maintain vehicle safety. Second, the safety requirement can be expressed in terms of timing, speed, resolution, and accuracy requirements on sensors, effectors, and computer models and algorithms.

Now the maintenance of vehicle safety requires that all four of these requirements must be met simultaneously. This implies that all of these requirements interact and are functions of vehicle speed and reaction time. The last section of the analysis part investigates these interactions and generates several theoretical results and conclusions. The two most important conclusions are as follows:

- Intelligent Mobility is a real-time control problem and because nonadaptive perceptual processing can destroy the ability of the system to react fast enough, adaptive perception mechanisms are necessary on contemporary serial computers in order to meet the response time requirements of high speed motion.
- The path planning horizon for hazard avoidance is limited by the capabilities of the sensor as well as the structure of the environment and the fact that the sensor must be mounted on the moving vehicle. As a result, the system can only predict its own performance over a period of time that is not much larger than its own reaction time. This implies that the system operates almost entirely in the transient regime of operations and necessitates the use of dynamic models to capture the large differences between what is commanded of the vehicle and how it responds.

These two conclusions lead into the design of the perception and path planning systems of the navigator.

7*2 Mobflity

Part HI: Intelligent Autonomous Mobility is a commentary on the standard hierarchical architecture of many robotic systems and how the navigator design adapts it to the problem. The key element of adaptation is the introduction of the tactical control layer which acts as an intermediary between the geometric strategic layer and the dynamic control layer. A central reliability concern in high speed mobility is the fact that, because the system operates in the transient regime, a control scheme cannot cause a massive vehicle to follow any arbitrary command given to it.

The design places the local hazard avoidance and goal seeking behaviors squarely in the tactical control layer. This layer exists in order to deal intelligently with the realities of actuator and vehicle dynamics and it is the basic reason for the enhanced reliability of the system.

7.3 Perception

Part IV: Perception is the first of three parts structured in to four chapters outlining a central concept, a design overview, implementation details, and results. In the case of perception, the concept is that of adaptive perception. Here, the basic idea is that if the environment is self stationary, then it is not necessary to view the same parts of the environment over and over. Rather, perception throughput can be reduced to a bare minimum through several mechanisms which together constitute the adaptive perception algorithm.

It may seem at first glance that adaptive perception is impossible because the problem of selection of which data to process is as difficult as the original problem. However, if the environment is self-stationary, it is possible to compute very simply where the data of interest will be in an image without processing the entire image. This approach to perception is responsible for the enhanced performance of the system because it eliminates the strong dependence of vehicle speed on perceptual throughput.

7.4 Planning

Part V: Planning/Control outlines the design of the hazard avoidance and goal seeking behaviors of the tactical control layer. In addition to tactical control, another important idea in the design is the concept of adaptive regard which makes it possible to implement the tactical control layer efficiently.

Tactical control exists first of all in order to account for dynamics, but dynamics have other implications. First, they must be modelled somehow, so the design introduces a state space model of an autonomous vehicle which makes the degree to which a vehicle does not follow its commands explicit in the planning process. Second, dynamics of all kinds actually imply that the planning problem is relatively easy. The key point here is that the maneuverability of a high speed vehicle is limited as a matter of physics. To the degree that the number of distinct alternative trajectories that exist is limited, the complexity of motion planning search is also reduced.

This last point is formalized in terms of a notion called adaptive regard. The design is such that the system explicitly computes where the vehicle is already committed to go, and where it cannot go, and it eliminates these regions and the planning alternatives associated with them from consideration in order to minimize search.

A final important implementation idea is that of forward modelling. The state space model that is introduced is trivial to solve in the forward direction and at times impossible to solve in the inverse direction. This distinction has been noted before in the context of path generation for nonholonomically constrained vehicles and it is even more important when dynamics enter the picture. The system design exploits the advantages of forward models in order to enhance both reliability and performance.

7.5 Position Estimation

Part VI: Position Estimation describes the Kalman filter algorithm which forms the basis of a position estimation subsystem specifically designed for autonomous vehicles. The earlier perception and planning sections point to a need for high positioning bandwidth in order to remove the distortions associated with vehicle motion from sensory data. Also, autonomous vehicles provide a unique ability to observe environmental landmarks in order to improve position

estimation. For these two reasons, a special Kalman filter was developed for the navigator in order to both meet the special requirements of the problem and exploit the advantages of the typical vehicle instrumentation package.

7.6 Conclusions

The major results and conclusions of the work are summarized in Part VII: Summary and Conclusions. The contributions of the work are summarized in Part VIII: Contributions. Several appendices follow which provide mathematical background and analytical results which were deemed too detailed for the body of the document.

Section 8: How to Read This Thesis

It is a difficult matter to lay out a document of this size to satisfy the needs of those readers who will wish to skim it as well as those readers who will want the details. For this reason, several levels of summary have been provided throughout the document. Three levels of reading the document are intended to be more or less complete at that level.

8.1 Summary Level

To read a summary only, skip immediately to Part VII: Summary and Conclusions and then skip Part VI: Contributions.

8.2 Intermediate Level

To read an intermediate level of detail, read the summaries of each part as well as the material mentioned above.

8.3 Detail Level

The read all but the most detailed sections, read

- Part i
- Part II, Chapter 1: Elements, Chapter 6, Interactions, Chapter 7: Summary
- Part III, Chapter 1: Concept, Chapter 4: Results, Chapter 5: Summary
- Part IV, Chapter 1: Concept, Chapter 4: Results, Chapter 5: Summary
- Part V, Chapter 1: Concept, Chapter 4: Results, Chapter 5: Summary
- Part VI, Chapter 1: Concept, Chapter 4: Results, Chapter 5: Summary
- Part Vn
- Part Vm

The material in the appendices is not relevant to the flow of the document, but detailed results are presented there for readers who will want to see how some results are derived.

PART II: Analysis

At the speeds required of the next generation of autonomous vehicles, it becomes necessary to explicitly address the need for the vehicle to react to hazards in real time. As speeds increase, computing and sensory hardware and software must be re-evaluated to remove the bottlenecks that were acceptable at the lower speeds encountered in earlier research.

Higher speeds require both looking further ahead and reacting faster. Hence, autonomous navigation must mature to the point where significantly more computation is performed in less time, or some new approach must be embraced.

This part examines the general problem of high-speed autonomous navigation from range image data as it applies to both stereo and lidar sensing systems and develops a rudimentary theory of high-speed autonomous navigation based on real-time analysis, dimensional analysis and complexity analysis.

The conclusions of this part are the theoretical justification for the adaptive, real-time controller design of the RANGER cross country navigator.

Chapter 1: Elements

In order to intelligently guarantee its own safety, a high-speed vehicle must be able to resolve the smallest obstacle that can present a hazard, process sensory data at a rate commensurate with its speed, respond fast enough to avoid obstacles, and maintain a sufficiently accurate model of the world to enable it to make correct decisions. This section develops these requirements into a simple set of nondimensional requirements, and a related set of assumptions and design rules. These dimensions of the problem are analysed in a nondimensional manner and the implications of satisfying all requirements simultaneously are investigated.

Section 1: Guaranteed Safety Policy

One of the most fundamental requirements of autonomous mobility is the requirement to stay alive while achieving mission-level goals. The fundamental requirements of fast cross-country navigation are that of *robust, high-speed*, navigation over *rough terrain*. Of these, the robustness requirement is considered paramount. This section will cast robustness in terms of vehicle safety, and analyze it along the four dimensions of **timing, speed, resolution, and accuracy**. These four requirements are intrinsic to autonomy at any speed on any type of terrain but they are more difficult to meet at high speed on rough terrain.

1.1 Four Dimensions of Safety

There is some speed beyond which the problem of high-speed autonomy becomes one of intelligent real time control. In such a problem, a few basic failure modes are typically encountered:

- the system cannot react fast enough to respond to environmental events
- the system cannot process all of the data presented to it in the time allotted
- the system cannot recognize environmental events when they occur
- the system cannot decide on the proper course of action for a specific event

The first is a question of **reaction time** (timing), of whether the system is fast enough to react once it perceives a hazard. The second is a question of **throughput** (speed), of whether the system is fast enough to perceive everything needed. These questions are conceptually independent but become related when they are considered to vary dependency with resolution. *These first two dimensions limit the speed of a system.*

It is possible to construct a system with excellent reaction time that cannot supply the throughput necessary by simply reducing the field of view of a sensor to one, or a few pixels. Such a system could respond provided it happened to see a hazard. Conversely, it is possible to design a system with sufficient throughput, but poor reaction time - by processing range data out to the horizon. Such a system could see, but would be unable to respond. *There is a spectrum of how often to see how much*, and there is a single point on this spectrum which optimizes performance against both requirements.

The third is a question of **acuity**¹ (resolution), of whether the system has high enough resolution perceptual equipment to enable it to recognize important aspects of the environment. The fourth is a question of **fidelity** (accuracy), of whether the system has sufficiently accurate models to permit it to make correct decisions. *These last two dimensions limit the reliability of a system.*

1.2 Safety

1.2.1 Requirement - Guaranteed Safety

Together, response, throughput, acuity and fidelity requirements must be continuously met in order to guarantee vehicle safety. This will be called the policy of **guaranteed safety**. An adaptive system can directly implement the policy of guaranteed safety by reasoning in real time about the four dimensions of safety discussed above and by adapting its perception and planning subsystems to comply directly with the need for safety.

1.2.2 Mitigating Assumption - Safe Terrain Assumption

Although it may be obvious, it is important to mention that many systems can profitably reject the guaranteed safety requirement in favor of an implicit assumption that the terrain is safe. This is the **safe terrain assumption**. Human automobile drivers routinely make this assumption when driving on highways, and they do so both out of necessity and with some success. There are often situations when a driver would be unable to react to a stationary obstacle which suddenly appeared around a bend in the highway. This key assumption permits high-speed driving.

1. The document is replete with invented expressions used as a brevity device. Consider these expressions to be an aspect of notation like a mathematical symbol or an acronym.

1.3 Response

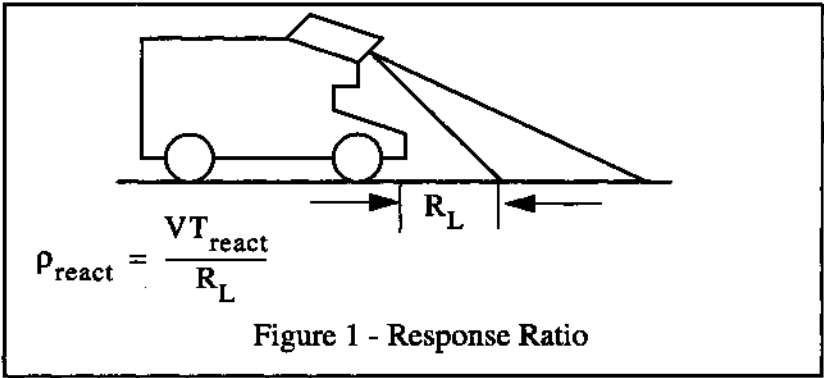
A vehicle must be able to react to an obstacle that it sees before it is reached. Therefore, the relationship between the distance that the vehicle looks ahead R_L , the reaction time T_{react} , and the speed V is central.

1.3.1 Requirement - Guaranteed Response

Following real-time systems terminology, the policy of guaranteeing a timely response to important sensory input events will be called the policy of **guaranteed response**. The problem of achieving guaranteed response will be called the **response problem**.

1.3.2 Nondimensional Requirement - Response Ratio

If it takes T_{react} seconds to react, then the **response ratio**⁴ is the ratio of the distance travelled during the reaction time to the distance at which the obstacle is perceived:



To say that a vehicle must avoid obstacles is equivalent to saying that the response ratio must never exceed unity. If the highest possible practical speed has been achieved, the only way to improve on this (while maintaining the ratio below 1) is to either decrease the reaction time or increase the lookahead distance. For a given vehicle with fixed latencies, and a given sensor with fixed maximum range, there exists some speed that cannot be safely exceeded without risking an encounter with hazards that were *seen too late to avoid them*. Adaptive systems can deliberately increase lookahead or reduce speed on the basis of the response ratio.

There are many related issues. There are practical limits on decreasing the reaction time due to the physical response of the vehicle - given power limitations on actuators, and the limited speed of the data processing hardware. There are also practical limits on increasing the lookahead because the maximum useful range of a sensor is often limited by image occlusions, limited accuracy, limited resolution, power limitations, or safety considerations, and most of these are extremely aggravated by shallow pixel incidence angles. Different reaction times apply to different obstacle avoidance maneuvers. Turning typically requires more time than braking, for instance. It is the response ratio which is the central concern, not the reaction time itself. A vehicle with half the speed which also responds half as quickly is equivalent. None of this matters until the sensor lookahead is considered.

2. The ratio of a velocity-time product to a distance is a central nondimensional variable in this kind of analysis.

1.3.3 Design Rules - Response Adapted Lookahead and Speed

Any element of the response ratio can be considered to be absolutely limited by the other two. The **response adapted lookahead rule** expresses how the planner lookahead must adapt to the state of the vehicle (speed, curvature) and its ability to respond (braking, turning) in order to guarantee that the response ratio remains less than unity. It can be written as:

$$\frac{R_T}{L} \leq \frac{V}{V_{\text{react}}}$$

Notice that the product $V T_{\text{react}}$ is a kind of characteristic vehicle distance which encodes the ability to respond expressed as a distance. An autonomous system must always scan for hazards *beyond* this characteristic distance.

The **response adapted speed** rule expresses how the vehicle speed must adapt to the sensor range and the ability of the vehicle to respond. It can be written as:

$$V \leq \frac{R_L}{T_{\text{react}}}$$

The ratio R_L / T_{react} is a kind of velocity which encodes the "speed" at which hazards present themselves to the vehicle. A more in-depth reaction time analysis reveals that various system latencies, including vehicle dynamics, cause reaction times that can be larger than might be expected, and therefore, maximum speeds are lower than might be expected.

1.3.4 Algorithmic Solutions - Response Adaptive Lookahead and Speed

The idea of **response adaptive lookahead** is to always ensure that the vehicle has time to react to any obstacle it may encounter at the current speed. The idea of **response adaptive speed** is to ensure that the vehicle speed always remains below the critical speed determined by the sensor maximum range. These measures are important because latencies are large and uncontrollable. Speed cannot be changed instantaneously and it is influenced to a great extent by the slope of the terrain.

1.3.5 Mitigating Assumptions - Low Latency Assumption, Wide Depth of Field Assumption

It may be possible under certain circumstances to simply ignore the issue of whether or not the system can respond quickly enough to avoid hazardous situations without explicitly considering it. Of course, this amounts to an assumption that response is instantaneous relative to any particular situation. This is the **low latency assumption**. An assumption with equivalent consequences is the assumption that sensor useful range is sufficiently large. This could be called a **wide depth of field assumption**. Vehicles which execute start-stop motions and slow speed vehicles can normally make these assumptions.

1.3.6 Related Subproblems - Latency Problem and Myopia Problem

The most central concern of guaranteed response is often the overall latency of the system. A **latency problem** exists when system latencies are too large for any particular goal speed and sensor maximum range. If the sensor maximum range is too small compared to the latencies and the speed, then a **myopia problem** exists.

1.4 Throughput

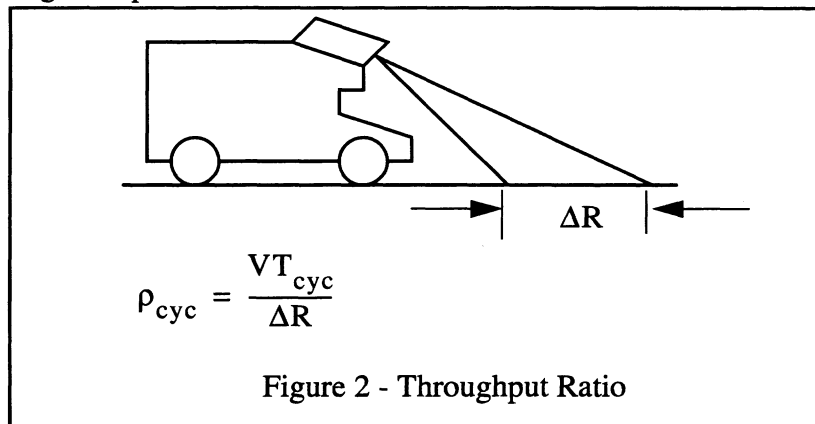
There is also a basic requirement that the system acquire geometric information as fast as it is consumed by driving over it. Therefore, the relationship between the projection of an image on the ground plane ΔR , the system cycle time T_{cyc} , and the speed V is also central³.

1.4.1 Requirement - Guaranteed Throughput

The policy of guaranteeing sufficient throughput will be called the policy of **guaranteed throughput**. The problem of achieving guaranteed throughput will be called the **throughput problem**.

1.4.2 Nondimensional Requirement - Throughput Ratio

The **throughput ratio** is the ratio of the distance travelled during the cycle time to the projection of an image on the groundplane:



To say that a vehicle must not drive over unknown terrain is equivalent to saying that the throughput ratio must never exceed unity. If the highest possible practical speed has been achieved, the only way to improve on this (while maintaining the ratio below 1) is to either decrease the cycle time or increase the image projection. For a given vehicle with fixed sensor and computational throughput, there exists some speed that cannot be safely exceeded without risking an *encounter with unseen hazards*. Adaptive systems can deliberately reduce speed or increase the sensor field of view on the basis of the throughput ratio.

There are many related issues. There are practical limits on decreasing the cycle time due to the computer speed and the sensor throughput. There are also practical limits on increasing the image projection due to pitching of the vehicle and the finite angular field of view of the sensor. Also, the maximum range is limited by many concerns, and the minimum range is limited by the height of the sensor and the extension of the vehicle nose in front of it. It is the throughput ratio which is the central concern, not the throughput itself. A sensor with half the field of view which generates twice the frame rate is equivalent. None of this matters until the velocity is considered.

3. Notice that response determines the minimum sensor range whereas throughput determines the vertical field of view. The two together specify a focus of attention which is the real issue at any point in time. This idea will be central later. Notice also that response determines reaction time whereas throughput determines cycle time. These two dimensions of real-time analysis are almost always important.

1.4.3 Design Rules - Throughput Adapted Sweep and Speed

Any element of the throughput ratio can be considered to be absolutely limited by the other two. The **throughput adapted sweep** rule expresses how the sensor field of view must be adapted based on the system cycle time and the vehicle speed. It can be written as:

$$AR \geq VT_{cyc}$$

The product VT_{cyc} is a kind of characteristic vehicle distance, the distance travelled per cycle, which encodes the basic throughput necessary. The **throughput adapted speed** rule expresses how the vehicle speed must be adapted based on the sensor field of view and the system cycle time. It can be expressed as follows:

$$V \leq \Delta R / T_{cyc}$$

The ratio AR/T_{cyc} is a characteristic speed which encodes the throughput necessary in terms of geometry per second. A more in-depth throughput analysis reveals that the efficiency with which traditional sensors generate geometry is unnecessarily low and that unprecedented throughput is possible by simply modifying the sensor geometry and optics without increasing its fundamental throughput⁴.

1.4.4 Algorithmic Solutions - Throughput Adaptive Sweep and Speed

The idea of **throughput adaptive sweep** is to always ensure that the vehicle acquires new environmental information as fast as it is consumed by driving over it. The idea of **throughput adaptive speed** is to ensure that the vehicle speed remains below a critical speed given by the cycle time and the sensor field of view. These measures are important because they can allow a system to achieve unprecedented speeds by computing only the minimum amount of information necessary to ensure safety.

1.4.5 Mitigating Assumption - High Throughput Assumption

It may be possible under certain circumstances to simply ignore the issue of whether or not the system can measure the environment fast enough. This amounts to an assumption that the computers are fast enough to process everything in an image without significantly affecting safety. This is the **high throughput assumption**.

1.4.6 Related Subproblems - Stabilization Problem, Tunnel Vision Problem

On rough terrain, it is possible that either the shape of the terrain in the image or the shape of the terrain upon which the vehicle moves will cause rapid motion of the sensor vertical sweep unless something is done about it. A **stabilization problem** exists when this motion can cause holes between images. Further, another aspect of sensor requirements is that they image all reachable terrain. When the field of view is too narrow to achieve this, a **tunnel vision problem** exists.

4. This is important for two reasons. Laser rangefinder throughput is limited fundamentally by the ability of the laser diode to shed generated heat coupled with the need to maintain reasonable signal-to-noise ratios. Stereo triangulation throughput is limited by the processing speed of the computer used. In both cases, there is much that can be achieved through judicious design of the sensor and/or processing algorithms.

1.5 Acuity

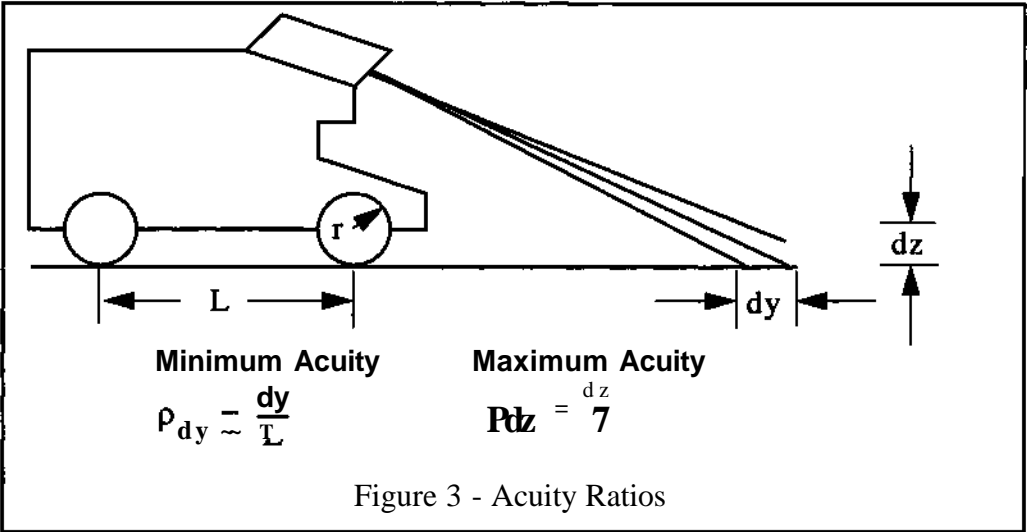
There is a basic requirement that the system be able to resolve the smallest obstacle that can present a hazard to the vehicle at operating velocity. This resolution requirement applies to both the sensor and the computations performed on the data. Clearly, the larger the vehicle, the larger the hazard necessary to challenge it, so it is to be expected that the resolution requirement will be dependent on the vehicle dimensions.

1.5.1 Requirement - Guaranteed Detection

In order to guarantee detection of hazards, some acuity requirement must be maintained at all times. This will be called the policy of **guaranteed detection**. The problem of maintaining adequate acuity will be called the **acuity problem**.

1.5.2 Nondimensional Requirement - Acuity Ratios

Considering the projection of a single range pixel onto the horizontal and onto a vertical surface, two very useful expressions can be formed:



These are called the **acuity ratios**. To say that a vehicle must resolve hazards reliably is equivalent to saying that the acuity ratios do not exceed one-half, as will be shown below.

1.5.3 Design Rules - Acuity Rules

In order to ensure that the vehicle pitch can be computed at all, Nyquist's sampling theorem tells us that the relevant acuity ratio must not exceed one-half. This can be written as:

$$dy \leq \frac{L}{2}$$

This will be called the **minimum acuity rule**, for below this resolution, a sensor measures nothing useful at all. Conversely, the resolution of a wheel step hazard will require that a pixel subtend no more than half of the height of the smallest obstacle. The smallest obstacle which presents a hazard is on the order of the wheel radius. Choosing to require two pixels on this surface can be written as:

$$dz \leq \frac{r}{2}$$

This will be called the **maximum acuity rule**, for pixel sizes smaller than this are excessively small⁵. There is some range at which the acuity requirements change relative severity, so in the most general case, both must be met simultaneously.

1.5.4 Algorithmic Solutions - Acuity Adaptive Scan and Planning

The idea of **acuity adaptive scan** is to actively modify sensor resolution in order to ensure adequate acuity over the entire field of view of the sensor. This measure is important because traditional sensors, such as the ERIM laser rangefinder, admit two orders of magnitude of variation in the density of pixels on the groundplane.

The idea of **acuity adaptive planning** is to accept that sensory information is finite in resolution and to perform planning computations at a consistent resolution. For example, vehicle pitch evaluated by moving the vehicle over the map cannot change until a distance of one map cell is moved, so there is no point in wasting computer cycles trying to extract higher resolution information. This would be equivalent graphically to using many small steps to integrate a function over an interval for which it is constant.

1.5.5 Mitigating Assumptions - Uniform Scan and Terrain Smoothness

In the most general case, pathological hazards may exist which are impossible to resolve for both sensor resolution and computational throughput reasons. A nail, for instance, could feasibly exist on the vehicle trajectory. Therefore it is necessary to assume that pathological cases do not exist. This will be called the **terrain smoothness assumption**. It must always be adopted to some degree, and it is adopted implicitly when a terrain map of any finite resolution is used in planning computations. This assumption is also important because the projection of a pixel onto the groundplane depends on the shape of the terrain surface itself⁶. To assume that pixel resolution is inherently adequate is to adopt a **uniform scan assumption**. One way to achieve almost uniform scan is to mount a sensor directly over the terrain of interest⁷.

1.5.6 Related Subproblems - Sampling Problem, Motion Distortion Problem

The true relationship between the angular resolution of any particular sensor and the linear resolution of the measurements it provides is affected primarily by the height at which the sensor is mounted and the shape of the terrain. In practice, it can vary by orders of magnitude over the field of view. This will be called the **sampling problem**.

Acuity has a temporal element as well as the spatial elements mentioned above. It is necessary to sample vehicle position fast enough to correctly localize range pixels. The distortion of an image that is caused by motion of the sensor is called the **motion distortion problem**.

5. Read excessively small as information theory. In practice, it is a good idea to require more than 2 pixels on the surface, as will be discussed later in terms of an **oversampling factor**.

6. From radiometry, the projected area of a surface varies with the cosine of the angle of incidence.

7. This, of course, can only be done for limited excursions or with a flying vehicle. Neither option is practical here, but both have limited domains of usefulness to other problems.

L6 Fidelity

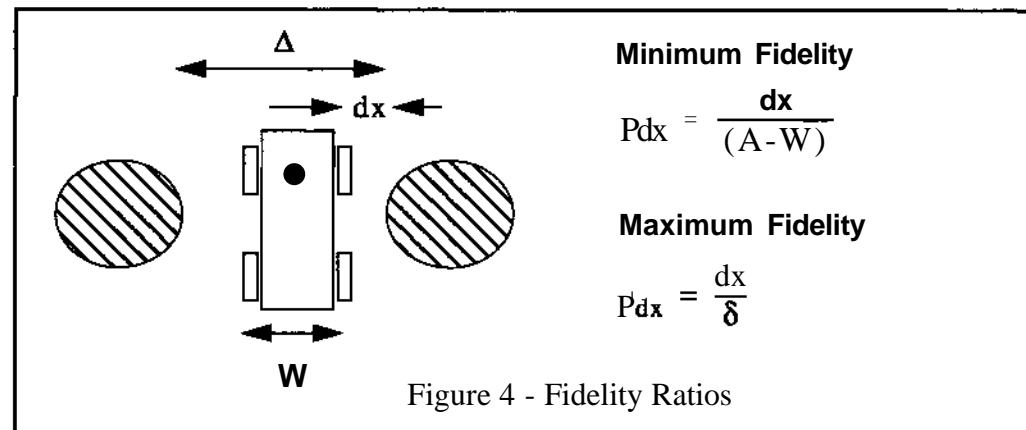
There is a basic requirement that the system be able to locate the vehicle with respect to hazards in the environment to within some limit of error⁸. This accuracy requirement applies to the range sensor, the position sensor, and the computations performed on the data. Clearly, the smaller the vehicle and the larger the distance between hazards, the lower this required accuracy is, so it is to be expected that the accuracy requirement will be dependent on the vehicle dimensions as well as the density of hazards.

1.6.1 Requirement - Guaranteed Localization

In order to guarantee localization of hazards, some fidelity requirement must be maintained at all times. This will be called the policy of **guaranteed localization**. The problem of maintaining adequate fidelity will be called the **fidelity problem**.

1.6.2 Nondimensional Requirement - Fidelity Ratios

Considering the groundplane projection of the vehicle and two hazard areas, two very useful expressions can be formed:



These are called the **fidelity ratios**. In the figure Δ is the minimum distance between hazards, W is the vehicle dimension aligned with the line between the hazards, dx is the maximum allowed error, and δ is the terrain **map resolution**.

1.6.3 Design Rules - Fidelity Rules

In order to ensure that the vehicle does not collide with the hazards, position error of the vehicle *relative to the hazards*, must not exceed the maximum, or equivalently the ratio must not exceed 1.

$$\frac{dx}{(\Delta - W)} \leq 1$$

This will be called the **minimum fidelity rule**, for below this accuracy, collision is guaranteed. Conversely, there is little point in measuring geometry to superb accuracy when it will be reduced to the intrinsic map resolution before it is used.

8. This is deliberately stated in relative terms. If both the hazard and the vehicle position are off by exactly the same error, system viability is not affected. Position estimation requirements for hazard avoidance are typically **relative accuracy** requirements.

$$\frac{dx}{\delta} \leq 1$$

This will be called the **maximum fidelity rule**, for accuracies better than this are excessive.

1.6.4 Algorithmic Solutions - Fidelity Adaptive Planning

When predicting the trajectory of the vehicle forward in time, the idea of **fidelity adaptive planning** is to accept that system accuracy is finite and to conservatively avoid obstacles so that hazards are guaranteed to be avoided despite the levels of uncertainty that exist. This measure is important because, in dense obstacle environments, the correct response to an overly dense collection of hazards is to avoid them as a unit instead of trying to drive between them.

1.6.5 Mitigating Assumption - Benign Terrain Assumption, Low Dynamics Assumption

It is clear that a vehicle cannot navigate between two hazards that are closer together than the size of the vehicle and in practice, it may be necessary to accept suboptimal localization from a poor sensor. To do so is to fundamentally assume that hazards are sparse in the environment. This will be called the **obstacle sparsity assumption** which is a special case of the **benign terrain assumption**. A special extreme case of the benign terrain assumption is the **flat world assumption**. In situations where dynamics can be neglected, a **low dynamics assumption** is being adopted.

1.6.6 Related Subproblems - Sensitivity Problem, Image Registration Problem, Stability Problem

There is often a high degree of sensitivity of particular parameters to changes in other parameters. For example, the localization of a range pixel is very sensitive to errors in the range measurement or in the angular measurement of the position of the ray through the pixel with respect to the navigation coordinate system. When sensitivity becomes an important consideration, a **sensitivity problem** exists. Terrain map⁹ fidelity is sensitive to many factors.

Also, guaranteed detection implies that an autonomous system is concerned mostly with predictions of vehicle position relatively far into the future. When nonlinear differential equations are involved in this prediction, there is the potential for extreme sensitivity to exist. The solution to the Fresnel equations for the vehicle position several seconds into the future, for example, is very sensitive to dynamic model miscalibration errors.

There are many other dimensions to the fidelity problem. The **image registration problem** arises when the relative accuracy of the entire system geometric model is insufficient to ensure that redundant measurements of the same geometry agree. Sometimes, situations can arise where there are not enough sensors to measure all quantities of interest. This problem is an extreme form of fidelity problem known as the **observability problem**. At times, poor models of the vehicle may cause control loop instability. This is called the **stability problem**.

9. A terrain map is a data structure used for representation of environmental geometry or other properties.

Section 2: Configuration - Normalized Vehicle

The shapes of most terrestrial vehicles vary little even while their sizes vary a lot. Most are about twice as long as they are high and about as wide as they are high. This is especially true of vehicles designed for roads. Simple models of tipover stability dictate that the height should be reduced as much as possible and, all other things being equal, length and width should be equal.

From the point of view of impact on a perception system, a few vehicle dimensions are most important. Consider the dimensions indicated in the following figure.

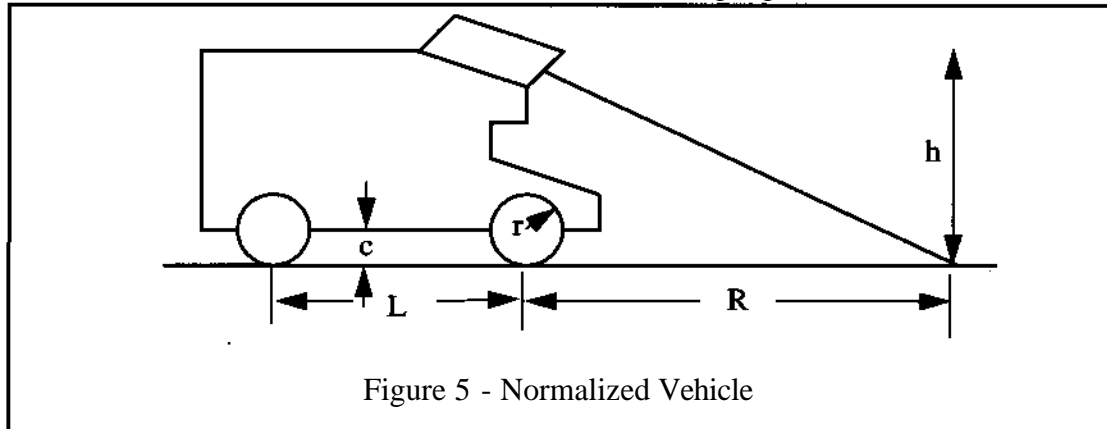


Figure 5 - Normalized Vehicle

Let the largest important dimension of the vehicle, the **wheelbase**, be called L . Define the **normalized wheelbase** to be:

$$\bar{L} = \frac{L}{R}$$

The normalized wheelbase relates each of the two following sets of three variables. Define the **perception ratio**, **normalized wheel radius**, and **normalized undercarriage clearance** to be:

$$\bar{h} = \left(\frac{h}{R} \right)$$

$$\bar{r} = \left(\frac{r}{R} \right)$$

$$\bar{c} = \left(\frac{c}{R} \right)$$

These variables will play a key role in rules which relate perception system requirements to the vehicle itself. By borrowing terminology from wing theory, the vehicle shape can be expressed in terms of aspect ratios. Define the **longitudinal aspect ratio**, **wheel fraction**, and **undercarriage tangent** to be:

$$A_L = \left(\frac{h}{L} \right)$$

$$A_r = \left(\frac{r}{L} \right)$$

$$A_c = \left(\frac{c}{L} \right)$$

These measure vehicle shape. The first measures how oblong the profile is. The second measures the overall roundness of the traction system and is one of many factors affecting terrainability. The third is another measure of terrainability and is a key element in uncovering the basic reason for occlusions in a perception system. Future results will refer frequently to the normalized vehicle. Of the seven variables presented, only four are independent.

Chapter 2: Response

As was shown earlier, the **response ratio** relates the ability of the vehicle to react, to its speed and its sensory lookahead. This section analyzes these aspects of vehicle performance for typical vehicles.

Section 1: Reaction Time

The **system reaction time** is the time period between the instant that an object appears in the field of view of the sensor and the instant of time when the vehicle can be considered to have reacted to it. It is important to be precise about this definition. The system level reaction time is the time from when an object is perceived until the actuator has finished its job. For example, in the case of braking, this time includes the time it takes for the vehicle to decelerate or skid to a stop.

1.1 Reaction Time Components

This aspect of performance depends on both software and hardware components. The total system reaction time includes the time required for all of the following operations:

- sensing the environment T_{sens}
- perceiving what the sensor data means T_{perc}
- deciding what to do T_{plan}
- commanding actuators T_{cont}
- actuator response T_{act}
- allowing the actuators to operate on the vehicle, environment, or both T_{veh}

The second last term may be called actuator dynamics and the last component will be referred to as vehicle dynamics - for lack of a better term. This can be expressed as follows:

$$T_{react} = T_{sens} + T_{perc} + T_{plan} + T_{cont} + T_{act} + T_{veh}$$

It will be useful later to distinguish three components of reaction time. These are the **sensing reaction time**:

$$T_{sens} = T_{sens}$$

the **processing reaction time**:

$$T_{sw} = T_{perc} + T_{plan} + T_{cont}$$

and the **dynamics reaction time**:

$$T_{dyn} = T_{act} + T_{veh}$$

These correspond to the sense-plan-act cycle view of autonomy. It will also be useful to group the first and last together into a **hardware reaction time**.

$$T_{hw} = T_{sens} + T_{dyn} = T_{sens} + T_{act} + T_{veh}$$

1.2 Nondimensional Response

The overall system reaction time is now simply:

$$T_{react} = T_{hw} + T_{sw}$$

The only physical dimension represented in this equation is time, so it is not very interesting from the point of view of dimensional analysis. A dimensional analysis can be accomplished by simple division. Consider:

$$\frac{T_{sw}}{T_{react}} + \frac{T_{hw}}{T_{react}} = 1$$

which states the intuitively obvious result that it is the relative ratios of the software and hardware components of the system reaction time which are important. This suggests that if software efficiency approaches the point where it accounts for only a small fraction of the total system reaction time, attempted performance improvements should concentrate on hardware and vice versa.

Define the **reactive ratio** as follows:

$$\rho_{react} = \frac{T_{sw}}{T_{react}}$$

This quantity measures the percentage of reaction time caused by software. Then the total system reaction time is:

$$T_{react} = T_{sw} / \rho_{react}$$

1.3 System Reaction Time

The system reaction time for a real case can be computed by tracing the flow of information through the basic system processing loops. These can be understood from the following flow diagram:

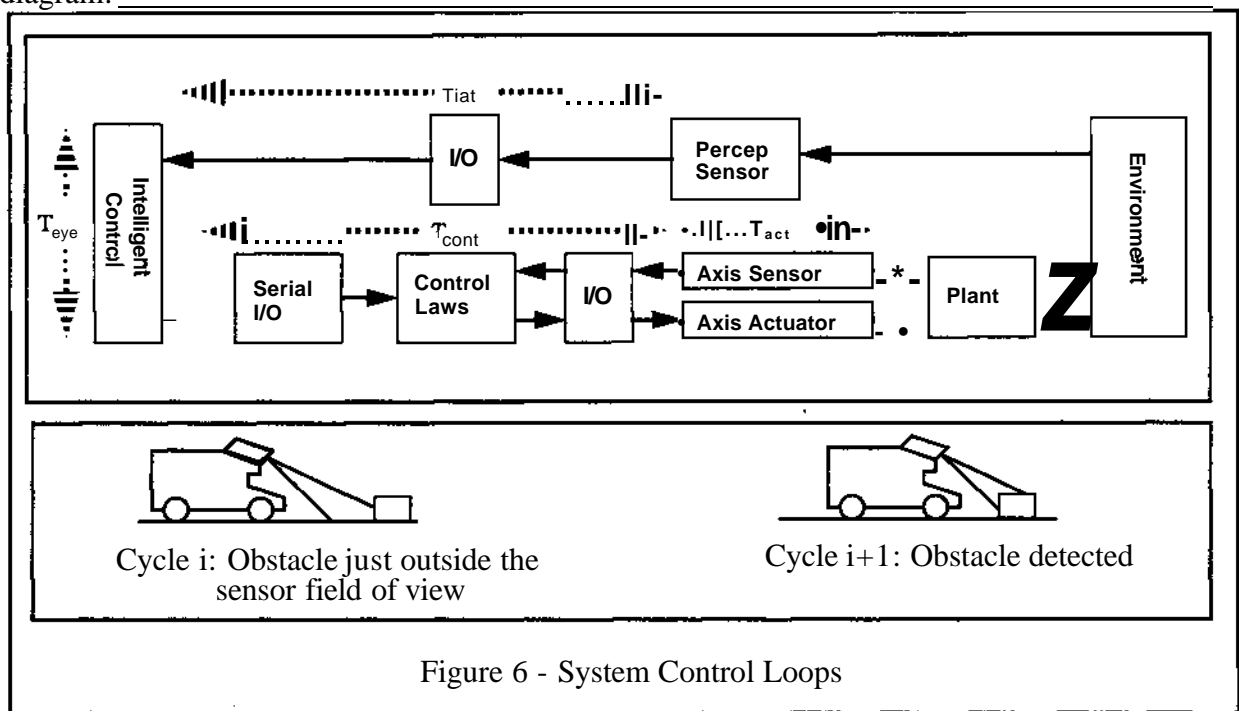


Figure 6 - System Control Loops

Consider that a clock is started the instant that an obstacle appears in the field of view of the sensor. After the frame buffer latency T_{lat} has elapsed, the obstacle appears in a new image in the frame buffer. Thus the sensing reaction time is the frame buffer latency:

$$T_{sens} = T_{lat}$$

Unless software is synchronized with the sensor, in the worst case, software has just started a new cycle immediately before the obstacle appears in the frame buffer. For this analysis, let the perception and planning subsystems cycle at some constant rate thus:

$$T_{eye} = T_{perc} + T_{plan}$$

After T_{cvc} more seconds have elapsed, software starts a new cycle on the latest image. This image may be the first, second or some other image since the obstacle was first seen, but the image does contain the obstacle.

When this image is processed in cycle $i+1$, T_{cyc} more seconds will elapse before the planner has decided to stop and issues the brake command. Hence $T_{lat} + 2T_{cyc}$ seconds of time have elapsed before the system has decided to react by braking the vehicle. This analysis assumes perfect obstacle detection. Next, the communication link to the control computer incorporates a delay called T_{cont} . After the controller receives the brake command, a delay of T_{act} applies before the

mass of the actuator moves far enough to be considered to have responded to the amplifier drive current. Finally, a time period of T_{veh} elapses before the vehicle comes to a complete stop. Thus the worst case time it takes for the system to react to a situation is given by:

$$T_{react} = T_{lat} + 2T_{cyc} + T_{cont} + T_{act} + T_{veh}$$

The coefficient of 2 arises from lack of synchronization between the sensor and the perception software. It is a worst case assumption. In reality, the coefficient of the software cycle time can be considered to vary randomly between 1 and 2 unless the cycle time is precisely constant.

Reaction time may be different for different actuators. In general, the steering and brake actuators may incorporate different delays. For this reason, two different reaction times are defined. The **braking reaction time** is called τ_B , and the **turning reaction time** is called τ_T .

1.4 Braking Reaction Time

According to the definition used here, the braking reaction time has elapsed after the brakes are fully engaged. That is, the time during which the vehicle decelerates is not included in this time. Therefore, an expression for the braking reaction time is:

$$\tau_B = T_{lat} + 2T_{cyc} + T_{cont} + T_{act}$$

where T_{act} is the small amount of time required for the brake actuator to move.

1.5 Turning Reaction Time

In the case of turning, the turning reaction time has elapsed after the steering mechanism reaches the commanded curvature. The time for which the vehicle moves at this curvature is not included. Without loss of generality, let the steering mechanism move through an angle $\Delta\alpha$ and let its maximum velocity be $\dot{\alpha}_{max}$. Then the actuator delay is given by:

$$T_{act} = \frac{\Delta\alpha}{\dot{\alpha}_{max}}$$

which can be as much as 3 seconds under some circumstances.

One of the most important aspects of the **latency problem** is the large value of the turning reaction time. For this reason, systems which attempt continuous motion in a dense obstacle field require excellent turning response characteristics.

1.6 Reaction Distance

In addition to considering the time required to react, the distance required can also be considered. This distance will turn out to be an important variable in later analysis, so it will be defined here as follows:

$$s_{react} = T_{react} V$$

This **reaction distance** gets its precise definition from the associated definition of the reaction time being considered. If it is important to be precise, the velocity is defined as the average velocity over the reaction time period.

Section 2: Maneuverability

In addition to its ability to react, the ability of a vehicle to maneuver to avoid obstacles or otherwise ensure safety is important. This section investigates the manner in which *computational reaction time and mechanical maneuverability together* determine the ability of a vehicle to avoid obstacles. The **actuation space** of a vehicle consists of a command vector with elements of steering, throttle, and brake - all of whose elements are time continuous functions. Exact analysis of vehicle maneuverability requires solution of the equations of vehicle dynamics while enforcing the constraint that the vehicle remain in contact with the terrain.

However, in order to simplify the analysis, four special hazard avoidance maneuvers will be defined. These maneuvers are the **panic stop**, the **turning stop**, the **impulse turn**, and the **reverse turn**. These maneuvers will be defined in an ideal sense for a point robot with actuators which respond instantaneously after a command is received. Actuator dynamics will be considered in a later section.

2A **Braking**

Once a command to stop is issued to the vehicle brakes, a certain amount of distance is travelled in order to exhaust the vehicle kinetic energy through friction between the wheels and the terrain. This is the **braking distance**. Simple analysis suggests that this distance is quadratic in vehicle velocity V . Let the vehicle mass be M , then its kinetic energy is $MV^2/2$. This energy must be removed through the work done by the frictional force. If the dynamic coefficient of friction is μ , and the distance travelled is s , and g is the local acceleration due to gravity, then this work is μMgs . Equating these gives:

$$\frac{MV^2}{2} = \mu Mgs \Rightarrow V^2 = 2\mu gs$$

2.2 Empirical Braking

An experiment was conducted¹⁰ which verifies this quadratic result. Fitting the data to the model generates a coefficient of friction from 0.5 to 1.0. The data are presented below:

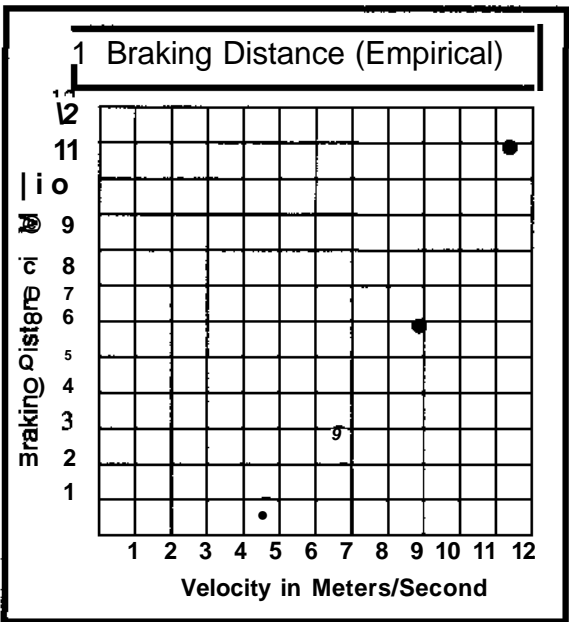


Figure 7 - Empirical Braking Distance

Table 1: Braking Distance

Speed	Stopping Distance
11.2 m/s	11 meters
8.9 m/s	6 meters
6.7 m/s	3 meters
4.5 m/s	1 meter

10. This graph is courtesy of R. Coulter of CMU based on an experiment conducted in the HMMWV.

2.3 Panic Stop Maneuver

Consider that the vehicle approaches an obstacle which it cannot avoid. That is, its only alternative is to stop as quickly as it can. This will be called a **panic stop maneuver**. The **stopping distance** is the distance travelled by the vehicle between the instant an object appears in the sensor field of view and the instant when the vehicle comes to a complete stop. If the vehicle velocity is constant and of magnitude V , then the **reaction distance** is the distance travelled before the brake is applied, or $[T_R] V$. The **braking distance** is the distance actually travelled while the brake is on. If the two are added, the following expression is obtained for the distance travelled by the vehicle:

$$s_B = \tau_B V + \left[\frac{V^2}{2\mu g} \right]$$

This relationship is plotted below for three values of reaction time using a conservative coefficient of friction of 0.5. Notice that stopping distance is quadratic in speed with a coefficient of about 0.1.

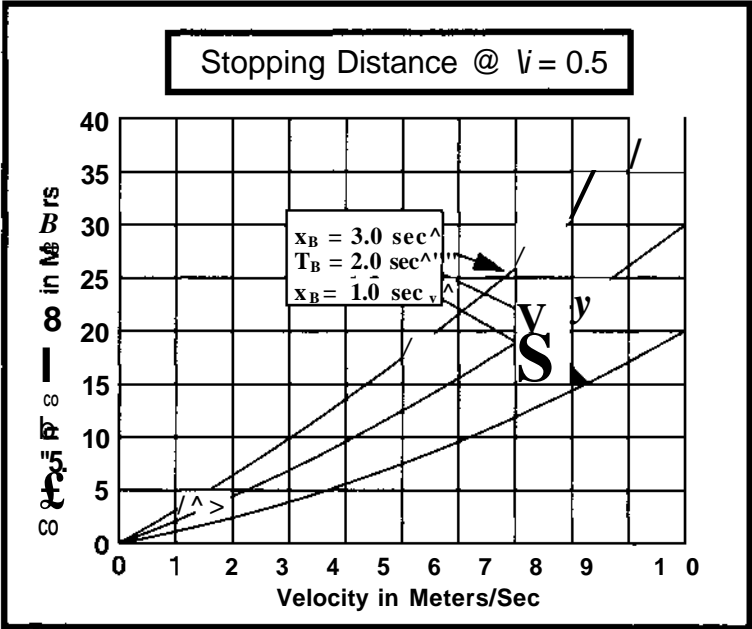


Figure 8 - Stopping Distance

2.4 Nondimensional Braking

A rudimentary dimensional analysis is performed by asking at what velocity of the vehicle do the **reaction distance** and the **braking distance** become equal. This happens when the ratio of the two distances is 1. The dimensions in the equation are distance and time. Hence there are two pi products. There are many ways to choose these because pi products are not unique. Later results will benefit from the following choice of variables.

The first is the **kinematic braking coefficient**:

$$b_k \equiv \frac{T_{\text{react}} V}{s}$$

which is the proportion of the stopping distance that is consumed by simply deciding to brake.

The second is the **dynamic braking coefficient**:

$$b_d \equiv \frac{V}{2\mu g T_{\text{react}}}$$

which is the ratio of the braking distance to the reaction distance.

Rewritten, the stopping distance expression takes the following forms:

$$b_k = \frac{1}{1 + b_d} \qquad b_d = \frac{1}{b_k} - 1$$

These expressions capture everything in Figure 8 in nondimensional form. Since both Pi products must satisfy this equation, either can be derived from the other. The dynamic braking coefficient will be considered to be fundamental and referred to simply as the **braking coefficient** 5. When the braking coefficient exceeds, say 0.1, it becomes important for a planning system to reason about the dynamics of braking.

Real Numbers

For the HMMWV at speeds of 5 meters/sec, with a 2 second reaction time, the braking coefficient is 0.25.

2.5 Braking Regimes

The braking coefficient identifies two key regimes of operation for autonomous vehicles. After substituting into the original expression, some algebra gives:

$$\frac{s_B}{\tau_B V} = 1 + \frac{V}{2\mu g \tau_B}$$
$$s_B = \tau_B V [1 + b]$$

When the braking coefficient is 1.0, braking distance and reaction distance are equal. At this point, stopping distance enters a regime of quadratic growth with initial velocity. Based on the braking coefficient, two regimes of operation can be defined. In the **kinematic braking regime** it is much less than 1.0. In the **dynamic braking regime** it is much greater than 1.0.

In the kinematic regime, stopping distance is linear in both initial velocity and reaction time. This is the reason why the curves in Figure 8 are basically linear even though they are given by a quadratic equation. In the dynamic regime, stopping distance is quadratic in initial velocity and independent of reaction time. In order to achieve a given target speed, *the path planning horizon can be reduced significantly if the system reaction time is reduced to a minimum.*

Real Numbers

For the HMMWV, with a 2 second reaction time, the braking coefficient is 1 at a velocity of 20 meters/second. Hence, the system will operate in the kinematic regime throughout the range of speeds considered in this document. Based on this single fact, many of the results which will follow can now be predicted based on intuition. Further, this explains why Figure 8 is basically linear although it is given by a quadratic equation.

2.6 Turning

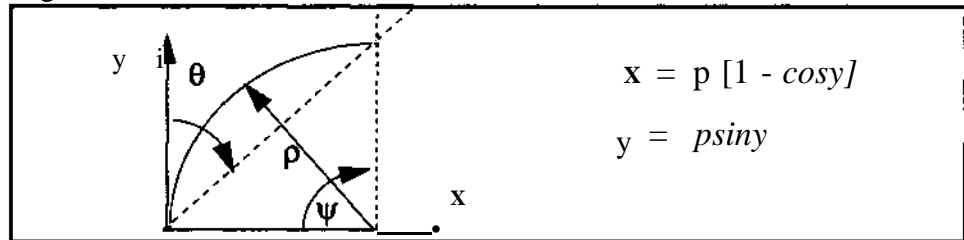
Consider a **constant curvature turn**. Let the vehicle yaw be given by ψ , the velocity be given by V , the curvature be given by K , and the radius of curvature be given by ρ . In a turning maneuver, the instantaneous vehicle **yaw rate** is given simply by the chain rule of differentiation:

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{d\psi}{ds} \frac{ds}{dt} = KV = \frac{V}{\rho}$$

If the time spent in the turn is T , the yaw of the vehicle after the turn is given by:

$$\psi = \frac{s_T}{\rho} = \frac{TV}{\rho}$$

Where s_T is the **turning distance**. When a constant curvature turn is executed, there is a simple relationship between the yaw of the vehicle and the angle subtended at the start point by the stop point. This is given below:



The range from the start point to the endpoint is given by:

$$R = \sqrt{x^2 + y^2} = \rho \sqrt{2(1 - \cos \psi)}$$

The angle from the start point to the endpoint is given by:

$$\theta = \tan^{-1} \left(\frac{y}{x} \right) = \tan^{-1} \left(\frac{\sin \psi}{1 - \cos \psi} \right) = \frac{\psi}{2}$$

This is a very useful result for determining the angular width of the region which is reachable in a given time at a given speed. **The angular width, subtended at the start point, of the region reachable by the vehicle in a turn, is the yaw of the turn itself**

If the vehicle decides to brake while in a turn, the **planning angle** is the angle travelled before the brakes are actuated. The **braking angle** is the angle travelled while the brakes are actuated before coming to a stop. The **stopping angle** is the sum of these two. In a manner analogous to the coefficient of friction, a **coefficient of lateral acceleration** can be defined thus:

$$\kappa = \frac{a_{\max}}{g} = \frac{V^2}{\rho g} = \frac{\kappa V^2}{g}$$

which is simply the lateral acceleration expressed in g's. The lateral acceleration at all points on a turning trajectory must be limited for safety reasons.

2.7 Steering Limits

There exists a maximum steering angle for any vehicle speed which will cause the vehicle to enter an unsafe state. This will be called the **dynamic steering limit** p_D . The radius of curvature of a vehicle may also be kinematically limited by the steering mechanism. This is the **kinematic steering limit** p_K . To determine the dynamic limits, let the maximum safe lateral acceleration be 0.5 g ¹¹. Let K be the path curvature, V be the velocity, and p be the radius of curvature. Then, the velocity and radius of curvature are related by:

$$\frac{V^2}{p}K = \frac{V^2}{p} < Vg \Rightarrow p_D = \frac{V^2}{Vg}$$

This relationship is plotted below for all vehicle speeds up to 10 m/s using a kinematic steering limit of 7.5 meters.

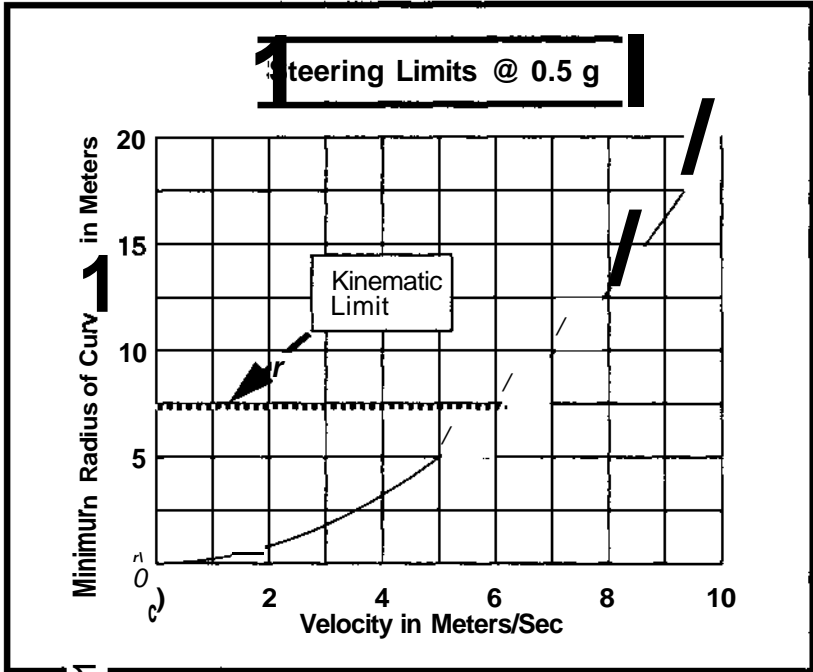


Figure 9 - Steering Limits

Hence, minimum radius of curvature is quadratic in speed with a coefficient of about 0.2. The minimum radius of curvature can be expressed by the function:

$$p_{min} = \max(p_K, p_D) = \max(p_K, \frac{V^2}{Vg})$$

The discontinuous derivative of this function will show up several times later.

11. This analysis is a gross approximation which is sufficient for the purpose it is used.

2.8 Turning Stop Maneuver

The **turning stop maneuver** occurs when the vehicle is executing a constant curvature turn and detects an obstacle in its path that is avoided by braking. The **stopping region** is the region of space in front of the vehicle which includes all possible braking trajectories. Clearly, this region is a function of the initial velocity and the initial curvature. It can be quantified by considering all initial curvatures at all speeds, and simulating forward along all trajectories until the vehicle comes to a stop. The stopping regions for two different speeds are plotted below.

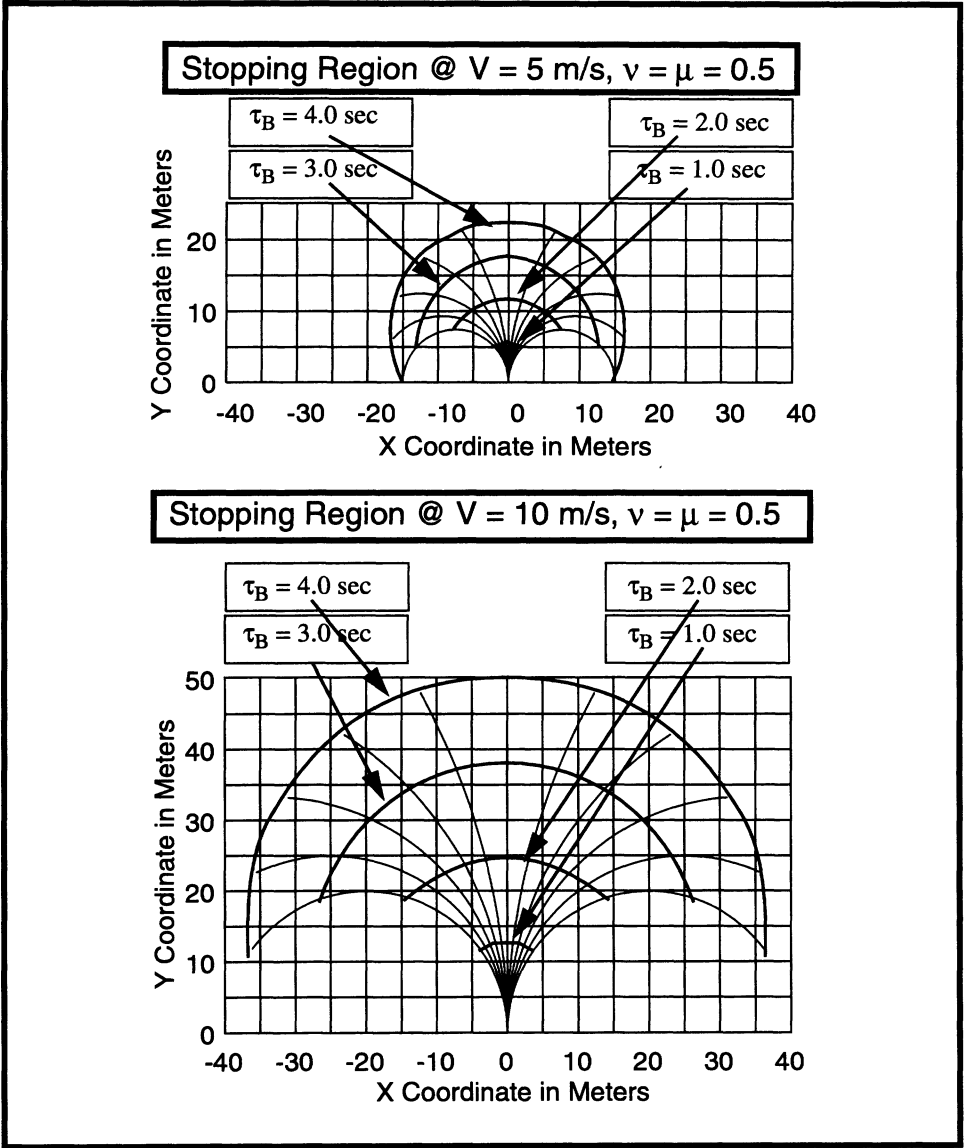


Figure 10 - Stopping Region

2.9 Impulse Turn Maneuver

Consider the case when the vehicle is executing a linear trajectory and decides to avoid an obstacle by executing a turn and not by hitting the brakes. This will be called an **impulse turn maneuver**. The vehicle will travel a distance given by the velocity and the reaction time before the steering mechanism is actuated. Then, the sharpest turn is determined by the kinematic and dynamic steering limits. Idealized turning trajectories for a point robot based upon a lateral acceleration limit of 0.5 g are plotted below¹².

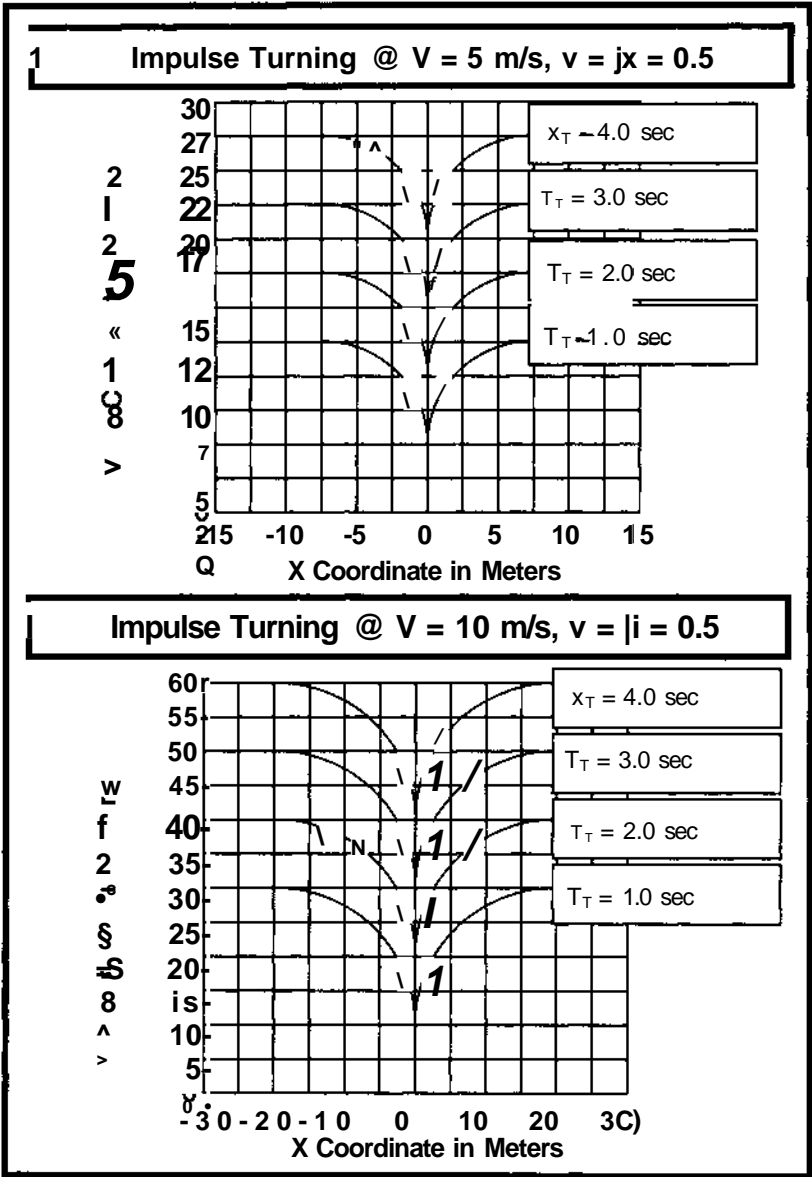


Figure 11 - Impulse Turning Trajectories

12. This analysis is a gross conservative approximation for several reasons. The dynamics of the steering mechanism are neglected even though it takes several seconds for a standard steering mechanism to reach full deflection. For this reason, reaction times applied to steering must be increased relative to braking. Also, the finite width of the vehicle must be considered. This can be achieved by further increasing the reaction time.

2.10 Impulse Turning Distance

In the impulse turn maneuver, notice the V-shaped region between the opposite turning trajectories. The width of this region is the width of the largest obstacle directly in the path of the vehicle that can be avoided¹³. This width becomes infinite after 90° of turn.

For the impulse turn maneuver, the **reaction distance** is the time taken for the steering mechanism to be actuated. The **turning distance** is the distance *along the original trajectory* consumed in turning. The **impulse turning distance** is the sum of these two.

An impulse turn of particular interest is the 90° impulse turn. This maneuver is required to avoid a large object without braking. The impulse turning distance is given by:

$$s_{IT} = \tau_T V + \rho_{min}$$

This can be written as:

$$s_{IT} = T_T V + \max(p_K, \frac{V^2}{vg})$$

The impulse turning distance is plotted below:

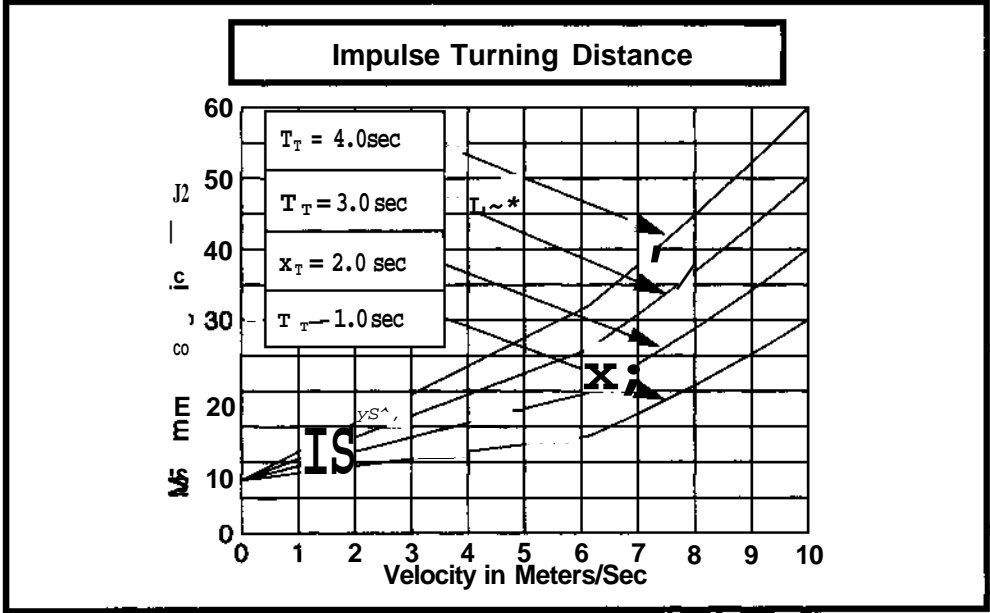


Figure 12 - Impulse Turning Distance

13. Remember that a point robot was assumed, so this V-shaped region should be moved back a few meters to account for this or the turning reaction time should be increased. On the other hand, point obstacles require substantially less turning than full deflection in order to avoid them which reduces the effective impulse turning distance. A detailed analysis is too extensive for the purposes of the document. The point is that large obstacles really do occur regularly, so this condition must drive sensor maximum range. Ideally, the offset of the vehicle nose from the sensor must be accounted for as well as the effect of vehicle motion on the scanning pattern.

2.11 Reverse Turn

Consider a situation in which the vehicle is already engaged in a sharp turn in one direction and decides to reverse curvature, and proceed in the opposite direction. This will be called a **reverse turn**. The vehicle will travel along a trajectory given by the velocity, the initial curvature, and the reaction time before the steering mechanism is actuated. Then, for an idealized trajectory, the curvature will instantaneously switch. Any dynamics in the steering response are accounted for in the turning reaction time. Idealized turning trajectories for a point robot, based upon a lateral acceleration limit of 0.5 g, are plotted below:

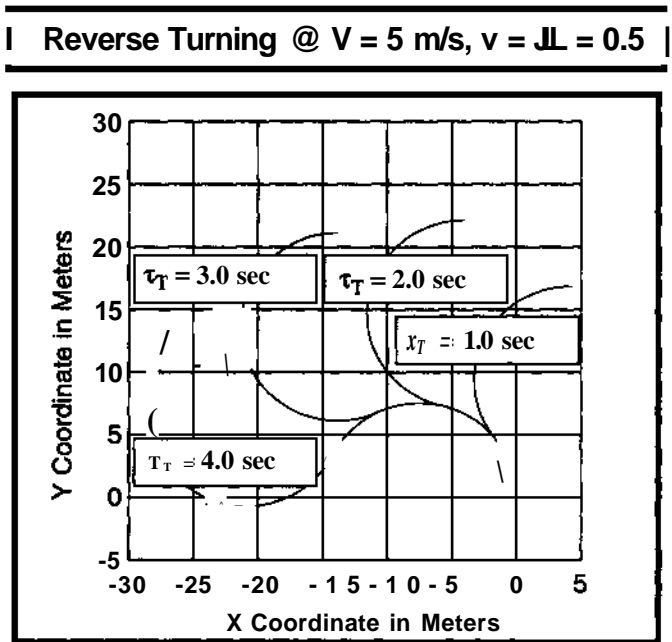


Figure 13 - Reverse Turning Trajectories

It is important to recognize that the vehicle will continue on its original curved trajectory in this model until the turning reaction time expires. This is why these trajectories deviate so significantly from the trajectory that would be followed if the reaction time were zero. The noninstantaneous response of the steering actuator is a major concern in the modelling of Ackerman-steered vehicles which travel at speeds exceeding even a few meters per second.

2.12 Stability Problem

The previous graph provides a basic reason for the stability problem. First of all, a kinematic model of steering response is fundamentally wrong when the product of speed and latency is large. This is one aspect of the **fidelity problem**. Further, hazard avoidance and goal-seeking behaviors which used such a model would go unstable as soon as a large turn was commanded. At even moderate speed, a kinematic model of steering gives rise to a **stability problem**.

2.13 Nondimensional Turning

The angle through which the vehicle turns in any constant curvature trajectory is given by:

$$\psi = \left(\frac{s}{\rho} \right)$$

If the vehicle were to execute a constant velocity turn for a time period of the reaction time at some radius of curvature, it would turn through an angle given by the arc length over the radius of curvature. Since it will be useful later, the inverse of this angle will be called the **kinematic turning coefficient** or simply the **turning coefficient**:

$$t_k = t = \frac{\rho}{VT_{\text{react}}}$$

This quantity can be defined for any velocity and any radius curve. It is the inverse of the amount that the vehicle turns over a period of one reaction time. Now a quantity analogous to the dynamic braking coefficient can be defined. Let the **dynamic turning coefficient** be given by:

$$t_d = \frac{V^2}{vg\rho}$$

It can be defined for any velocity and any radius curve. For any curve and velocity, this quantity measures the proximity of the vehicle to its lateral acceleration limit. As such, it is a measure of safety of a turn.

2.14 Turning Regimes

If the variation in the angular width of the stopping region is plotted against speed, it exhibits a maximum when the dynamic turning coefficient first reaches 1. Beyond this speed, the vehicle tends to become less omnidirectional as the dynamics of turning come into play. Hence there are turning regimes which are analogous to the braking regimes. In the **kinematic turning regime**, the dynamic turning coefficient is less than 1 and turns of the minimum kinematic radius are safe. In the **dynamic turning regime**, the dynamic turning coefficient is greater than or equal to 1.

In the dynamic regime, the effect of increased speed is to **reduce** the angular width of the stopping region. The vehicle approaches kinematic omnidirectionality when the turning coefficient significantly exceeds unity. It will be shown that, at any speed, the horizontal field of view required is given by the minimum value of the turning coefficient for any safe trajectory.

2.15 Turning Stop Maneuver

Suppose a vehicle is required to stop to avoid an obstacle while it is executing a turning trajectory. The angle that it will turn through before it stops is again given by:

$$\psi = \frac{s}{\rho}$$

Substituting for the stopping distance in terms of the braking coefficient:

$$V = \frac{t_B v [1 - 1.5]}{P} = \frac{[1 + 6]}{t}$$

This is a pleasingly simple result.

By analogy to the linear braking trajectory, three angles can be defined. The **planning angle** is the angle through which the vehicle turns while deciding to stop. The planning angle for a turning stop maneuver is the inverse of the turning coefficient.

The **braking angle** is the angle through which it turns while braking. This is given by the ratio of the braking coefficient and turning coefficients. The **stopping angle** is the sum of these two. For the turning stop trajectory, the ratio of the braking angle to the planning angle is the braking coefficient just as it was in the linear braking case.

Further, for a turning stop, the braking and turning coefficients are related as given below:

$$t_d = 2 \frac{\bar{b}}{t} \left(\frac{\mu}{v} \right) \qquad \bar{b}_k = \frac{1}{t} \left(\frac{\rho}{s} \right)$$

It has been shown that only one braking coefficient is independent. From this result, it is clear that there is only a single second independent variable - the ratio of stopping distance to radius of curvature. This is, of course, the **stopping angle** which was just defined.

For a turning stop trajectory, only the braking and turning coefficients are independent; all others can be derived from them. When the stopping distance is much greater than the minimum radius of curvature, the turning coefficient is significantly less than 1. Under these conditions, the vehicle can be considered to be kinematically omnidirectional at the resolution of the stopping distance.

2.16 Impulse Turn Maneuver

The impulse turning distance was defined as the reaction distance plus the minimum radius for the tightest possible turn. This is given by:

$$s_{IT} = \tau_T V + \rho_{min} = \tau_T V \left[1 + \frac{\rho_{min}}{\tau_T V} \right] = \tau_T V [1 + t]$$

2.17 Impulse Turning Regimes

This result is a perfect analogy to the stopping distance relationship. When the turning coefficient is greater than one, the second term dominates and growth is quadratic. More precisely, the nature of the growth of the impulse turning distance then depends on the dynamic turning coefficient. The different regimes are illustrated in Figure 12. At sufficient speeds, the quadratic growth of radius of curvature dominates the linear growth of the reaction distance. This occurs when:

$$\rho_{min} = \tau_T V = \frac{V^2}{g} \Rightarrow V = x_T Vg$$

Real Numbers

On the HMMWV, for a 2 second turning reaction time, this occurs at 10 m/s. For a 4 second reaction time, it occurs at 20 m/s. Unless reaction times can be reduced below 2 seconds (and this is not likely) the impulse turning distance can be considered linear in velocity.

Notice that the turning coefficient plays the same role for impulse turning that the braking coefficient plays for the panic stop. The dynamic turning coefficient plays a similar role for continuous turning.

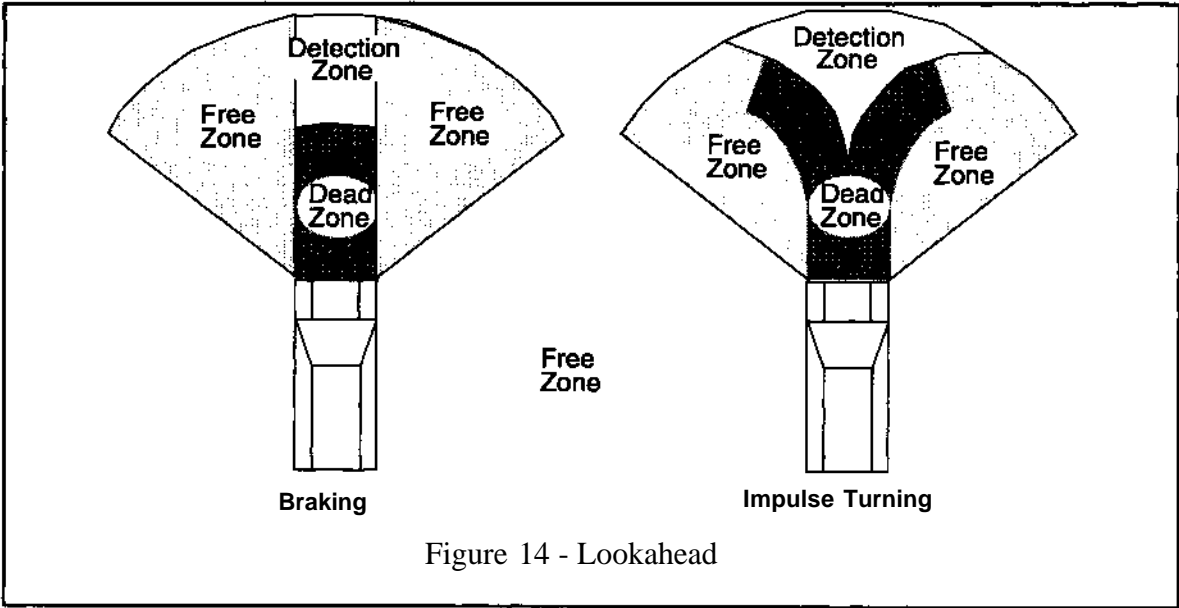
Section 3: Lookahead

The final element of the **response ratio** is the lookahead distance. The highest level requirement on perception is to image all of the terrain that the vehicle still has an option of traversing at any particular point in time. Ideally, a range sensor should provide geometric information *beyond* the entire region of terrain that can be reached by the vehicle before coming to a stop.

3.1 Adaptive Regard

From the perspective of turning and its impact on the horizontal field of view, it is clear from earlier sections that there are situations during turns when the entire left half or right half of the field of view need not be processed because the vehicle cannot go there. Further, from the perspective of braking and its impact on the vertical field of view, it is clear that there is some distance roughly given by the reaction time times the speed within which the vehicle cannot stop.

It is useful to partition the space in front of the vehicle into different zones based upon the different implications of an obstacle in each zone. Consider the following figure:



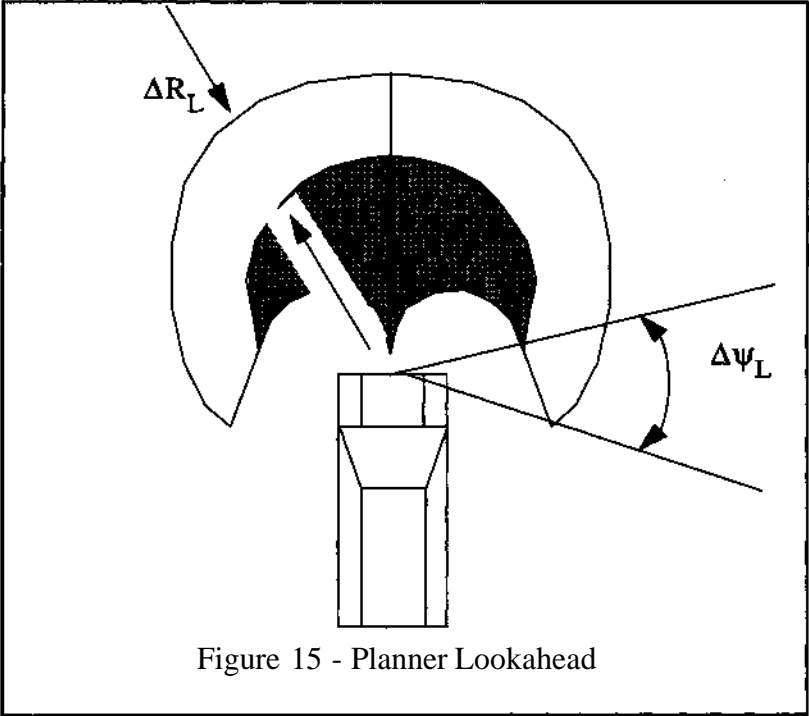
Regions are defined for each particular possible trajectory and for each type of avoidance maneuver. The **dead zone** is the region within the swath of the planned vehicle trajectory which the vehicle is committed to travelling. *Should an obstacle ever enter this region, a collision is unavoidable.* The **detection zone** is the region along the trajectory which can still be avoided. **Free zones** are areas in neither of the other two. These are regions where the vehicle cannot go because of its steering and braking limitations.

Since an obstacle in the dead zone cannot be avoided, there is no reason to allocate precious range pixels there. Rather, obstacles must be detected in the detection zone before they ever reach this region. As a minimum, a range sensor must allocate range pixels over the region formed by the union of the detection zones of all possible trajectories for all possible avoidance maneuvers. The dead zone grows as vehicle speed increases. This is the mechanism which drives the sensor field of view away from the vehicle with increased velocity.

In the context of planning, these notions are important and the dead and free zones are a very large fraction of the total. There is no justification for wasting perceptual and planning cycles in discovering the geometry of regions in these categories because *there is no useful decision that the planner can make*¹⁴. This notion will be called **adaptive regard**. The principle of adaptive regard is to process geometry only in the detection zone. In practice, this means that *a system must adapt in real time to both its speed and its curvature*,

3.2 Pointing Rules

In order to detect obstacles beyond the dead zone, a range sensor must supply range pixels in the region beyond the dead zone, called the detection zone. The width of the detection region used is called the **incremental lookahead distance** and the angular width beyond the dead zone is called the **incremental lookahead angle**. The planner lookahead required for a turning stop maneuver is indicated in the following figure:



The lookahead distance must be chosen such that obstacles can be detected beyond the dead zone of all possible avoidance maneuvers. There are two considerations involved in choosing this distance.

- In order to compute a reliable prediction of vehicle pitch, it is necessary for the sensor field of view to extend for at least one vehicle wheelbase beyond the dead zone.
- Reliable small obstacle detection may require that the sensor field of view be such that several images fall on an obstacle before it leaves the detection zone. On this basis, the planner lookahead should exceed the dead zone by several times the product of the velocity and the sensor frame period.

In this thesis, the vehicle wheelbase will be chosen as the incremental lookahead distance.

14. One could argue that it is useful to hit an obstacle at reduced speed. Thanks to Todd Litwin of JPL for pointing this out.

3.3 Adaptive Lookahead

Adaptive regard is a notion defined for path planning purposes. It restricts the region that an obstacle detector considers to the detection zone for some set of avoidance maneuvers. A direct implementation of the pointing rules for the purposes of guaranteeing response will be called **adaptive lookahead**. Adaptive lookahead is a notion defined for perception purposes. It restricts the focus of attention of perception to the detection zone. Practically, this mechanism computes the minimum range or maximum range based on the obstacle avoidance maneuver in question.

3.4 Nondimensional Lookahead

For the purpose of providing information to the path planner, the perception system must look beyond the dead zone. The amount of lookahead is called the **incremental lookahead distance** ΔR_L . Similarly, the maximum sensor horizontal field of view is greater than the stopping angle by an amount called the **incremental lookahead angle** $\Delta \psi_L$. Based on these, two convenient nondimensionals can be defined.

The **normalized incremental lookahead distance** is the lookahead distance normalized by reaction distance.

$$\Delta \bar{R}_L = \frac{\Delta R_L}{V T_{\text{react}}}$$

It is related to the **throughput ratio** defined earlier. The **incremental lookahead angle** is already nondimensional. It is the incremental lookahead distance divided by the instantaneous radius of curvature

$$\Delta \psi_L = \frac{\Delta R_L}{\rho}$$

Let the ratio of maximum to minimum range be called the **range ratio**:

$$\bar{R} = \frac{R_{\text{max}}}{R_{\text{min}}}$$

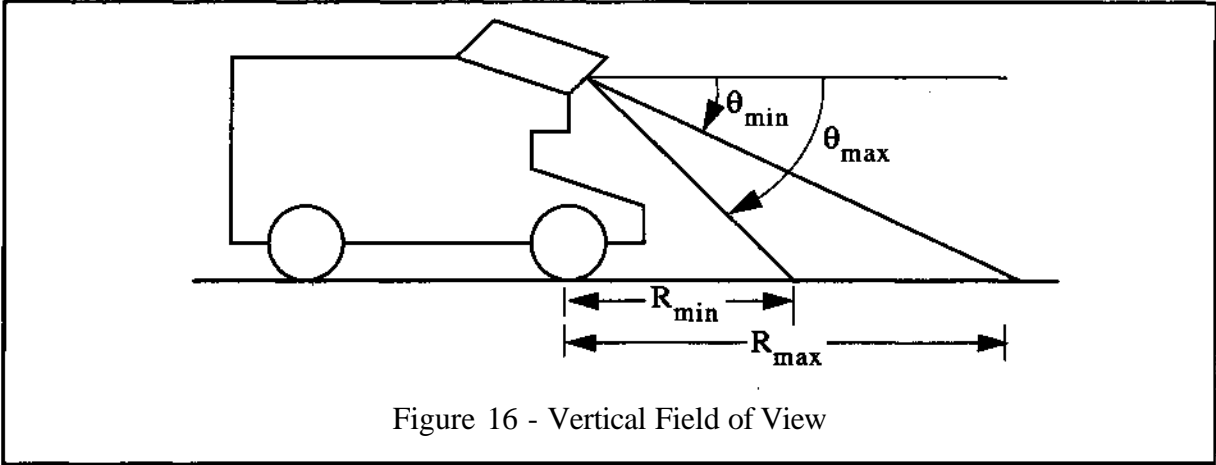
A quantity that will be important later is the ratio of the distance travelling while thinking about stopping to the distance required to stop. This will be called the **lookahead ratio** because its inverse measures the number of system cycles that the system is looking ahead:

$$\rho_R = \frac{V T_{\text{cyc}}}{R_{\text{max}}}$$

The lookahead ratio relates throughput and response and will be important later in assessing the processing requirements of mobility.

3.5 Nondimensional Pointing Rules

The pointing rules relate the sensor field of view to the vehicle maneuverability at any speed. The vertical field of view is shown below:



The pointing rules that will be used to calculate the field of regard for planning purposes can be expressed as follows:

$R_{min} \approx s_B$	stopping distance
$R_{max} = s_{IT} + L$	impulse turning distance + lookahead
$\theta_{max} = \frac{s_T + L}{R_{min}}$	stopping angle + lookahead

Chapter 3: Throughput

As was shown earlier, the **throughput ratio** relates the ability of the vehicle to process information at a sufficient rate to its speed and its incremental sensory lookahead. Throughput is a completely independent matter from response. This section analyzes these aspects of vehicle performance for typical vehicles.

Section 1; Depth of Field

This section and the following two sections investigate the relationship between the maneuverability of the vehicle and the sensor field of view. The sensor field of view is considered to be a solid cone of rectangular cross section which can be described by the horizontal field of view, vertical field of view, minimum range, and maximum range.

1.1 Minimum Sensor Range

The minimum range required of a sensor as a function of velocity is given by the closest point of any detection zone for any trajectory and any avoidance maneuver. Of the four avoidance maneuvers considered, the turning stop is the one which remains closest to the vehicle start point. The following figure was generated by tracing the endpoint of all turning stop trajectories, and computing the minimum range to each endpoint.

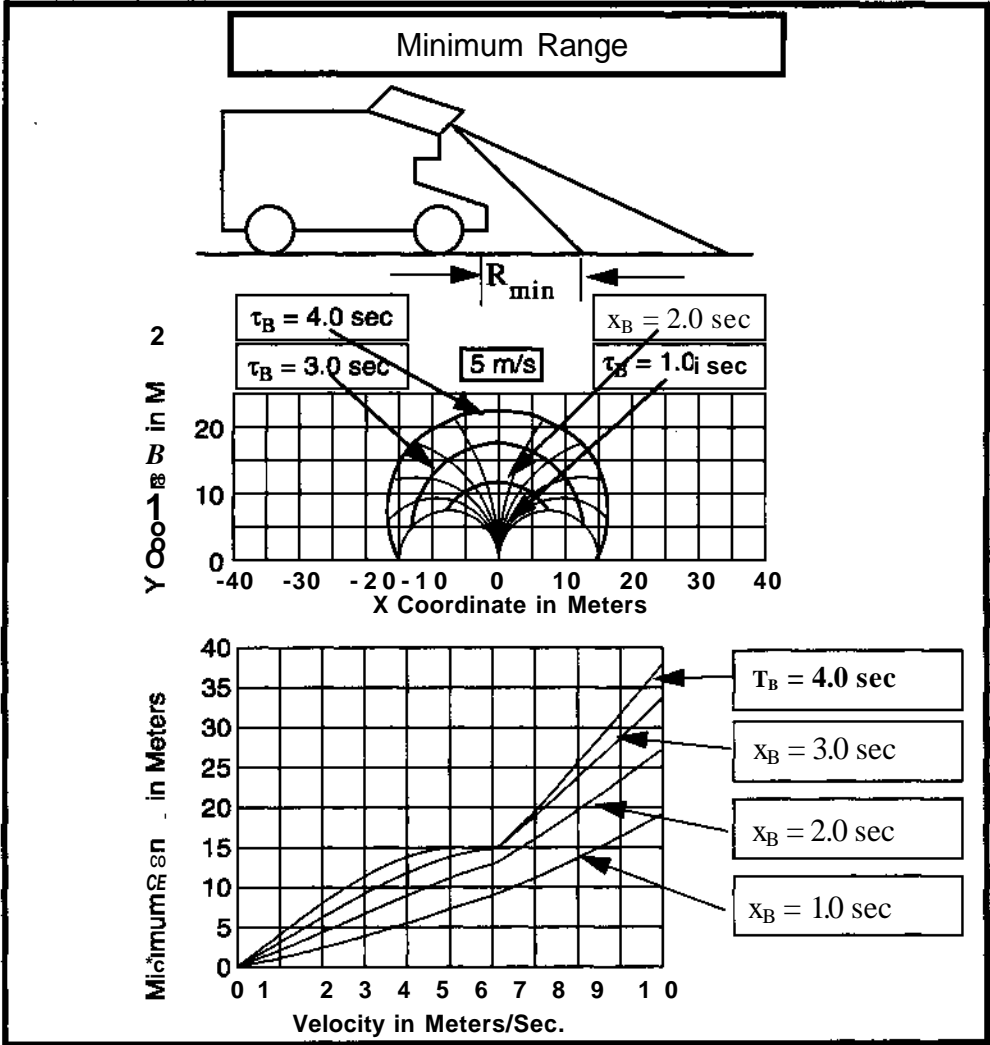


Figure 17 - Minimum Range

The peculiar knee in the higher curves arises from the discontinuous derivative of minimum radius of curvature with velocity. Notice that at the knee, the minimum is twice the kinematic minimum radius. By comparing this figure with Figure 8, it is clear that the stopping distance overestimates the minimum range slightly, but it is a good approximation.

The minimum range is a loose requirement. There is no cost to response incurred by increasing the minimum range. It can be increased up to the point where throughput is barely guaranteed.

1.2 Maximum Sensor Range

Turning is preferred over braking for obstacle avoidance, so it is reasonable to let the impulse turning maneuver drive the specification of maximum range even though this maneuver requires much more space. The sensor maximum range is given by the sum of the impulse turning distance and the planner lookahead distance. The maximum range derived from this condition is given in the figure below¹⁵. Again, the discontinuity arises because of the discontinuity in the derivative of the radius of curvature.

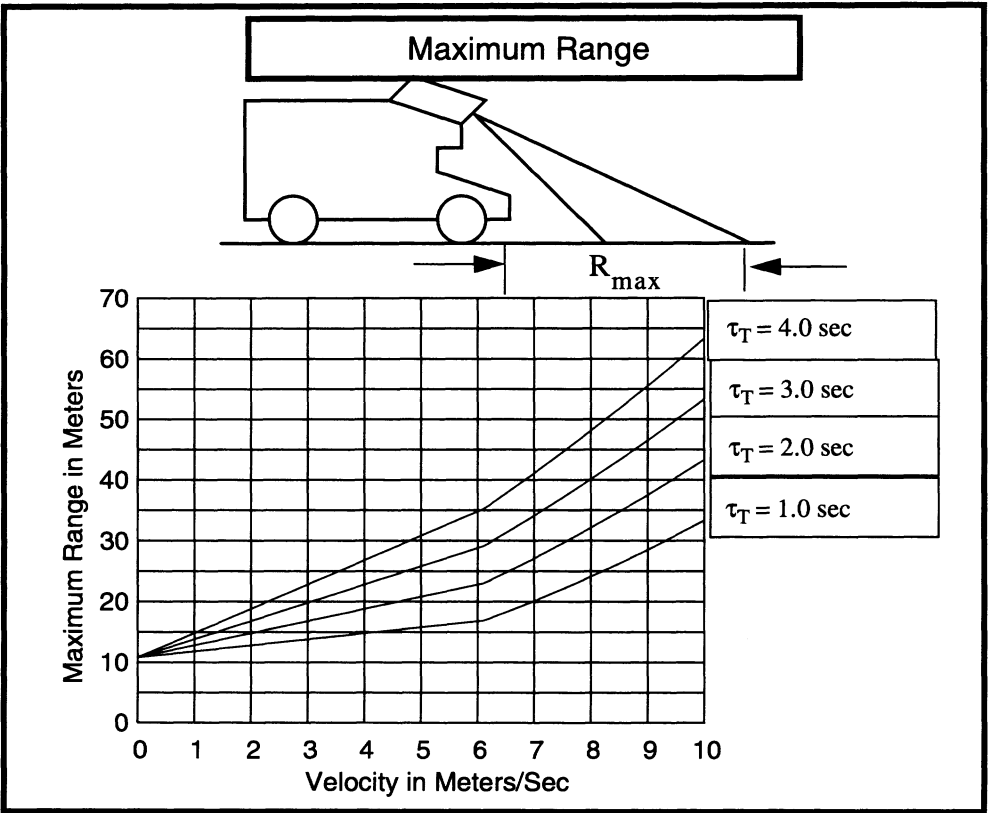


Figure 18 - Maximum Range

For a single scanline sensor, the line must be placed at or beyond the maximum range computed above. Such sensors rely on very good obstacle detection.

Real Numbers

The maximum range of the ERIM scanner is 20 meters. Reaction times below 3 seconds for full deflection turns are probably not realistic. On the basis of this curve, the ERIM sensor supports no more than 3 m/s speeds while avoiding wide obstacles. This estimate incorporates no safety margin.

15. This graph is based on a very ambitious lateral acceleration limit and a small lookahead of the wheelbase. For this reason, a practical maximum range may be far larger than indicated by the graph. No numbers are available for realistic lateral acceleration limits.

1.3 Nondimensional Maximum and Minimum Range

According to the pointing rules, the sensor minimum range is given by the stopping distance:

$$R_{\min} = s_B = \tau_B V [1 + \bar{b}]$$

Also, the sensor maximum range is given by the impulse turning distance plus the lookahead distance. This gives:

$$R_{\max} = s_{IT} + L = \tau_T V [1 + \bar{t}] + \Delta R_L$$

Dividing gives an expression for the range ratio in terms of the vehicle maneuverability:

$$\frac{R_{\max}}{R_{\min}} = \frac{\tau_B V [1 + \bar{b}]}{\tau_T V [1 + \bar{t} + \Delta R_L]} = \frac{\tau_B [1 + \bar{b}]}{\tau_T [1 + \bar{t} + \Delta R_L]}$$

1.4 Myopia Problem

The obstacle resolving power of contemporary stereo and lidar systems is poor at the high ranges necessary to resolve obstacles at speed and *a system must attempt to succeed with a poor idea of what is out there*. The effective maximum range of a contemporary environmental perception sensor is limited by any of three concerns: first, angular resolution is often poor; second, signal-to-noise ratios degrade with increased range and with shallow incidence angles; third, occlusion is aggravated by the inherently shallow incidence of range pixels which arises from long range measurement.

Section 2: Horizontal Field of View (HFOV)

The horizontal field of view will be determined by the turning stop maneuver. It can be argued that the **reverse turn** is a worst case, but this would drive the field of view to unreasonable size. It is reasonable to assume that stopping is the only option used when executing a tight turn.

The angle subtended at the vehicle by the dead zone is the maximum angle subtended at the vehicle by the stopping region for all vehicle speeds, since, in the worst case, the vehicle executes a continuous turn into new terrain. Recalling the 5 m/s stopping region, the angles can be read from the graphs of the stopping region for each value of system reaction time. The horizontal field of view is determined by the planner lookahead which is added to the angle subtended by the dead zone. The HFOV is derived by adding a lookahead distance of the vehicle wheelbase to the braking trajectory. In this way, a planner could detect and respond to a roll or pitch obstacle by braking.

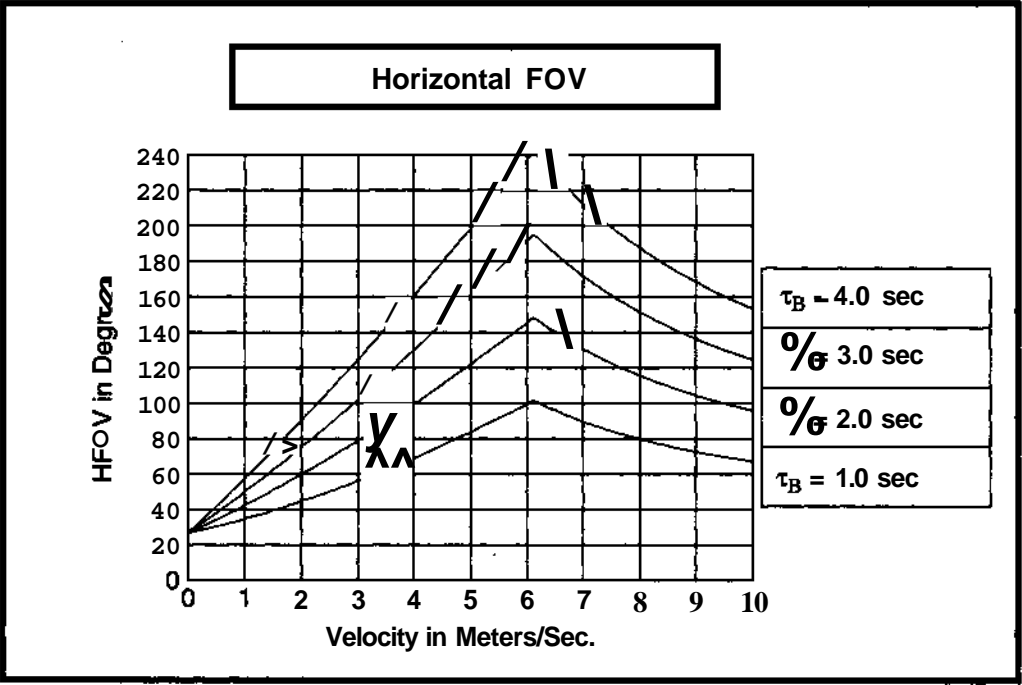


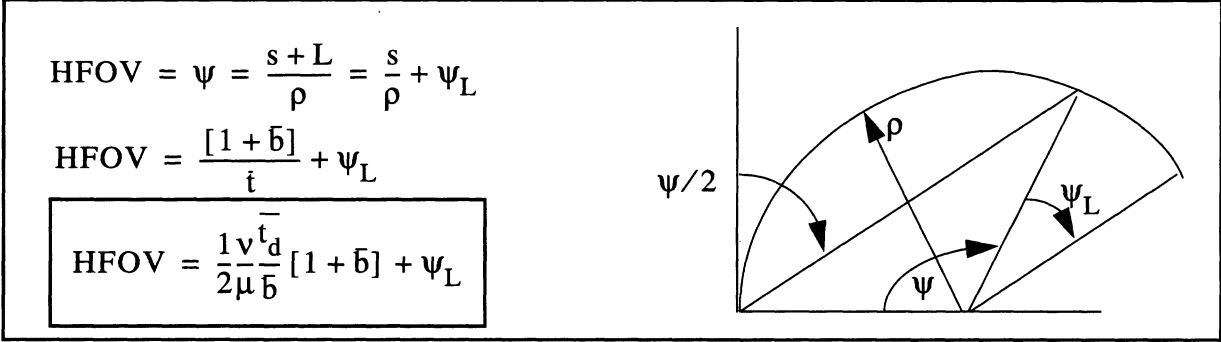
Figure 19 - Horizontal Field of View

The reaction times applicable to the turning stop are braking reaction times. Notice that the maximum is achieved at the fastest speed for which the smallest radius of curvature arc can be executed. From the previous graphs, this is 6.0 m/s. Beyond this curvature, the required HFOV actually *decreases* because turns of the minimum radius are no longer possible.

A HFOV beyond 180° seems counterintuitive. This is because the analysis assumes that information from previous images cannot be counted upon to provide information. The validity of this assumption depends on the frame rate, the scanning pattern, and the vehicle speed. More importantly, it depends on whether there are any occlusions which the sensor must see around. To adopt a HFOV smaller than the worst case requirement is to artificially reduce the maneuverability of the vehicle. This may be acceptable in some cases.

2.1 Nondimensional Horizontal Field of View

In dimensional analysis, ratios of important lengths often turn out to be important angles. As speeds increase beyond the maximum speed for which turns of the minimum radius of curvature are safe, curvature must be decreased to avoid excessive lateral acceleration. Consider the following figure:



This relationship gives the variation of the required HFOV with velocity for any vehicle. This is a very pleasing result. It states the intuition that both the hardness of the turn and the stopping distance should determine the result. In the kinematic braking regime, the result is linearly increasing with velocity. In the dynamic braking regime, it decreases quadratically. So the formula is consistent with the graphs in Figure 19.

Real Numbers

For the HMMWV, executing a minimum radius of curvature turn, with a reaction time of 2 secs, and 6 meters/sec speed, the formula gives:

$$\text{HFOV} = \left[\frac{T_{\text{react}} V}{\rho} \right] \left[1 + \frac{V}{2 T_{\text{react}} \mu g} \right] + \frac{L}{\rho} = \frac{2V}{7.5} \left[1 + \frac{V}{20} \right] + \frac{3.3}{7.5} = 2.52 \text{rads}$$

which is, from Figure 19, exactly correct.

Real Numbers

The maximum in HFOV is reached at 6 m/s for the HMMWV. Clearly, this must be the velocity at which the dynamic turning coefficient is 1.

$$\bar{t}_d = \frac{v^2}{vg\rho} = \frac{6 \times 6}{\frac{1}{2} \times 10 \times 7.5} = \frac{72}{75}$$

The curious growth of HFOV followed by a decrease with velocity is unique to non omnidirectional vehicles. The minimum radius of curvature is zero for truly omnidirectional vehicles. They operate solely in the dynamic turning regime and the angular width of the stopping region is determined solely by the lateral acceleration limit.

2.2 Tbnnel Vision Problem

Contemporary sensors have horizontal image sizes which are far too small to permit aggressive obstacle avoidance maneuvers on rough terrain. The analysis suggests that a HFOV of 120° is a reasonable engineering "guesstimate". Yet, typical stereo field of view is 40° and the best rangefinder has a HFOV of 80°. Consider the following figure in which the vehicle is executing a reverse turn. Turning dynamics imply that the entire region that the vehicle can reach is contained within the set of curves shown. Aggressive maneuvers cause collision simply because a contemporary system cannot look where it is going. In fact, in the reverse turn indicated below, there is no overlap at all between the field of view and the region that the vehicle is committed to travelling.

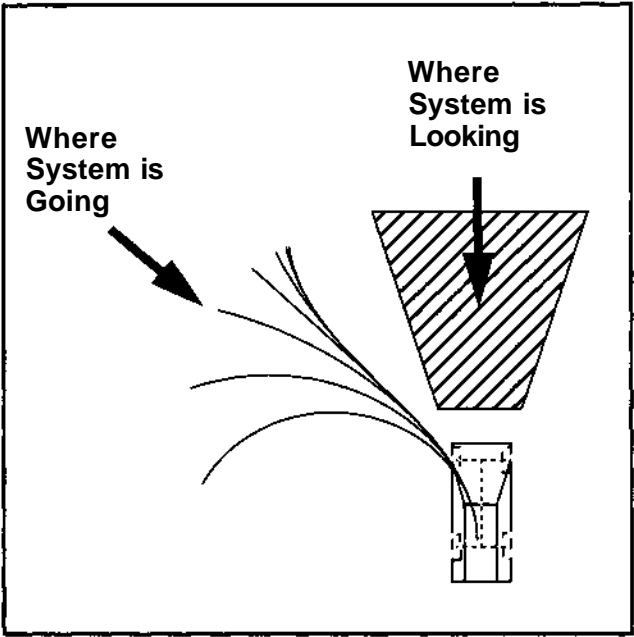


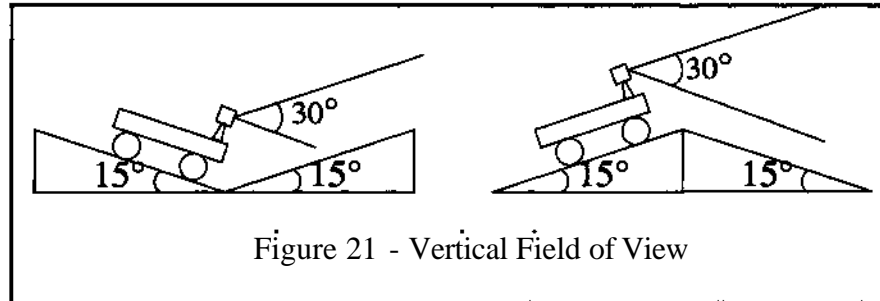
Figure 20 - Tunnel Vision Problem

This consideration argues strongly for mechanisms which physically point the narrow field of view sensor - at least in azimuth. For non-pointable sensors, overall latency severely complicates this problem. If the vehicle turns with angular velocity $\dot{\psi}$ and the horizontal field of view is small, it is not unusual for the vehicle to have driven completely off of the imaged terrain by the time that the data is processed. If the overall system reaction time is T_{react} , then by the time that a command reaches the hardware, the vehicle has turned through an angle:

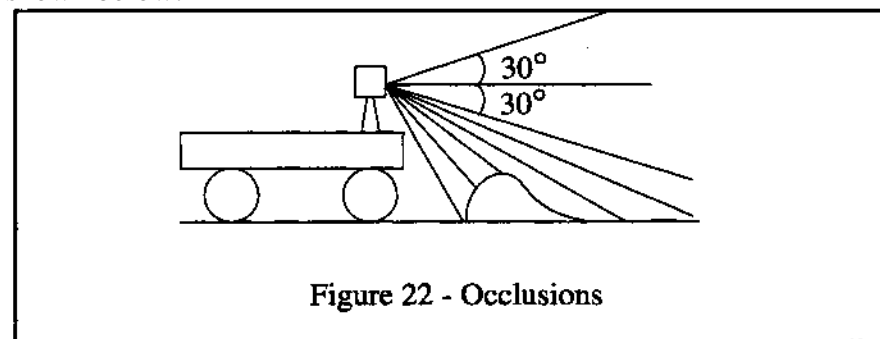
$\Delta\psi = \dot{\psi} T_{react}$
Real Numbers
For the HMMWV - with a 2 sec braking reaction time and an angular velocity of 6 meters/sec linear velocity divided by a turn radius of 5 meters, this angle is roughly 2 radians or 120 °. Clearly this problem is a very severe one.

Section 3: Vertical Field of View (VFOV)

The major *kinematic* system requirement which influences the vertical field of view is the pitch angle induced in the vehicle body by the most challenging, yet navigable, terrain. Later sections will develop *dynamic* requirements on the VFOV. Let the highest achievable pitch angle be 15° . Then the following figure illustrates the two cases which determine the vertical field of view required to ensure that the vehicle is able to see up an approaching hill or past a hill that it is cresting.



Therefore, the vertical field of view required is four times the maximum pitch of the body. It would also be useful to extend the lower half of the field of view to allow the vehicle to see behind occlusions as shown below:

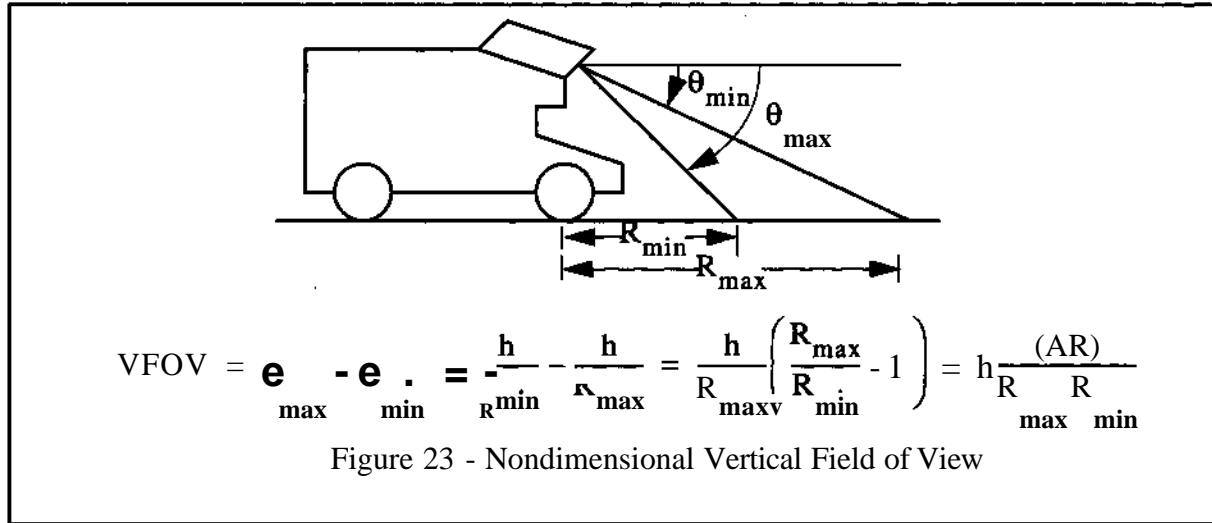


An extreme disadvantage of measuring range pixels above the horizontal is that most of the time they measure nothing. These pixels come into play only when the vehicle is executing challenging maneuvers. For this reason, it is advantageous to reduce the angle above the horizon as much as possible.

Another reason to reduce the VFOV above the horizon is that the left case in Figure 21 can be detected by a reduction in the highest measured range, so mechanisms to point the field of view or slow the vehicle are viable alternatives. In the right case, the highest measured range will likewise decrease, but slowing the vehicle does not deal with the problem, so sensor pointing is the only solution. Hence, it is better to allocate margin on the lower half of the vertical field of view.

3.1 Nondimensional Vertical Field of View

It is useful to assume that the height of the sensor is always less than the range measured. This is the **small incidence angle assumption**. The vertical field of view can then be expressed as follows:



When the sensor height is significantly less than the minimum range (and therefore the stopping distance), the vertical field of view can be expressed in terms of minimum value of the perception ratio and the range ratio.

$$VFOV = h \frac{(R_{\max} \cdot R_{\min})}{R_{\max} R_{\min}}$$

$$VFOV = \frac{h}{R_{\min}} (R - 1) - K_{mi0} * 5$$

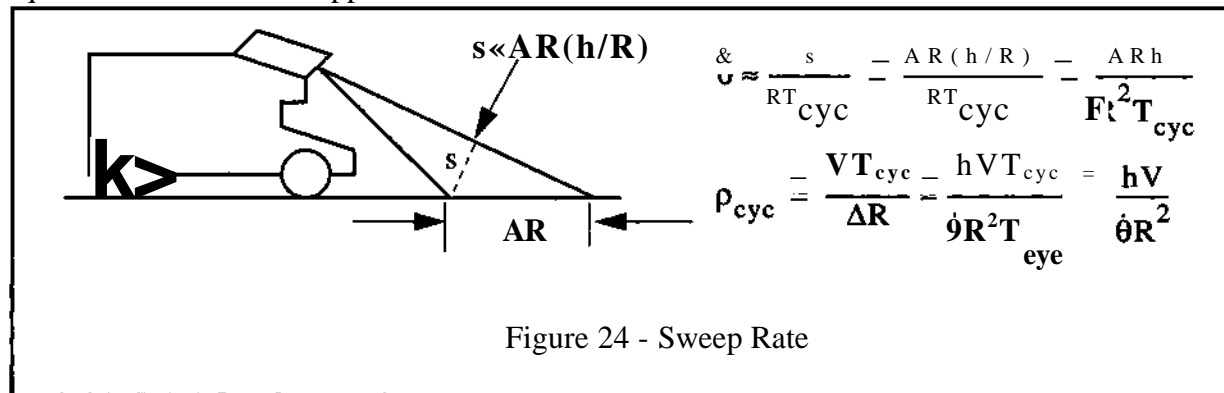
This relationship gives the variation in required VFOV with speed for any vehicle. Substituting the pointing rules permits elimination of the range ratio and expressing the result in terms of the maneuverability coefficients:

$$VFOV = \frac{h}{R_{\min}} - \frac{h}{R_{\max}} = h \left[\frac{1}{\tau_B V [1 + b]} + \frac{i i}{\tau_T V [1 + t + R_L] J} \right]$$

Therefore, in kinematic regimes, the VFOV decreases linearly with velocity. In dynamic regimes, it **decreases quadratically with velocity**. This result will be of extreme importance later.

Section 4: Sweep Rate

The **sweep rate** of a sensor is defined in image space as the VFOV generated per unit time. It may represent the physical motion of the elevation mirror in a rangefinder or the product of the VFOV and the frame rate for a camera. The sweep rate required of a perception sensor is related to the velocity of the vehicle under guaranteed throughput. If $\dot{\theta}$ is the sensor sweep rate, the requirement on it can be approximated as follows:



Thus, insisting that the throughput ratio never exceed unity, leads to the equivalent requirement that the sweep rate always exceed:

$$\dot{\theta} \geq \frac{hV}{R^2}$$

which is pleasingly simple. This gives the minimum sweep rate, or equivalently the minimum vertical field of view for some frame rate which guarantees that there are no holes in the coverage of the sensor. This is the linear velocity component of the **sweep rate rule**. Of course, if maximum range is related back to stopping distance, the sweep rate can be expressed solely in terms of reaction time, sensor height, and velocity - making it a function only of vehicle parameters.

Equivalently, on rough terrain, the vehicle may pitch as a result of terrain following loads, and in the worst case, these motions add to the sweep rate requirement. If $\dot{\theta}_{max}$ is the maximum pitch rate of the vehicle, then the sweep rate rule becomes simply:

$$\dot{\theta} = \dot{\theta}_{max} + \frac{hV}{R^2}$$

which relates the sensor field of view to both the velocity and the vehicle natural frequency. Thus rough terrain affects throughput to the degree that vehicle natural frequency may affect the pitch of the body.

4.1 Stabilization Problem

Notice that the linear component of the rule benefits from higher speeds whereas the angular component suffers. For the vertical field of view, the process of decreasing it in order to reduce throughput requirements will eventually lead to the creation of a **stabilization problem**. The following figure indicates that rough terrain can create severe stabilization constraints for a narrow VFOV sensor. However, it will also be shown later that *a wide VFOV can be difficult to achieve* because it aggravates the **throughput problem**.

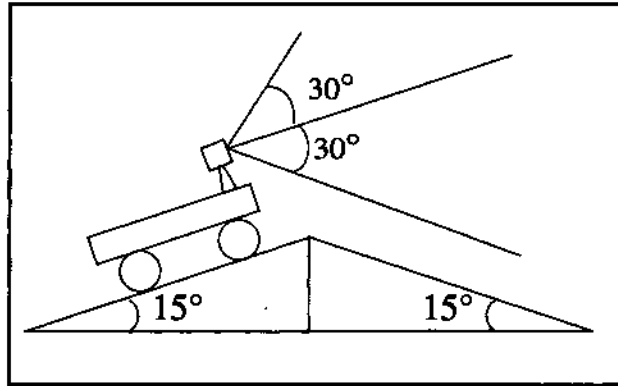


Figure 25 - Stabilization Problem

4.2 Nondimensional Sweep Rate

The **imaging density** c_I is the average number of images that fall on any patch of terrain. It is also the inverse of the **throughput ratio**. The expression of the sweep rate rule can be written in terms of the imaging density as:

$$\dot{\theta}_{eye} = \frac{A R f_{images}}{\sigma_I V}$$

Reusing the result for vertical field of view from the pointing rules:

$$\frac{VFOV(R_{max} - R_{min})}{h} f_{images} \sim \frac{\sigma_I V}{I}$$

This can be written in terms of nondimensionals as:

$$\frac{VFOV \times R_{max}}{\left(\frac{h}{R_{max}}\right) \left(\frac{R_{max}}{R_{min}}\right)} f_{images} = \sigma_I V$$

The **sweep rate** is therefore given by:

$$\dot{\theta} = VFOV \times f_{images} = \frac{\sigma_I V h}{R_{max} R_{min}}$$

So, for a fixed angular field of view, the rule says that a constant imaging density is achieved by modulating the sweep rate by roughly the inverse of the square of the range. For any speed, there is some minimum sweep rate of the beam. Kinematic arguments were given earlier which constrain the vertical field of view based on vehicle dimensions, maneuverability, and the terrain. This rule specifies a **dynamic constraint** as well.

4.3 Adaptive Sweep

There is no fundamental requirement that an autonomous system have an arbitrarily large vertical field of view. The line scanner has been an important existence proof of this fact. It is possible to employ adaptive techniques that either physically point a narrow VFOV sensor or which computationally stabilize a wide VFOV sensor by processing a small portion of each image. This idea will be called **adaptive sweep**.

In an earlier section, **adaptive lookahead** was proposed as a mechanism for moving the position of the projection of the vertical field of view on the groundplane in order to guarantee response. While this mechanism addresses the position of the sweep, adaptive sweep addresses the width of the sweep. These two aspects of the sweep are related through the velocity.

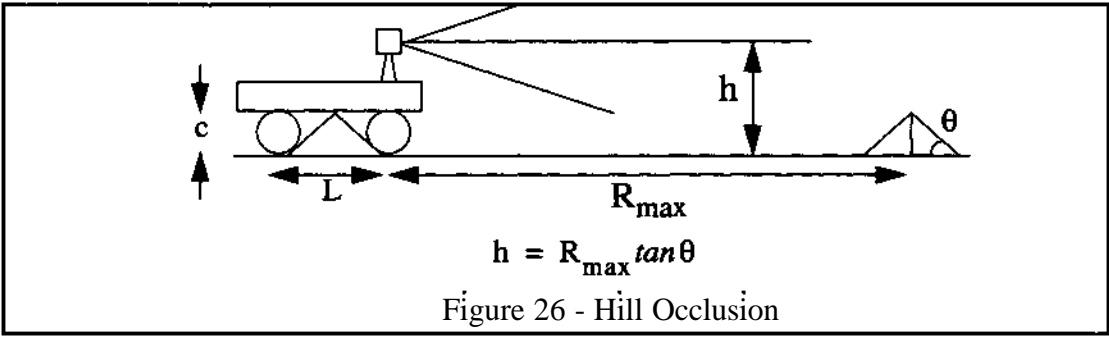
Section 5: Occlusion

The field of view analysis conducted so far is based on a **flat world assumption** and it is not entirely correct on rough terrain. This section investigates the relationship between vehicle configuration and the prevalence of terrain self-occlusions. It turns out that merely mounting a sensor on the roof of a vehicle implies, for typical geometry, that terrain self-occlusions are inevitable, and that holes cannot be detected until it is too late to react to them.

These are two aspects of the **occlusion problem**. Any system which does not deal with occlusions or which attempts to deal with hole obstacles may be designed suboptimally. There is an additional problem that the sampling problem is itself aggravated by rough terrain. There is no getting around the fact that an optical sensor cannot see through hills so a rough terrain system must be designed to live gracefully with this problem.

5A Hill Occlusion

A hill can also be called a **positive obstacle**. Ideally, a sensor could see behind a navigable hill at the maximum sensor range. The required sensor height can be derived from this requirement. The highest terrain gradient which is just small enough to avoid body collision is determined by the vehicle undercarriage tangent as shown below.



In order for occlusions of navigable terrain to be completely eliminated, the following condition must be met:

$$\left(\frac{h}{R} \right) = \left(\frac{c}{L/2} \right)$$

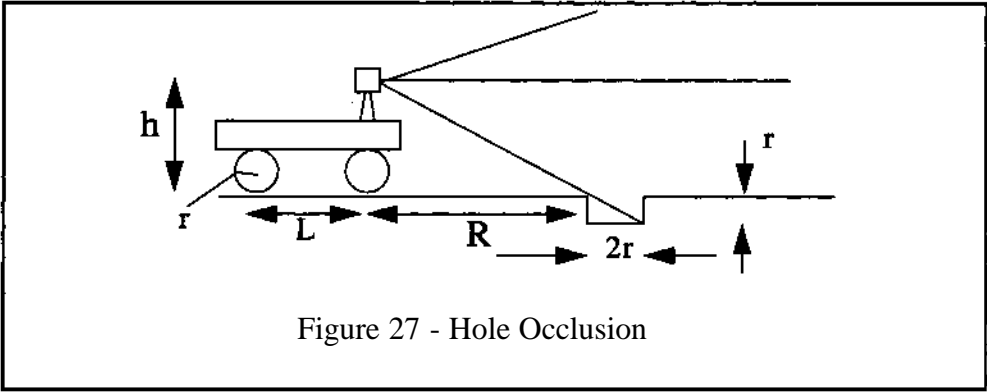
So, for complete avoidance of occlusion of navigable terrain, the ratio of sensor height to maximum range must equal or exceed half the undercarriage tangent. This will be called the **hill occlusion rule**. In order to satisfy the hill occlusion rule, a sensor must be mounted at a fantastic height which is not realistically achievable without taking very special measures. The hill occlusion rule is almost always violated. The **perception ratio**, h/R , can easily exceed the undercarriage tangent by a factor of three or four. Hence, *occlusions of navigable terrain are common when the terrain is rough.*

Real Numbers

For the HMMWV, the undercarriage tangent is about 1/3. For a maximum range of 50 m, this condition requires a sensor height of 16.6 m.

5.2 Hole Occlusion

A hole can also be called a **negative obstacle**. Such obstacles are particularly problematic to an autonomous vehicle. Consider a hole which is roughly the same diameter as a wheel and which is as deep as a wheel radius. Such a hole is roughly the smallest size which presents a hazard.



In order to detect that the hole was deep enough to present a hazard, the vehicle would have to wait until the hole was close enough to satisfy:

$$\frac{h}{R} = \frac{r}{2r} = \frac{1}{2}$$

This will be called the **hole occlusion rule**. While it may seem that properly placed high depression scanlines are all that is required to detect the hole, this is not the case. Recall from the pointing rules that obstacles inside the stopping distance cannot be avoided at all. It is often the case that to allocate range pixels for hole detection is to waste throughput on obstacles that cannot be avoided anyway. Hole detection *requires a separate higher speed obstacle avoidance channel*.

Holes generate range shadows at the leading edge when the map resolution is sufficiently high. Alternatively, a vehicle could adopt a policy of avoiding such shadows specifically when they are in the path of the wheels.

Real Numbers

On the HMMWV, this rule requires that the minimum range be about 5 meters. From Figure 12, it is clear that, depending on the braking reaction time, this may limit speeds unreasonably. Later results will show that the cost to throughput is extreme when the lower scanline is depressed to ranges close to the sensor height.

5.3 Occlusion Problem and Unknown Hazard Assumption

One of the fundamental problems of rough terrain navigation at any speed is the occlusion problem and little can be done about it. The occlusion problem can be mitigated somewhat by noticing that, most of the time, regions which are occluded are occluded by hazards. To accept occlusion then is to assume that occlusions are hazards - that is, to consider large unknown regions to be unnavigable by default¹⁶. This will be called the **unknown hazard assumption**. **Guaranteed safety** requires that this assumption be made.

A further mechanism for dealing with the problem is to limit the roughness of the terrain that will be navigated, or at least to design a system to assume some limit. This will be called the **benign terrain assumption**.

5.4 Lateral Occlusion

A special case of the occlusion problem arises when a large portion of an already narrow HFOV is occluded by a large object as shown in the following figure. Oftentimes, the system cannot see a clear path ahead even though there is one. This problem, the **lateral occlusion problem**, aggravates the tunnel vision problem and it is aggravated by increased system cycle time and steering dynamics.

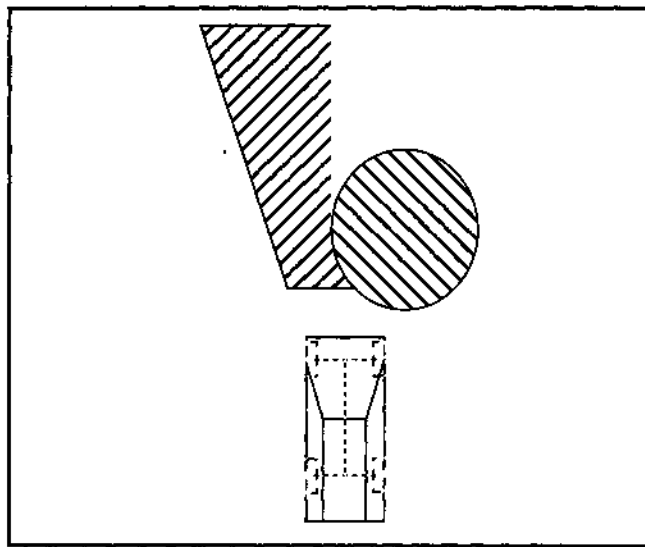


Figure 28 - Lateral Occlusion Problem

16. This assumption is far more necessary than it may appear. Without it, the system will happily drive straight off a cliff.

Section 6: Perceptual Bandwidth

Another aspect of throughput is the raw data rate required of communications electronics and the computation required for a single image pixel. This section investigates these perceptual bandwidth requirements.

6.1 Sensor Flux

The **sensor flux** Ψ represents the solid angle subtended by the field of view generated per unit time. It can be written as:

$$\Psi = \text{HFOV} \times \text{VFOV} \times f_{\text{images}}$$

6.2 Sensor Throughput

The number of range pixels generated per unit time by a sensor will be called the **sensor throughput** f_{pixels} . Existing sensors can be characterized as completely nonadaptive since neither the field of view nor the angular resolution change with time. For such sensors, the sensor throughput is given by:

$$f_{\text{pixels}} = \frac{\Psi}{(\text{IFOV})^2}$$

The IFOV is the angular resolution of the sensor. A sensor for which Ψ is constant is called **constant flux**, and one for which the IFOV is constant is called **constant scan**.

6.3 Sweep Rate

The product of the vertical field of view and the frame rate is a measure of the angular velocity of the beam, and is known as the **sweep rate**:

$$\dot{\theta} = \text{VFOV} \times f_{\text{images}}$$

6.4 Processor Load

In a simple case, a range pixel must have its coordinates converted at least three times in order to place it into world coordinates. The complete transform is developed in the appendices. First, the ray is converted to a sensor-fixed cartesian system. This operation can be implemented via lookup tables. Each pixel is looked up and multiplied by the range to get the RHS vector of the transform. This costs 6 flops.

Next, the first RPY matrix converts from the sensor frame to the body frame. This matrix is usually fixed so its elements can be computed at initialization time. The matrix multiplication costs 12 flops for a total of 18 so far.

Finally, the elements of the second RPY matrix depend on the vehicle pose, so they must be computed in real time. Assume the trig functions are implemented via lookup table and hence cost 1 flop each, to give 6 flops. The matrix terms cost about 14 flops. The multiplication costs 12 flops. Thus the left matrix multiplication costs an additional 32 flops. The total processing of a range pixel is therefore 50 flops.

More generally, it is useful to define the **processor load** σ_P as the number of flops necessary to process a single range pixel.

$$\sigma_P = \frac{\text{flops}}{\text{pixel}}$$

Real Numbers

While the analysis suggests a value of 50 for this, actual data for an entire navigator is ten times as large. Specifically, the ERIM pulse rate is 16 KHz and navigation consumes 8 Mflops. This gives a value of 500.

Thus, the relationship between processing load and sensor throughput is:

$$f_{\text{cpu}} = f_{\text{pixels}} \times \sigma_P$$

6.5 Perceptual Software Efficiency

The inverse of the processor load is called the **perceptual software efficiency** η_S .

$$\eta_S = \frac{1}{\sigma_P}$$

6.6 Computational Bandwidth

The computational bandwidth is the number of flops required of a processor per unit time. If *the geometric transforms of perception are the only aspect of the system considered*, this quantity is related to the sensor bandwidth by the processor load:

$$f_{\text{cpu}} = \frac{f_{\text{pixels}}}{\eta_S} = \frac{1}{\eta_S} \frac{\Psi}{(\text{IFOV})^2} = \frac{1}{\eta_S} \frac{\text{HFOV} \times \text{IFOV} \times f_{\text{images}}}{(\text{IFOV})^2}$$

6.7 Communications Bandwidth

Define the **communications load** σ_C as the width of a range pixel in bits. Assuming that beam angular position information is implied by position in the scan, the bandwidth required is:

$$f_{\text{comm}} = f_{\text{pixels}} \times \sigma_C$$

At 10 bits per pixel, the required bandwidth in MHz is $1/5$ the required processing rate in Mflops. Hence high bandwidth communications between the sensor and the computing engine is also a fundamental necessity of high-speed range image-based navigation.

Chapter 4: Acuity

As was shown earlier, the **acuity ratios** relate the configuration of the vehicle to the resolution required of the sensor in order to resolve obstacles. This section analyzes these aspects of vehicle performance for typical vehicles.

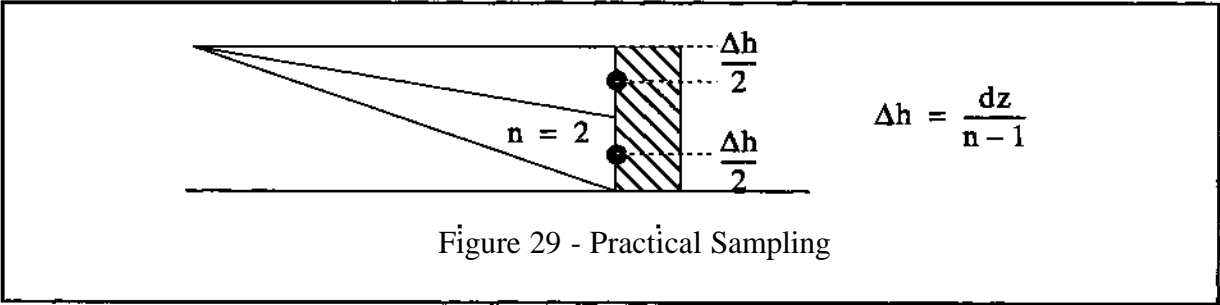
Section 1: Range Image Acuity

The ability of a vehicle to resolve obstacles in order to ensure safety is important. This section investigates the manner in which *vehicle configuration and sensor resolution together* determine the ability of a sensor to resolve obstacles. The following analysis is based on the **flat world assumption** and is not entirely correct in rough terrain. Nonetheless, it is a useful theoretical approximation.

1.1 Sampling Theorem

The acuity ratios were stated in the introduction according to a strict application of the **sampling theorem**. According to this theorem, proper recovery of a signal from its samples requires a sampling frequency at least twice that of the highest frequency of the original signal. This is a theoretical requirement and it must be interpreted carefully. Formally, **resolution** is *the smallest difference that a system can resolve*, and *not* the smallest thing that a pixel can cover.

Taking a wheel collision hazard as an example, a system with a 6 inch wheel radius requires a 3 inch vertical projection of a pixel. Such a system can only resolve a 3 inch step from a 6 inch step. Further, a 6 inch step appears as a 3 inch step if the center of the pixel is used in calculations. As shown in the figure, the size of a step obstacle is underestimated by Δh which is, in turn, a larger fraction of the true height as the pixel size increases. The resolution dz is related to the height underestimate Δh by:



A practical sampling theorem requires "n" times the signal frequency for some number n greater than 2. A practical acuity requirement will require an **oversampling factor** of $\frac{n}{2}$. An oversampling factor of 3 will allow discrimination of a 5 inch step from a 6 inch step, for example. As a theoretical analysis, the document will continue to use an oversampling factor of 1.

1.2 Terrain Smoothness Assumption

The smallest feature of interest to a planning algorithm is the smallest feature which can cause the vehicle to be in an unsafe configuration. Three sizes of hazard can be identified.

From the point of view of tipover, the motion of the vehicle over the terrain amounts to a process whereby high spatial frequencies present in the terrain are filtered. The vehicle longitudinal and transverse wheelbase determine the vehicle pitch and roll. Hence, from the point of view of tipover obstacles, cells a fraction of the size of the vehicle seem sufficiently small.

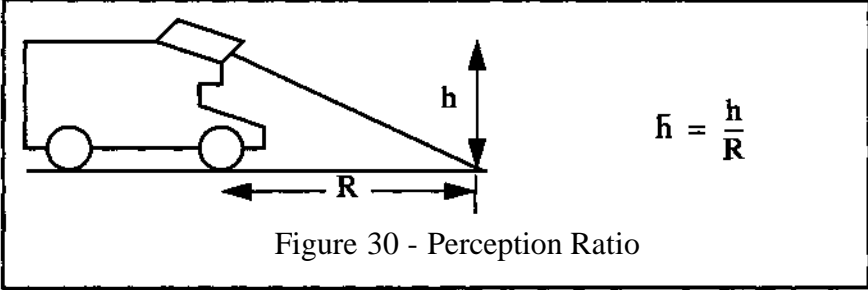
Another important form of obstacle is one which presents a gradient sufficient to collide with or trap a tire at operating velocity. Potholes and steps fall into this category. Based on this reasoning, cells on the order of the size of a wheel radius are needed to ensure that a wheel does not fall in a hole or drive over a step which would cause damage.

A man-made fixture such as a fence post or a natural feature such as a tree may feasibly exist in the path of the vehicle. Such features represent the extremes of terrain gradient and present the vehicle with both wheel collision and front bumper collision hazards. In the worst case, a nail in a board could feasibly exist in the vehicle's path.

Cell sizes on the order of a nail diameter are, as will be shown, infeasible for both throughput and sensor resolution reasons. It is therefore *necessary to assume that pathological cases do not exist*. A practical system must always assume that there are no man-made or natural hazards that are smaller than some practical limit. This is the **terrain smoothness assumption**.

1.3 Impact of Imaging Geometry on Acuity

Nothing could be more natural than to bolt the environmental sensor to the front of the vehicle - especially when long excursions are intended. For the present purpose, it can be considered a requirement to use such a sensor geometry. However, one important aspect of this approach to high-speed autonomy is that the ratio, called the **perception ratio**¹⁷ H:



is always a number much less than 1.0. This number shows up in many places in dimensional analysis. It can be shown that this geometric limitation of the sensor height is the root of some of the fundamental technological problems of autonomous navigation.

Real Numbers
For the HMMWV, the sensor height is about 2.7 meters and up to 30 meters of lookahead is required. Hence, for this vehicle, the perception ratio is 0.1.

17. The ratio of two important lengths is often an important angle in dimensional analysis. Here, the perception ratio is the angle of incidence of the beam with the terrain.

1.4 Nomenclature

The **instantaneous field of view** is the angular width of the laser beam¹⁸ or the angular width of a camera pixel. This is also known as the **beam dispersion** for laser rangefinders. The angular coordinates of a pixel are often expressed in terms of horizontal sweep or **azimuth** $\backslash/$, and vertical sweep or **elevation** $\backslash/$. When elevation is measured down from the horizon, it is called **depression**. The relationships between the beam width and its projections onto three orthogonal axes are given below. In the analysis, three orthogonal axes are considered to be oriented along the vehicle body axes of symmetry:

- x - **crossrange**, in the groundplane, normal to the direction of travel
- y - **downrange**, in the groundplane, along the direction of travel
- z - **vertical**, normal to the groundplane

1.5 Sampling Problem

The differential mapping from image space onto cartesian space is both nonlinear, and a function of the terrain geometry. The density of pixels on the groundplane can vary by three orders of magnitude, and it varies with both position and direction. Hence, the shape and density of pixels in traditional sensors is not optimal.

Significant variation in groundplane resolution can cause **undersampling** at far ranges and **oversampling** close to the vehicle. If resolution is chosen appropriately at the maximum range, pixel density at the minimum range can be extreme and cause significant waste of computational bandwidth if all pixels are processed. This is a subproblem of the **acuity problem** which has been called the **sampling problem**.

The traditional solution to the sampling problem is to use **interpolation** to fill in small unknown regions. There is a limit to how much interpolation can be performed. Fundamentally, the limit used is another aspect of the **terrain smoothness assumption**.

18. Actually, the IFOV is the angle subtended by the **receive** optics, but this distinction will be ignored here.

1.6 Differential Imaging Kinematics

Note that the following differential relationships come from projective geometry and not from differentiating the coordinate transforms. More precise relationships are available from the Jacobian of the coordinate transformation, but this depends on the precise scanning pattern chosen. In order to avoid dependence on the scanning pattern, the following approximations are used which are good enough for the purpose of the discussion here.

Approximations for the transformations from image space to the groundplane are easy to write as follows:

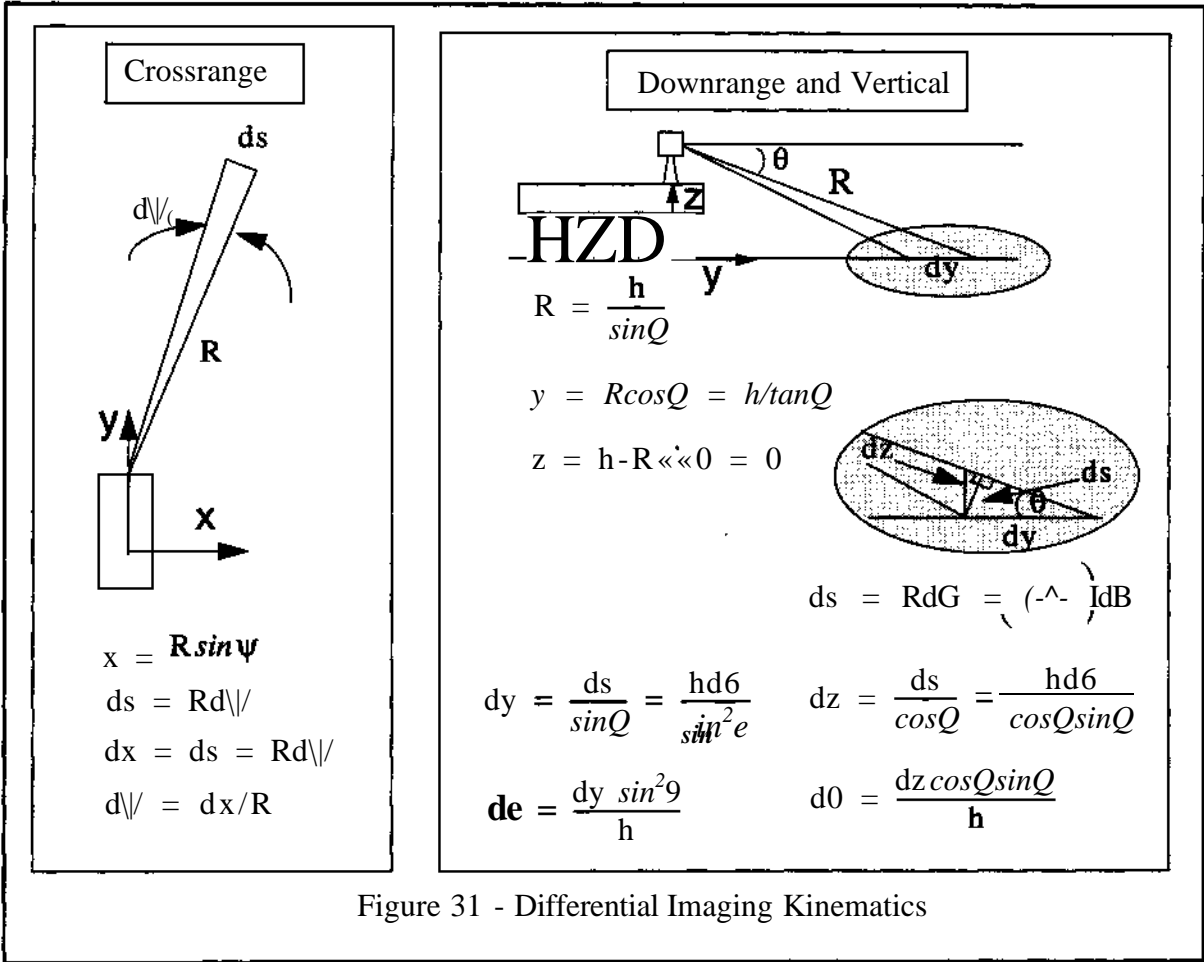


Figure 31 - Differential Imaging Kinematics

Consider the following approximations to these when elevation spacing $d\theta$ equals azimuth spacing $d\psi$ as is almost always the case:

$$dy = \frac{ds}{\sin \theta} \approx \frac{R d\theta}{\left(\frac{h}{R}\right)} \quad dx = dz = R d\theta$$

These approximations will be used extensively throughout the document.

1.7 Pixel Footprint Area and Density Nonuniformity

Multiplying the above expressions:

$$dx dy = R d\theta \frac{R d\theta}{\left(\frac{h}{R}\right)} = \frac{R^2 d\theta^2}{\left(\frac{h}{R}\right)}$$

Hence, *the area of a pixel when projected onto the groundplane is a constant times the cube of the range*. Due to the projection onto the groundplane¹⁹, it is increased by the inverse of the perception ratio over what would be expected based on the area of an expanding wavefront. This is *the variation of pixel size with position*.

It is typical to allocate the sensor vertical field of view such that $R_{min} \approx 2h$. Under this configuration, the density variation over the entire field of view is given by:

$$\left(\frac{R_{max}}{R_{min}}\right)^3 = \frac{1}{8} \left(\frac{R_{max}}{h}\right)^3$$

The **range ratio** is an important variable because it determines the degree to which groundplane resolution is wasted in the current generation of sensors.

Real Numbers

For the HMMWV using the first ambiguity interval of the ERIM sensor, this density variation is 50. So if data were of appropriate density at the extremes of the field of view, it would be 50 times too dense near the vehicle.

19. Again from radiometry. The pixel solid angle is constant with range, so an R^2 growth would be expected. However, the projection onto the groundplane by close to 90° provides the extra R .

1.8 Pixel Footprint Aspect Ratio

Dividing the above expressions:

$$\frac{dx}{dy} = \frac{dz}{dy} = \left(\frac{h}{R}\right)$$

Hence, *the pixelfootprint aspect ratio at any range for which R>his given by the perception ratio*. The assumption R » h comes from the approximations to the resolution transforms. These formulae are extremely good approximations whenever R is larger than 2h. For most vehicles, this condition is always true²⁰. This is the *variation of pixel size with direction*.

The impact of this is that, since the perception ratio is often close to 1/10, terrain is often oversampled in the crossrange direction by an order of magnitude with respect to the downrange direction. Practically, then, only one *column* in ten is needed to ensure adequate coverage of the environment if square pixels are used and downrange resolution is adequate. More importantly, a sensor designed specifically for high-speed autonomy would have significantly nonsquare pixels. This arises fundamentally because stopping distance exceeds sensor height by a large factor.

Real Numbers

On the HMMWV, at the 5 meter minimum range given by 30° depression, it is 2:1. At 27 meters, this is 10:1, at 50 meters, it is just under 20:1.

20. Perhaps the most important distinction of high-speed autonomy is the fact that sensor height is an order of magnitude smaller than the vehicle braking distance. This fact has many implications.

1.9 Minimum Sensor Acuity in Image Space

The nonuniformity of pixel spacing is quite extreme when the perception ratio is close to 0.1 and the scanline depression angle approaches 30°. Consider now what happens when the spacing between pixels begins to approach the size of the vehicle itself. At far ranges dy is much larger than dx . So it is dy which will first approach L .

Real Numbers

This is no mere abstraction. At 50 meters range, for the ERIM sensor, the spot spacing is an incredible 10 meters or three times the size of the vehicle.

Consider that since computation of the vehicle pitch angle depends on having two different elevations under the front and rear wheels, the pixel spacing dy must be no larger than one-half the wheelbase for this to be practical. Beyond the range at which this occurs, sensor data contains no useful information at all, unless data from several images can be registered accurately enough to contain useful information.

Equating dy to one-half the wheelbase:

$$dy = \frac{1}{2} \cdot R \sin \theta$$

Rewriting gives a very elegant expression:

$$\left(\frac{L}{R}\right)\left(\frac{h}{R}\right) = \sin \theta$$

This is a very interesting relationship which says that at some resolution, the sensor measures nothing useful, or equivalently, the holes in the map are almost large enough to swallow the whole vehicle. This resolution occurs when the product of the normalized wheelbase and the perception ratio equals one-half the angular resolution of the sensor. This is another expression of the **minimum sensor acuity rule**. Any of the variables can be considered to be absolutely limited by the others in the expression. The equation relates two key nondimensional variables and connects the vehicle shape to the required sensor angular resolution.

Two conclusions can be immediately drawn from this relationship. First, the IFOV itself limits the maximum range and therefore the maximum speed of the vehicle. Second, for any given IFOV, there exists a vehicle speed beyond which sufficiently accurate map registration becomes *essential* to reliable hazard detection.

1.10 Maximum Sensor Acuity in Image Space

It is possible to formulate a similar rule by considering the much more stringent requirements of resolving a wheel collision hazard at the maximum range. In order to resolve a wheel collision hazard, spatial resolution in the vertical direction must be sufficient to land, say, 2 pixels on a vertical surface at any given range. A collision hazard for a given wheel radius r is a step of the same order as the wheel radius. Therefore, resolving a collision hazard requires:

$$dz = \frac{r}{2}$$

which can be rewritten in terms of the approximations to the forward resolution transforms thus:

$$Rd\theta = \frac{r}{2}$$

Solving for the pixel angular resolution:

$$d\theta = \frac{1}{2}\left(\frac{r}{R}\right)$$

This is another expression of the **maximum sensor acuity rule**²¹. Pixel sizes below this limit are excessive. Pixel angular resolution must be no larger than one-half the normalized wheel radius in order to resolve a collision hazard at a given range. This can be rewritten in several ways:

$$d\theta_{\min} = \frac{1}{2}\left(\frac{r}{R}\right) = \frac{1}{2}\left(\frac{r}{h}\right)\left(\frac{h}{R}\right) = \frac{1}{2}\left(\frac{r}{L}\right)\left(\frac{L}{h}\right)\left(\frac{h}{R}\right)$$

Notice that maximum acuity is related to wheel size and lookahead. Since lookahead increases as the wheelbase increases, maximum acuity is related to the **wheel fraction**. On this basis, an all-wheel vehicle configuration (where the wheels are large compared to the body size) requires the lowest perceptual resolutions. It is often necessary to violate this rule and adopt assumptions about the terrain in order to lower throughput requirements to a practical level. Notice that the minimum rule is quadratic in 1/R, whereas the maximum rule is linear. Both constraints are equal when:

$$R = \frac{Lh}{r}$$

Real Numbers

For the HMMWV, using a wheelbase of 3.3 meters, a sensor height of 2.7 meters, and a wheel radius of 0.475 meters, this range is 18.8 meters. This implies that minimum acuity is the more stringent requirement beyond this range.

21. Other acuity rules can be defined by relating any resolution requirement to some vehicle dimension, and substituting the resolution transforms. For instance, if downrange resolution must be less than a wheel radius, set dy = r. Then a rule is generated which is similar to the minimum acuity rule, but more stringent.

1.11 Kinematic Maximum Range and the Myopia Problem

The minimum sensor acuity rule gives a kinematic limit on maximum sensor range which says that at this range, the holes in the map are too large. This limit is surprisingly close. Solving for R:

$$R = \sqrt{\frac{hL}{2d\theta}}$$

Real Numbers

Using the HMMW vehicle wheelbase of 3.3 m, the sensor height of 2.7 m, and the ERIM beam dispersion of 10 mrad, this gives a remarkable 21.1 meters. The conclusion is that the information at the extremes of the first ambiguity interval of the ERIM sensor is barely useful unless some mechanism is available to register subsequent images accurately enough in pitch.

1.12 Maximum Angular Resolution

The minimum sensor acuity rule also gives a limit on sensor angular resolution. For a given nondimensional specification of the vehicle, the largest feasible sensor angular resolution is given by:

$$d\theta_{\max} = \frac{1}{2}\left(\frac{L}{R}\right)\left(\frac{h}{R}\right)$$

The minimum acuity rule takes any of these forms:

$$\left(\frac{L}{R}\right)\left(\frac{h}{R}\right) = 2d\theta_{\max} \qquad \left(\frac{L}{R}\right)^2\left(\frac{h}{L}\right) = 2d\theta_{\max} \qquad \left(\frac{L}{h}\right)\left(\frac{h}{R}\right)^2 = 2d\theta_{\max}$$

and it can be rewritten in terms of any of the other nondimensionals describing the vehicle shape.

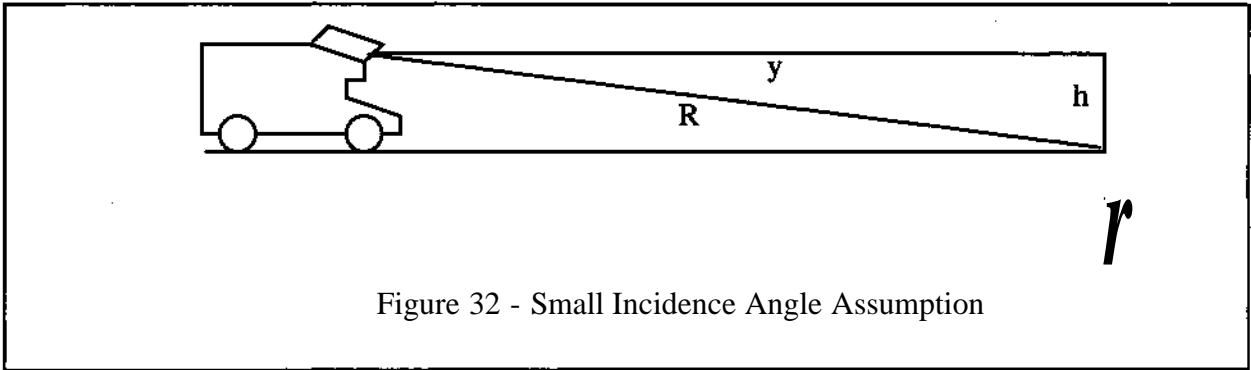
Notice that all of these results are *independent of h or R itself*, and *independent of the scanning pattern* of the sensor, except that uniform pixel angular spacing is assumed. All of these results are intrinsic when h << R. The issue which causes nonuniform sampling is the perception ratio: it has very little to do with the sensor itself.

1.13 Acuity Problem

For contemporary vehicles, the myopia problem and the acuity problem are linked because poor angular resolution is the typical limit on the useful range of a sensor. The above analysis is based on the **flat world assumption**. On rough terrain, there is no practical way to guarantee adequate acuity over the field of view because there will always be situations where pixels have glancing incidence to the terrain.

1.14 Small Incidence Angle Assumption

When the perception ratio is small, the range measurement from the sensor to the environment is almost identical to its groundplane projection (because the angle involved is so shallow). Indeed, the relative error in assuming the two are equal is the square of the perception ratio:



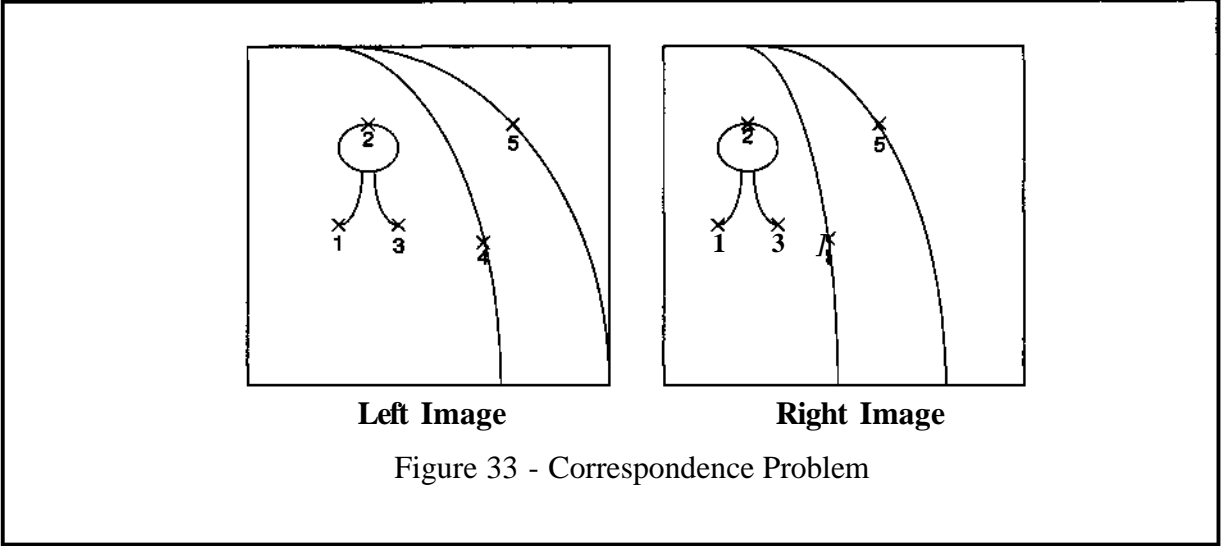
This is on the order of 1% for high speeds simply because the range is so large. The assumption of a small perception ratio will be called the **small incidence angle assumption**. It will be a critical assumption required later in order to manage perceptual throughput.

Section 2: Brief Introduction to Stereo Vision

Stereo vision, or **stereoscopy**, is a fundamental ranging technique having its roots in biological vision systems. That humans use two their eyes to deduce range is well known. This section is provided to introduce terminology and concepts upon which later analysis sections will depend.

2.1 Correspondence Problem

The fundamental issue in the implementation of stereo vision is the computation of the correspondence between pixels in a pair (or more generally a set) of images of the same scene. This problem is called the **correspondence problem**.



2.2 Disparity and Epipolar Geometry

The **disparity** d of a pixel in a stereo pair is the difference in the positions in the image plane of two pixels that correspond. In the most general case, disparity is a vector-valued function because the pixel that corresponds to pixel $[i,j]$ in the left image may be located anywhere in the right image. Thus, if x and y axes are used in each image plane, and l and r subscripts denote each image:

$$\vec{d}(x, y) = (x_r - x_l)\hat{i} + (y_r - y_l)\hat{j}$$

However, the **epipolar geometry constraint** can be used to reduce the vector disparity field to a scalar field. In principle, the ray through any pixel in one image corresponds to a line in the other image, and this line can be forced to be aligned with either the rows or columns of the second image for all pixels in the first.

It is useful to represent disparity as the tangent of an angle by dividing by the **focal length**. Thus the **normalized disparity** is defined as:

$$\delta = \frac{d}{f}$$

2.3 Area-Based Stereo

Mobile autonomy requires dense geometric models of the environment. For this reason, area-based stereo algorithms - which attempt to provide a range estimate for every pixel in an image - are the algorithms of choice. Three basic operations are fundamental to area-based stereo.

2.3.1 Normalization

Let $l[i, j]$ and $r[i, j]$ denote the intensity values of the left and right images of a horizontal baseline pair and let $a[i, j]$ denote either of them when a distinction is unnecessary. Similarly, let $L[i, j]$, and $R[i, j]$ denote preprocessed versions of these images and $A[i, j]$ denote either of them when a distinction is unnecessary. Typically, preprocessing is used to remove local gain and bias differences between individual cameras.

One of two common forms of preprocessing are often used. Let ∇^2 denote an actual or approximated²² Laplacian operator:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Similarly, let N denote the statistical normalization operator which scales a random variable onto normalized deviation from the mean in units of the standard deviation:

$$N(x) = \frac{x - \mu}{\sigma}$$

Either of these operators can be used to rescale image intensity based on convolution with an operator of support w by h (which may be unrelated to the correlation support discussed next):

Laplacian	$A[i, j] = \sum_{\substack{u = -h/2 \\ v = -w/2}}^{\substack{u = h/2 \\ v = w/2}} [u, v] a[i + u, j + v]$
Normalization	$A[i, j] = \frac{1}{N} \sum_{\substack{u = -h/2 \\ v = -w/2}}^{\substack{u = h/2 \\ v = w/2}} \frac{a[i + u, j + v] - \mu[i, j]}{\sigma[i, j]}$

N denotes the number of pixels in the normalization window. In the latter case, the mean and standard deviation are computed from the sampling distribution of intensities contained within the normalization window itself.

22. A typical approximation is the difference of Gaussians (DOG) operator.

2.3.2 Correlation

Various underlying formulations for computing the degree of match between two candidate pixels are in common use, but all are based on surrounding each pixel with a window and associating the properties of the entire window with its central pixel. Some common measures of match over a h by w window are the **sum of squared differences (SSD)**, the **sum of absolute differences (SAD)**, the **normalized correlation**, and the **sign correlation**. These are given, respectively, below:

$$\begin{aligned}
 D_s[i,j,d] &= \sum_{u=-h/2}^{h/2} \sum_{v=-w/2}^{w/2} (L[i+u, j+v] - R[i+u-d, j+v])^2 \\
 D_A[i,j,d] &= \sum_{u=-h/2}^{h/2} \sum_{v=-w/2}^{w/2} |L[i+u, j+v] - R[i+u-d, j+v]| \\
 C_x[i,j,d] &= \sum_{u=-h/2}^{h/2} \sum_{v=-w/2}^{w/2} L[i+u, j+v] R[i+u-d, j+v] \\
 C_s[i,j,d] &= \sum_{u=-h/2}^{h/2} \sum_{v=-w/2}^{w/2} \text{sgn}(L[i+u, j+v]) \text{sgn}(R[i+u-d, j+v])
 \end{aligned}$$

The first pair of operators measure the degree of mismatch, or decorrelation, of candidate image window pairs, and are typically minimized at a match. The second pair of expressions measure the degree of match, or correlation, of candidate image window pairs, and are typically maximized at a match. If memory permits, these expressions can be computed for every possible row, column, and candidate disparity. The resulting 3 dimensional matrix will be called the **correlation tensor**.

2.3.3 Disparity Estimation

Note that if row and column indices are fixed, then the resulting line through the correlation tensor can be regarded as a single-valued function of disparity. This curve is called the **correlation curve** of the pixel at $[i,j]$. *The fundamental operation of area-based stereo is to extremize this curve* and then associate the value of disparity at the extremum with the pixel in question. This operation involves the assumption that two regions that look the same are the same, called here the **maximal visual similarity assumption**. Clearly this assumption is invalid where the scene contains repetitive patterns. This problem is called the **repetitive texture problem**. However, the assumption is also invalid in more subtle ways to be investigated shortly.

Note also that *disparity estimation and correlation are coupled*. The computation of correlation is dependent on some assumption about the shape in the right image of the region corresponding to a rectangle in the left. Any assumed shape, including the classical rectangular shape assumption is tantamount to assuming a local disparity field consistent with that shape.

2.3.4 Disparity / Correlation Coupling

Thus computation of correlation requires knowledge of disparity and the computation of disparity comes from extremizing correlation. This expresses the notion of **disparity / correlation coupling**. This is an important idea in general stereo and a particularly important one in the target application.

2.4 Fundamental Inconsistency of Area-Based Stereo

Typical implementations of area-based stereo rely on several assumptions which are questionable in natural outdoor scenes. Indeed, the assumptions that are historically made are inconsistent and this inconsistency causes poor performance in the target application of mobile autonomy.

2.4.1 Uncorrelated Disparity Assumption

The **uncorrelated disparity assumption** asserts that the range of any pixel in the scene is independent of the range to any other pixel in the scene. While realistic scenes certainly have structure, the occluding edges which identify obstacles may occur anywhere without prior knowledge of the scene, so this assumption is a safe one to adopt.

In implementation terms, this assumption occurs implicitly when the correlation curve is extremized without any reference to the information in adjacent curves. There is, however, one important constraint on disparity which must be mentioned for completeness.

The **occlusion constraint** states that certain disparity gradients are physically impossible. For example, if pixel i in the left image is matched to pixel k in the right, then pixel $i+1$ cannot be matched to any pixel whose index is less than k . If this were the case, then pixel k would not have been visible in the left image and the original match of i to k would not have been possible.

This constraint is apparently a contradiction, but it can be resolved as follows. While the geometry of the environment is unconstrained, the geometry of stereo triangulation implies that certain perfectly legitimate scene characteristics cannot be seen from both images simultaneously and hence cannot be legitimately generated by stereo vision.

2.4.2 Correlated Disparity Assumption

While disparity estimation typically makes the uncorrelated disparity assumption, correlation typically assumes precisely the opposite. The opposite assumption is the **correlated disparity assumption**. Indeed, if the ranges of adjacent pixels are *independent*, then there is absolutely no reason to expect that the properties of a window around a pixel will be *dependent* on the properties of the central pixel. That is, if pixel i in the left matches pixel k in the right, then there is no basis for also assuming that pixel $i+1$ matches $k+1$ and hence no justification for summing their correlations together in a local window.

2.4.3 Inconsistency

Thus, area-based stereo directly violates its own justifying assumptions as it is classically implemented. This inconsistency²³ is intrinsic in almost all current implementations and it likely arises for efficiency reasons. Another way to express the inconsistency is to say that area-based stereo assumes that the disparity gradient is zero over the correlation window, for if it were not, there would be no reason to expect the extremum of the correlation curve to have any meaning. Having visited the extremes of the disparity correlation assumption, following subsections will investigate the spectrum of the assumption in terms of a quantity called **geometric decorrelation**. The computation of this quantity will first require an investigation of the disparity gradient.

23. Ultimately, this issue is related to the perception ratio which drives so much of the design. The disparity gradient is nonzero because the typical angle of incidence is the perception ratio - which is certainly not zero.

Section 3: Range Acuity in Stereo Vision

Stereo vision range data is distinct from laser rangefinder data in that the dominant acuity concern is often range resolution. This section investigates this issue and a few related ones in the context of mobile autonomy.

3.1 Range Resolution

The equations of stereo triangulation are given in the appendices as:

$$y = b \frac{f}{d} = \frac{b}{\delta}$$

where y is the range to a matched feature measured normal to the image plane, b is the baseline separation of the cameras, f is the focal length, d is the disparity, and δ is the normalized disparity.

This can be differentiated to give the range resolution of stereo as a function of range:

$$\Delta y = \frac{-b}{\delta^2} \Delta \delta = -\frac{y}{\delta} \Delta \delta = -\frac{y^2}{b} \Delta \delta$$

which shows the well-known quadratic growth of stereo range resolution with range when disparity resolution is fixed [58]. Disparity can be resolved no better than a single pixel unless **subpixel disparity estimation** is performed. Intuitively, the quadratic growth can be seen from the following figure:

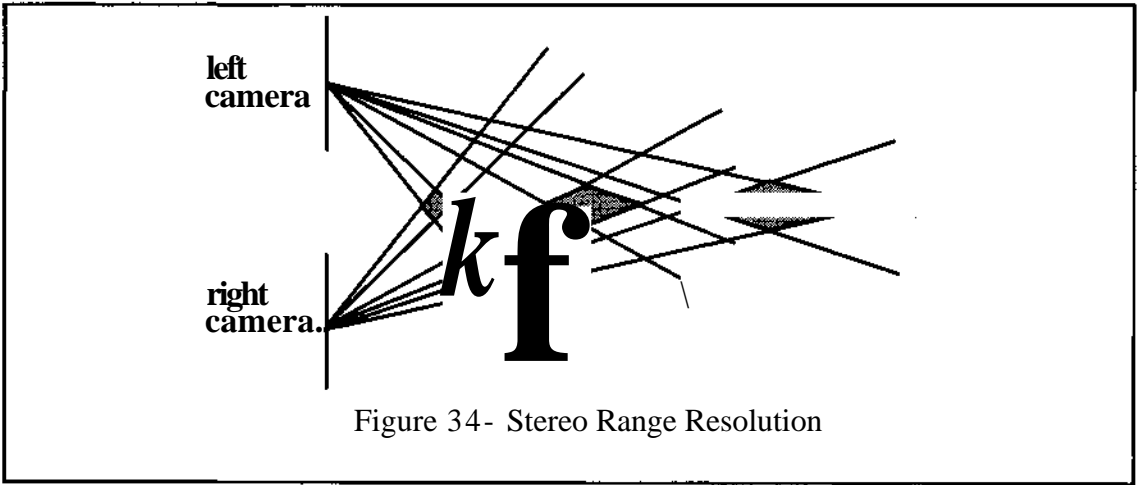
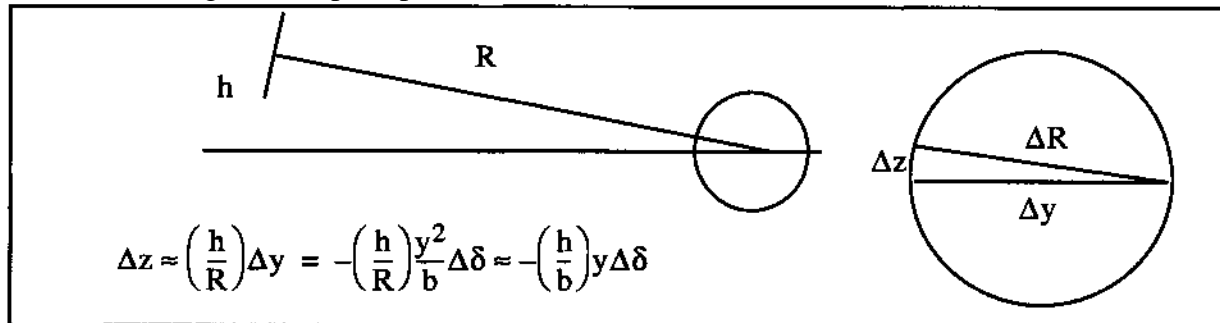


Figure 34- Stereo Range Resolution

3.2 Need for Subpixel Disparity Estimation

For high-speed vehicles, subpixel disparity estimation is a necessity on contemporary systems. Indeed, it is barely possible to resolve the pitch of the vehicle from a stereo range image otherwise. The height resolution resulting from disparity resolution can be computed kinematically by noting that stereo range resolution is normal to the image plane. A good approximation to the tangent of the sensor tilt angle is the perception ratio h/R , thus:



The quantity b/h will be called the **normalized baseline**. It is shown later to be equal to the image disparity gradient for flat terrain. Note that although range resolution is quadratic in range, *elevation quantization noise is linear in range*.

3.2.1 Minimum Acuity

Under minimum acuity, the elevation noise of a processed image must be less than, say, 10% of the noise required to tip the vehicle. Let θ_{\max} be the maximum allowable pitch of the vehicle, and L be its wheelbase. Then, minimum acuity requires the noise in predicted pitch to satisfy:

$$\Delta\theta < \frac{\theta_{\max}}{10}$$

In the worst case, the elevations under the front and rear wheels both contribute to the pitch error, so that:

$$\Delta\theta = \frac{\Delta z}{L}$$

and the minimum acuity criterion becomes:

$$\frac{2\Delta z}{L} < \frac{\theta_{\max}}{10} \Rightarrow \frac{2\left(\frac{h}{b}\right)y\Delta\delta}{L} < \frac{\theta_{\max}}{10} \Rightarrow \Delta\delta < \frac{Lb\theta_{\max}}{20hy}$$

Real Numbers

Using the HMMWV vehicle wheelbase of 3.3 m, the sensor height of 2.7 m, a baseline of 1 meter, maximum pitch of 15° , and a range of 20 meters, this criterion gives 0.0006 rads. The disparity resolution of a typical 512 X 480 camera of 40° fov is 0.001 rads - which is almost good enough to satisfy the criterion. This implies that the technique of reducing image resolution to, say, 128 X 120 in order to improve throughput will not meet the minimum requirement unless subpixel disparity estimation is performed.

3.2.2 Maximum Acuity

Under maximum acuity, elevation noise should be less than, say, 10% of the radius of a wheel. Let r be the wheel radius. Then, maximum acuity requires the noise in predicted elevation to satisfy:

$$\Delta z < \frac{r}{10}$$

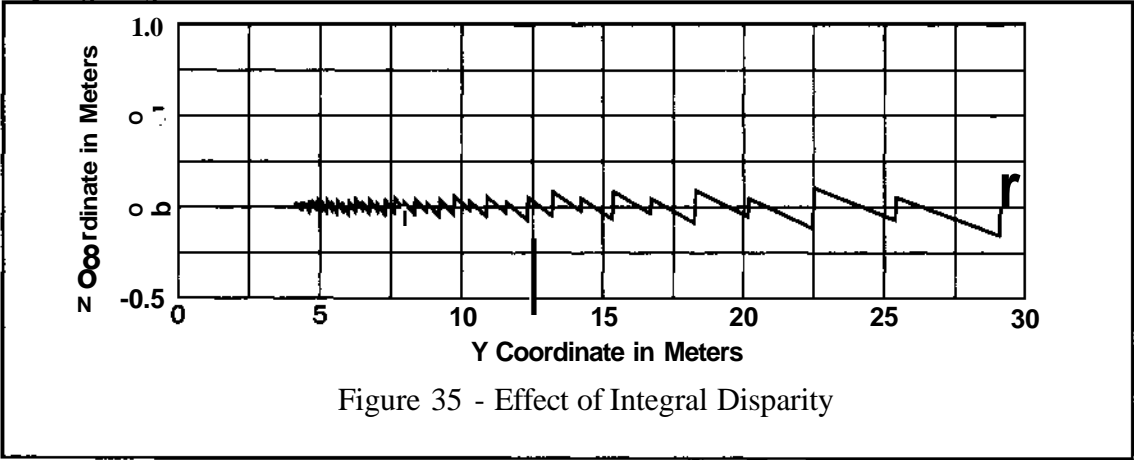
so the maximum acuity criterion becomes:

$$\Delta z < \frac{r}{10} \Rightarrow \left(\frac{h}{b}\right)y\Delta\delta < \frac{r}{10} \Rightarrow \Delta\delta < \frac{rb}{10hy}$$

Real Numbers
Using the HMMWV wheel radius 0.475 meters, the sensor height of 2.7 m, a baseline of 1 meter, and a range of 20 meters, this criterion gives 0.0008 rads.

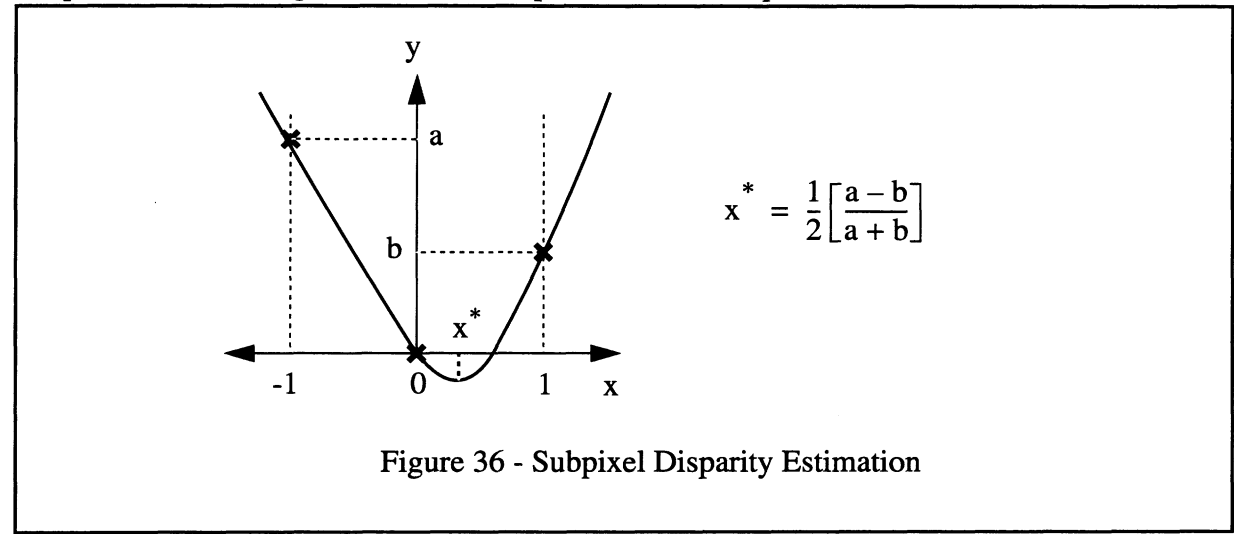
3.2.3 Integral Disparity

The following figure shows the effect of using discrete integral disparity values only for a sensor of height 2.5 meters, a VFOV of 30°, a sensor tilt of 16.5°, and a baseline of 1 meter. Although the ground truth terrain is perfectly flat, quantized disparity measurement causes step artifacts that increase in both amplitude and wavelength as range increases.



3.3 Subpixel Disparity Estimation

A well-known technique for enhancing the quality of stereo data is to interpolate the curves of correlation (or alternately, sum of squared differences) versus disparity to attempt to generate subpixel resolution. This problem can be expressed in its simplest form with minimum degrees of freedom if a coordinate system is established at the central of three points in (disparity, SSD) space which span a local minimum. Let x represent disparity, and y represent SSD value. The central SSD value is subtracted from the other two in order to reduce the variables to the two variables a and b, both positive. It is straightforward to fit a parabola and compute its minimum. The result is:



where the result is expressed in local coordinates and must be converted back to absolute disparity before used by adding the true disparity value of the local origin.

Section 4: Angular Acuity in Stereo Vision

Unfortunately, stereo vision range data is often also poor in angular resolution. The resulting range image is typically much poorer in angular resolution than the underlying angular resolution of the cameras used. This section investigates this issue and a few related ones.

4.1 Disparity Gradient

The disparity gradient is the spatial derivative of disparity. Under the **epipolar geometry assumption**, the disparity is a scalar field and an associated vector gradient field can be derived from it:

$$\nabla\delta = \frac{\partial\delta}{\partial x}\hat{i} + \frac{\partial\delta}{\partial y}\hat{j}$$

A simple expression, accurate to first order, is available for the gradient of disparity in an image of flat terrain. In such an image, the gradient is wholly vertical and therefore it is a scalar. The disparity gradient is related to the Δy spanned by the correlation window height $\Delta\theta$. It is known from the earlier angular acuity analysis for flat terrain that:

$$\Delta y = \frac{R^2}{h}\Delta\theta$$

and it was just derived that:

$$\Delta\delta = -\frac{b}{y^2}\Delta y$$

Substituting the first relationship into the second gives:

$$\Delta\delta = -\frac{b}{y^2}\frac{R^2}{h}\Delta\theta$$

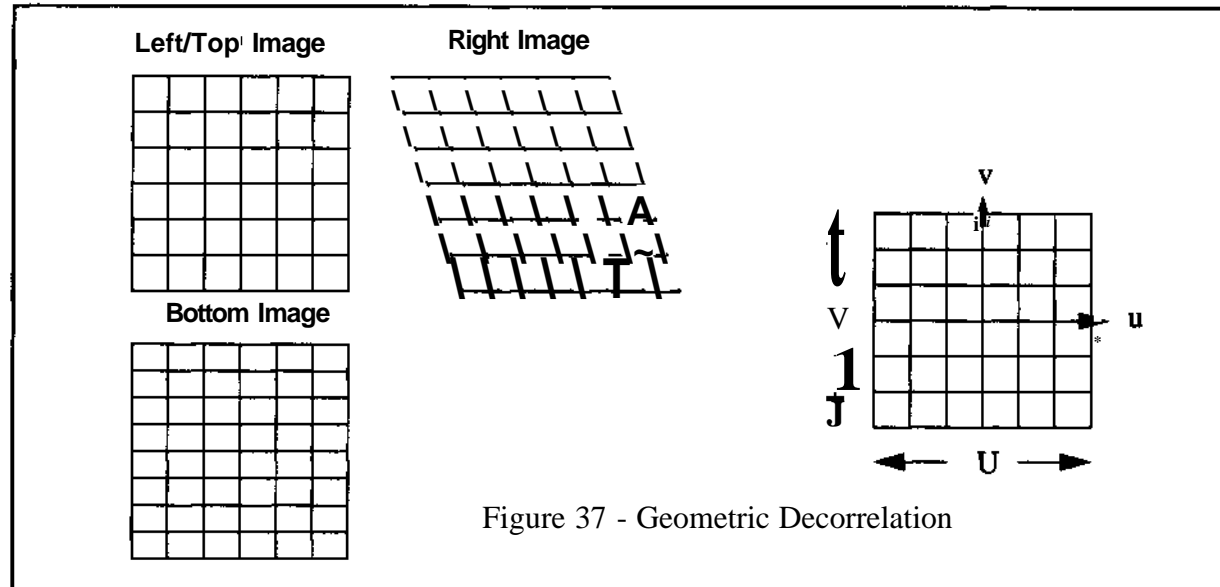
To first order, y and R are identical so that the disparity gradient is simply equal to the **normalized baseline**:

$$\frac{\Delta\delta}{\Delta\theta} = -\frac{b}{h}$$

This fact is a good justification for the use of fixed-sized correlation windows in outdoor settings. Of course, the disparity gradient at any point in an image is a direct function of the range gradient when it is projected into the image plane. In general, it may vary significantly as the terrain slope varies, but the above figure is an acceptable average value.

4.2 Geometric Decorrelation

One of the unavoidable difficulties of correlation-based stereo is the fact that even small fixed windows around matching points in both images will not overlap perfectly on the ground because the disparity gradient is normally nonzero. Consider the following figure which indicates the overlap of small correlation windows for both horizontal and vertical baselines when the disparity gradient is vertical and of magnitude $1/3$. Assuming square pixels, and assuming that the disparity gradient is uniform over the window, the right image will be shifted by one pixel horizontally for every three moved vertically. Likewise, the bottom image will squeeze 4 rows into every 3 rows in the top image.



This distortion is purely a geometric matter - independent of the image data itself and dependent solely on the disparity gradient field. Let a local coordinate system (u, v) be attached to the image plane at the central pixel of the reference image²⁴. Let the disparity gradient be given by:

$$\nabla \delta = \begin{bmatrix} \frac{\partial \delta}{\partial u} & \frac{\partial \delta}{\partial v} \end{bmatrix}^T = \begin{bmatrix} \delta_u & \delta_v \end{bmatrix}^T$$

and assume it is constant over the correlation window. Then the **geometric decorrelation** C_{err} will

be defined as the weighted integral of the area of the window where the weight of each differential region is given by its shift relative to the reference image²⁴. Thus, if δ is the position vector in the image plane, then:

$$C_{err} = \int_{-\frac{U}{2}}^{\frac{U}{2}} \int_{-\frac{V}{2}}^{\frac{V}{2}} (8_u u + 8_v v) du dv = \frac{5U^2V}{4} + \frac{8V^2U}{4}$$

Notice that the decorrelation depends only on the disparity gradient and the window dimensions.

24. This quantity is identical to the quantity called a_d in [65].

43 **Halo Effect**

The reduction in angular resolution due to the computation of correlation over the correlation window will be called the **halo effect** because it tends to manifest itself as a range halo that artificially increases the size of objects in the scene.

At least one reason for the halo effect is the use of correlation as a measure of region similarity. Precisely speaking, the correlation operator that is used in stereo is cross-correlation, not autocorrelation, because the two intensity fields being correlated are not identical. In continuous terms, the correlation curve is:

$$C(d) = \frac{1}{4hw} \int_{-h}^h \int_{-w}^w l(u,v) \cdot r(u-d,v) \, du \, dv$$

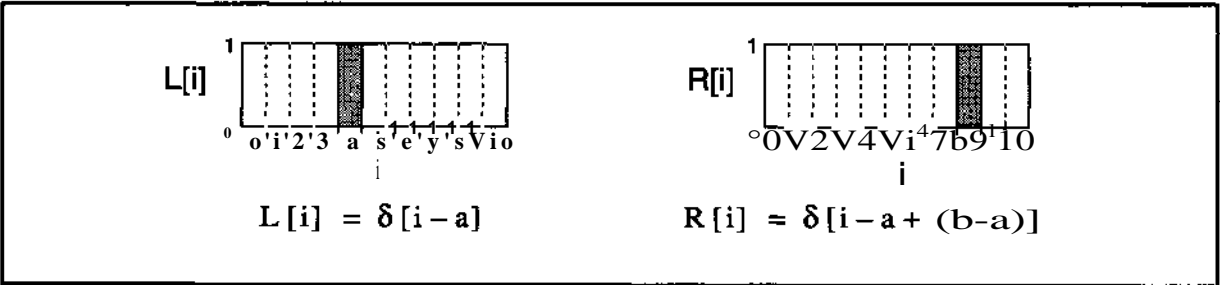
Ideally, a measure of region match would be the number of pixels that match normalized by the size of the window. This can be expressed in terms of a convolution with a voting operator $h(u,v,d)$ which is 1 when the pixels match and 0 otherwise.

$$C(d) = \frac{1}{4hw} \int_{-h}^h \int_{-w}^w Jh[l(u,v),r(u-d,v)] \, du \, dv$$

The essential difference between these two approaches is that, in correlation, *each pixel has a vote strength proportional to the square of its intensity*. The effect of intensity value on the computation of disparity will be called **intensity crosstalk**. Its practical implication is that a single very bright or very dark pixel in a correlation window can dominate the value of the normalized correlation and hence, its disparity can be associated with the entire correlation window.

4.3.1 **Intensity Crosstalk in Correlation**

This effect can be illustrated with detail and rigor using one dimensional binary images. Let the scene consist of a white sphere at a range from the two cameras which corresponds to a disparity of (b-a). Further let the sphere span exactly one pixel in either image, and let the background be completely dark and of disparity 0. This eliminates the issue of geometric decorrelation. Let the symbol δ temporarily represent the Dirac delta function, rather than normalized disparity. For this scene, the two images can be represented by the intensity functions:



Recall that the Dirac delta function is a generalized function whose value is 1 precisely when its argument is zero and 0 elsewhere. Its basic property is the **sampling property**:

$$\int_a^b \delta(x - c) f(x) \, dx = \begin{cases} f(c), & a < c < b \\ 0, & \text{otherwise} \end{cases} \quad \sum_a^b \delta[n - c] f[n] = \begin{cases} f[c], & a \leq c \leq b \\ 0, & \text{otherwise} \end{cases}$$

For the two idealized images above, the correlation tensor and the disparity curves can be computed easily in closed form. Let the correlation window be of width $2w+1$. The correlation tensor is a matrix:

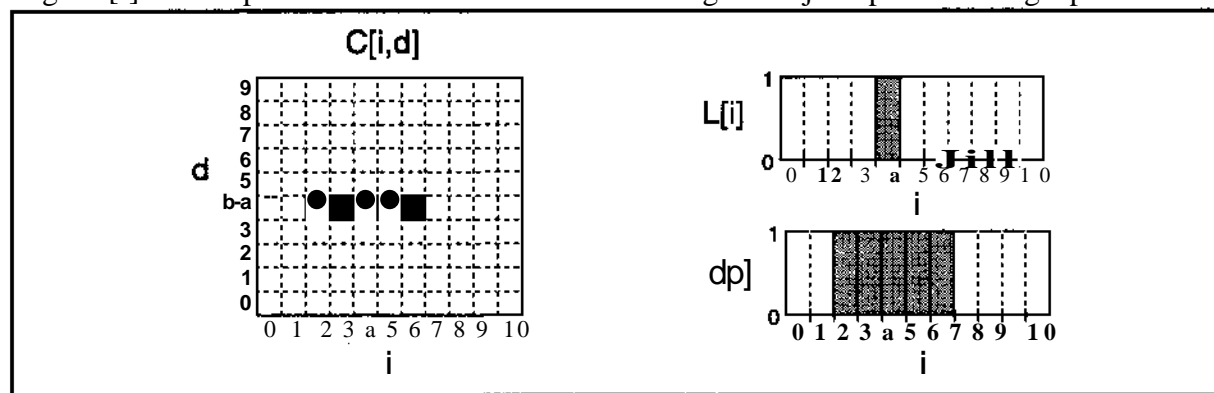
$$\begin{aligned} C[i, d] &= \sum_{n=-w}^w L[i + n] R[i + n - d] \\ C[i, d] &= \sum_{n=-w}^w S[i + n - a] R[i + n - d] \\ C[i, d] &= \sum_{n=-w}^w \delta[n - (a - i)] R[i + n - d] \\ C[i, d] &= \begin{cases} R[i + (a - i) - d], & |a - i| \leq w \\ 0, & \text{otherwise} \end{cases} \\ C[i, d] &= \begin{cases} \delta[-d + (b - a)], & |a - i| \leq w \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

Interpreting the result takes some care. First of all, the condition $|a - i| \leq w$ requires that 'a' be reachable by the correlation window from T in order for it to be contained within the correlation window centered at i. Unless this is true, the correlation tensor vanishes. Secondly, within the remaining region, the correlation tensor is unity only when: $-d + (b - a) = 0$ ^{^^}

$$-d + (b - a) = 0 \Rightarrow (b - a) = d$$

If the correlation tensor is rendered as an image where black indicates 1 and white indicates 0, the result is as shown below. The correlation curves are slices through the tensor for each fixed value of i. Extremizing each curve generates the disparity estimate for each pixel. The resulting disparity

image $d[i]$ is computed and rendered. Note that the original object spanned a single pixel whereas



the corresponding region in the disparity image is as large as the entire correlation window. Indeed, if the correlation window were arbitrarily large, *the disparity of a single pixel could be associated with the entire image*. For this scene and an arbitrarily large correlation window, the angular resolution of stereo vision is arbitrarily poor. This example illustrates that the use of correlation as a measure of region similarity is not sound in all cases because of intensity crosstalk.

4.3.2 Mitigation of Crosstalk with Logical Equivalence

As an alternative to correlation, it seems advisable after this example to investigate the value of logical equivalence as a measure of similarity. This is motivated by the fact that although background pixels *do match*, their *correlation with each other is zero* because their intensity is zero. There is a fundamental difference between correlation and matching which seems to be the root of the problem. Let $x \odot y$ denote the logical equivalence of x and y . This boolean function is the inverse of exclusive or, which is to say that it is one when x and y are either both 1 or both 0.

$$c[i, d] = \sum_{n=-w}^{n=w} L[i+n] \odot R[i+n-d]$$

$$c[i, d] = \sum_{n=-w}^{n=w} \delta[i+n-a] \odot \delta[i+(b-a)-a+n-d]$$

$$c[i, d] = \sum_{n=-w}^{n=w} \delta[n-(a-i)] \odot \delta[n-(a-i) + (b-a)-d]$$

Now both delta functions are 1 only when $n = (a-i)$ and $d = (b-a)$ simultaneously.

4.3.3 Summary

In summary, the classical process of computing correlation is not always correct because it fails to account for the disparity gradient. That is, the **correlated disparity assumption** is wrong in general. Also, the classical process for computing disparity from correlation is also incorrect because correlation biases the disparity by an amount dependent on the local intensity function. The **maximal visual similarity assumption** is also wrong in general. That these assumptions are not always correct is generally known but they are typically adopted for reasons of computational efficiency. However, they also contribute to the relatively poor quality of stereo range images.

Section 5: Perceptual Uncertainty

In the context of autonomous navigation, where terrain maps are used to represent the environment, it can be useful to explicitly represent elevation uncertainty and propagate it through planning computations. This section investigates the motivation behind representing elevation uncertainty and some simple techniques for decision-making in the presence of uncertainty.

5.1 Motivation

There are at least two motivations for representing uncertainty. First actual perception sensors typically have nonzero random variations in both range and angular position of pixels. Second, the environment is actually continuous and an entire distribution of elevation exists within the footprint of a single range pixel or a single map cell. The following two sections will consider the value of first order statistical modelling in the detection of discrete obstacles and terrain gradient obstacles.

5.2 Mechanism

Elementary decision theory provides a simple mechanism for estimating the probability that a random variable has a value within some interval given its sample statistics. Let x denote a random variable which will be interpreted as the degree of hazard expected at a particular point in the environment. Let x_{\max} denote a threshold value on x beyond which the hazardous condition represented by x is considered unacceptable. Further let x be distributed normally:

$$p(x) = N(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right)$$

The representation of a continuous probability distribution in terms of its first order statistics is equivalent to an assumption that the true distribution is Gaussian. For a Gaussian distribution, it is known that:

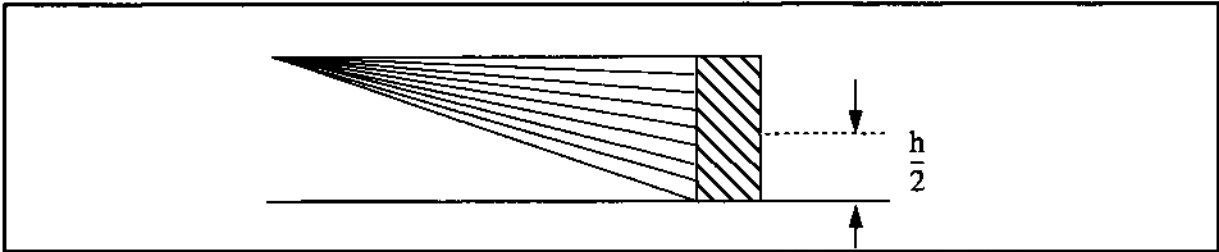
$$p[x \geq \mu + k\sigma] = \int_{\mu + k\sigma}^{\infty} p(x) dx = \text{erfc}(\mu + k\sigma)$$

for any positive real number k . This is called the complementary error function. As an example, when $k = 3$, the integral is roughly 0.01. Thus, there is a 1% probability that an observation of x will exceed $\mu + 3\sigma$.

This can be used as follows. Suppose a hazard estimate is computed from some sample of range points suitably transformed into a hazard estimate and its variance. Under an assumption that the hazard estimate is a Gaussian random variable, and under an assumption that the sample statistics for mean and variance approximate the true population parameters, then the hazard estimate $\mu + ka$ can underestimate the true hazardous condition only 1% of the time when $k = 3$.

53 Obstacle Height

A terrain map representation of the environment involves an implicit assumption that the world can be represented in 2-1/2 D because each point (x,y) gets a single elevation value. However, it is also possible to represent the variation of elevation within a cell in terms of its first order statistics. Consider an obstacle of height h, upon which n range pixels have landed as shown below:



Clearly, *if the mean elevation is used to represent the height of the obstacle, it will underestimate it by 50%*. Using a discrete integral, for n odd, the standard deviation of the elevations is:

$$\sigma_h^2 = \frac{1}{n} \sum_{i=-n/2}^{n/2} \left(\frac{h}{2} - \frac{h}{2} \frac{i}{n/2} \right)^2 = \frac{h^2}{4} \frac{1}{n} \sum_{i=-n/2}^{n/2} \left(\frac{i}{n/2} \right)^2 = \frac{h^2}{4} \frac{1}{n} \frac{1}{(n/2)^3} \sum_{i=-n/2}^{n/2} i^2 = \frac{h^2}{4} \frac{1}{n} \frac{1}{(n/2)^3} \frac{n^3}{12} = \frac{h^2}{12}$$

and thus an estimate of the height of an obstacle given the standard deviation is:

$$h = \sqrt{12} \sigma$$

Converting this to absolute coordinates, the maximum elevation of the cell is best approximated as:

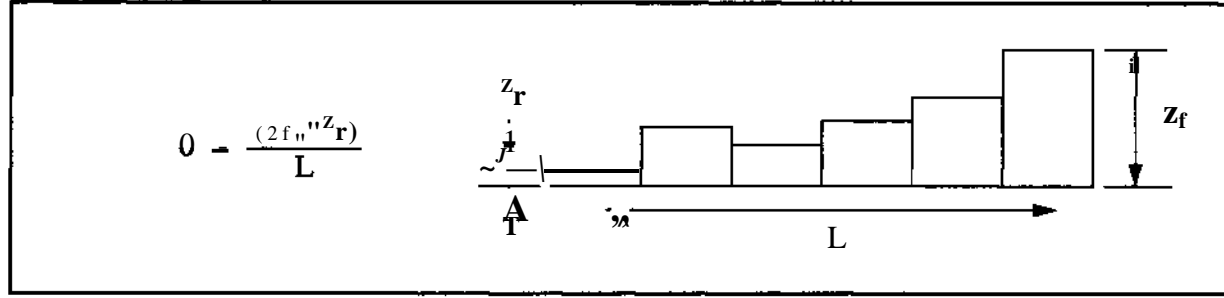
$$z = \mu + \frac{\sqrt{12}}{2} \sigma$$

where μ is the mean of the elevation distribution. This simple rule provides a trivial mechanism for estimating the elevation variation *within* a cell.

5.4 Terrain Gradient

Many forms of hazards that present themselves to a vehicle can be interpreted in terms of a gradient operator over some support length. For example, static stability involves estimation of vehicle pitch and roll under an assumption of terrain following. In the case of pitch, the appropriate length is the vehicle wheelbase. In the case of roll, the appropriate length is the vehicle width.

Consider the problem of estimation of the slope angle of the terrain in a particular direction over a certain distance. The distance L represents the separation of the wheels. Let the slope angle be represented in terms of its tangent, thus:



The two elevations can be considered to be a 2-vector whose covariance is:

$$C_z = \begin{bmatrix} G_r^2 & 0 \\ 0 & G_f^2 \end{bmatrix}$$

Under first order uncertainty propagation techniques, the covariance C_θ of the computed pitch angle is a scalar. It can be computed from the covariance C_z of the elevations as:

$$C_\theta = J^T C_z J$$

where J is, in this case, the gradient vector of pitch with respect to the elevations.

$$J = \begin{bmatrix} \frac{\partial \theta}{\partial z_f} & \frac{\partial \theta}{\partial z_r} \end{bmatrix} = \begin{bmatrix} \frac{1}{L} & -\frac{1}{L} \end{bmatrix}$$

The uncertainty in pitch is therefore simply:

$$\sigma_\theta^2 = \frac{1}{L^2} [\sigma_f^2 + \sigma_r^2]$$

This simple formula permits an assessment of confidence in the existence of an obstacle at any point. It is based on variations *between* cells.

Section 6: Positioning Bandwidth

Another aspect of the acuity requirement is the rate at which the vehicle position is sampled. This section investigates the interaction between vehicle speed and the bandwidth required of position estimation. Although the position of the ray through each pixel is known in body coordinates and can be accounted for, the entire image field of view sweeps over the terrain as the vehicle moves. It is necessary to localize a range pixel to within the map resolution in order to meet the maximum fidelity requirement. The smearing of the environmental model which arises from incorrect models of vehicle motion during the digitization process is called the **motion distortion problem**. The following analysis can be conducted for translational motion as well.

6*1 Heading and Positioning Bandwidth

For a turn of radius p , and a velocity of V , the yaw rate of the vehicle is given by:

$$\psi = \kappa V = \frac{V}{\rho}$$

If the range pixel returns a value of R , then the velocity of the pixel due solely to this vehicle motion is given by:

$$V_{\text{pixel}} = \frac{RV}{\rho}$$

At this speed, the pixel moves through a distance equal to the map resolution over a time period of:

$$\Delta t = \frac{\delta}{V_{\text{pixel}}} = \frac{\delta \rho}{RV}$$

Therefore, the vehicle yaw must be sampled at a rate of:

$$f_{\text{heading}} = \frac{RV}{\delta \rho}$$

If this positioning bandwidth is not met, the terrain map will smear unacceptably and hazards will be localized incorrectly due to poor resolution in the computations. In practice, interpolation can be used to supply this position estimate provided the sampling of the underlying heading signal satisfies the sampling theorem. This requirement implies that *every scanline* in a rangefinder image must use an independent vehicle pose estimate.

Real Numbers

Using the HMMW with a 30 meter lookahead, a velocity of 4.5 m/s, a turn radius of 7.5 meters, and a map resolution of 1/3 meter, this gives a frequency of 60 Hz.

6.2 Attitude and Positioning Bandwidth

In the case of attitude, the vehicle pitch will be considered. Let the vehicle pitch as it encounters a patch of rough terrain. The front wheel will lift by a small amount dz and the pitch of the vehicle will change by a small amount given by:

$$d\theta = \frac{dz}{L}$$

The pitch rate is therefore related to the terrain gradient in the forward direction as follows:

$$\dot{\theta} = \frac{1}{L} \frac{dz}{dt} = \frac{1}{L} \frac{dz}{dy} \frac{dy}{dt} = \left(\frac{dz}{dy}\right) \frac{V}{L}$$

For a terrain gradient of 1, a velocity of 3 m/s and a wheelbase of 3 meters, this figure is 1 rad/sec. This is unrealistically high because the suspension will attenuate this figure by a large factor. Let the maximum *instantaneous* pitch rate be called $\dot{\theta}_{max}$. Recall from the acuity analysis earlier that the differential relationship between the downrange position of a pixel and the elevation angle of the pixel is as follows:

$$dy_{pixel} = \frac{R^2}{h} d\theta$$

Therefore, the velocity of the pixel in the downrange direction due solely to this vehicle motion is:

$$V_{pixel} = \frac{dy_{pixel}}{dt} = \frac{dy_{pixel} d\theta}{d\theta dt} = \frac{R^2}{h} \dot{\theta}_{max}$$

At this speed, the pixel moves through a distance equal to the map resolution over a time period of:

$$\Delta t = \frac{\delta}{V_{pixel}} = \frac{\delta h}{R^2 \dot{\theta}_{max}}$$

Therefore, the vehicle pitch must be sampled at a rate of:

$$f_{attitude} = \frac{R^2 \dot{\theta}_{max}}{\delta h}$$

If this positioning bandwidth is not met, the terrain map will smear unacceptably and hazards will be localized incorrectly due to poor resolution in the computations. In practice, interpolation can be used to supply this position estimate provided the sampling of the underlying attitude signal satisfies the sampling theorem. This requirement often implies that *every pixel* in a rangefinder image must use an independent vehicle pose estimate.

Real Numbers

Using the HMMW with a 30 meter lookahead, a maximum instantaneous pitch rate of 1/8 rad/sec, a sensor height of 2.7 meters, and a map resolution of 1/3 meter, this gives a frequency of 150 Hz.

6.3 Motion Distortion Problem

Although the last analysis made some assumptions about the instantaneous vehicle pitch rate, the assumption is not too important in practice. In the most general case, the motion distortion problem has no solution because it also depends on the terrain gradient. As a vehicle crests a hill, for example, there will come a time when a single pixel will sweep several meters forward in a small fraction of a second. This can be seen, for example, in Figure 25.

6.4 Need for Simultaneous Digitization in Stereo Vision

In the case of stereo vision, a particularly serious motion distortion problem applies. Due to the nonlinear mapping from stereo disparity onto range, a very small apparent disparity due to relative motion of the cameras can cause a huge change in the perceived range to an object. As an example, let the vehicle turn with a high angular velocity ψ and let the camera digitization be out of synch by an amount Δt . Then, the relative angle introduced between two matching features due solely to this synch error is:

$$\Delta\psi = \psi\Delta t$$

As stated earlier, for a turn of radius ρ , and a velocity of V , the yaw rate of the vehicle is given by:

$$\psi = \kappa V = \frac{V}{\rho}$$

This gives an angular disparity error of:

$$\Delta\psi = \frac{V}{\rho}\Delta t$$

For a horizontal baseline system, this can be converted to units of pixels as follows:

$$\Delta\psi = \frac{V}{\rho}\Delta t\frac{\text{cols}}{\text{HFOV}}$$

Now, require that the disparity error be less than 10% of a pixel by solving for Δt :

$$\Delta t = \frac{\rho}{V}\frac{\text{HFOV}}{\text{cols}}\left(\frac{1}{10}\right)$$

This requirement often implies that cameras must be synchronized in image capture to the resolution of the MHz data clock.

Real Numbers

Using the HMMW with a 10 m/s second speed, executing a turn of the minimum radius of 7.5 meters, a standard 512 X 512 pixel CCD, and a 0.6 radian field of view gives a desynch threshold of 200 microseconds.

If the range pixel returns a value of R , then the velocity of the pixel due solely to this vehicle motion is given by:

$$V_{\text{pixel}} = \frac{RV}{\rho}$$

At this speed, the pixel moves through a distance equal to the map resolution over a time period of:

$$\Delta t = \frac{\delta}{V_{\text{pixel}}} = \frac{\delta \rho}{RV}$$

Therefore, the vehicle yaw must be sampled at a rate of:

$$f_{\text{heading}} = \frac{RV}{\delta \rho}$$

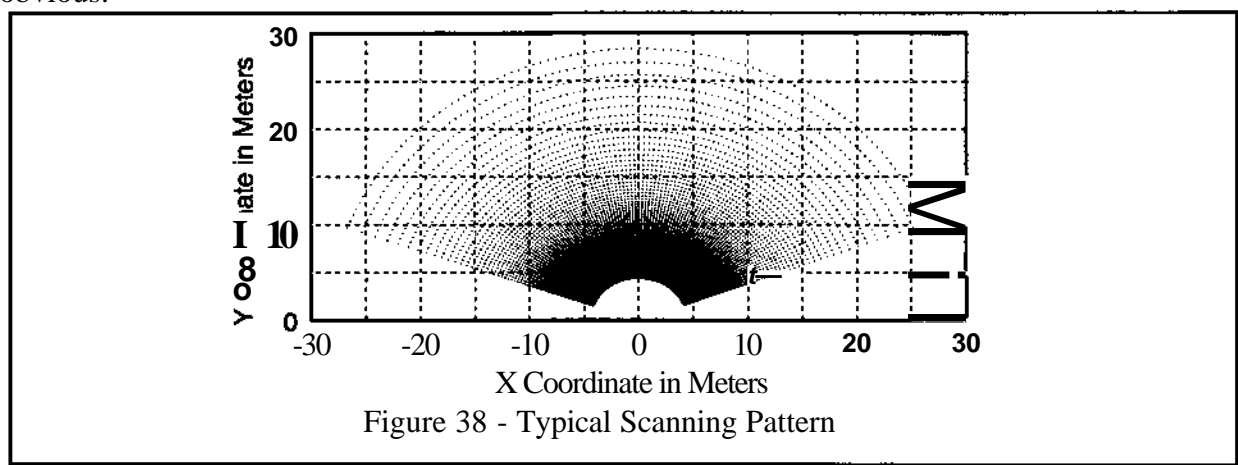
If this positioning bandwidth is not met, the terrain map will smear unacceptably and hazards will be localized incorrectly due to poor resolution in the computations

Section 7: Geometric Efficiency

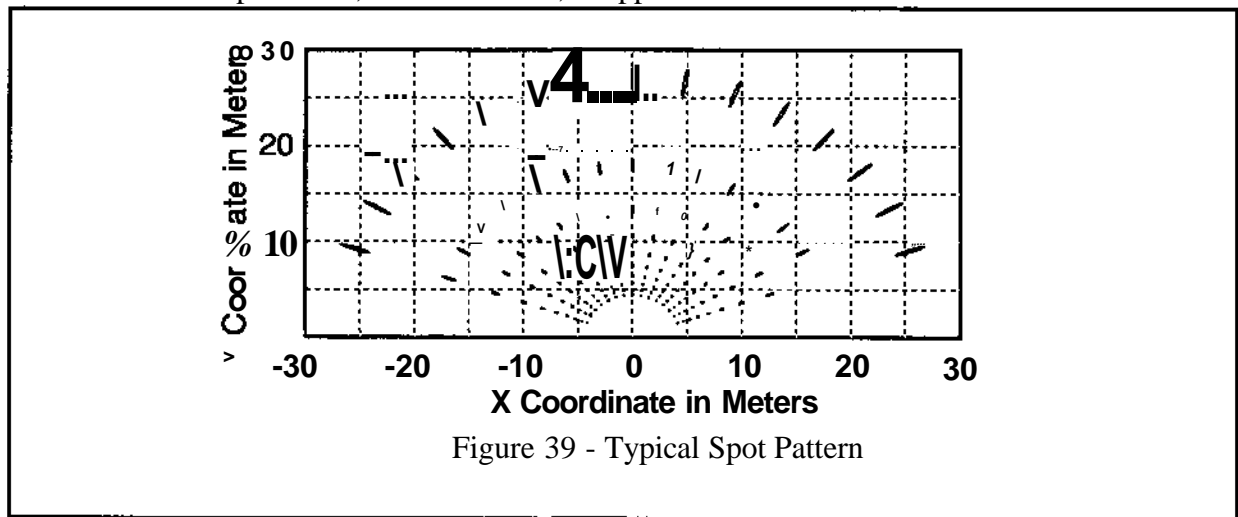
This section investigates the impact of typical imaging geometry on the efficiency with which a system can sense its environment at appropriate resolution.

7.1 Imaging Geometry

For typical imaging geometry, the density of range pixels on the groundplane varies significantly with both position and direction. Consider the following graph of the groundplane positions of each pixel from a range image for a typical rangefinder. A 10 m. grid is superimposed for reference purposes and every fifth pixel is shown in azimuth to avoid clutter. The density variation is more extreme than can be shown on paper. The variation of density with range is obvious:



The spot pattern is obtained by intersecting the ground plane with each pixel. Spots are shown below for the same sensor at one-tenth density in elevation and one-seventh density in azimuth: The variation in aspect ratio, as well as size, is apparent.



7.2 Scanning Density

In a single range image, the average number of range pixels that fall into a particular map cell is not necessarily one. A quantity called the **scanning density** σ_s can be defined which is the average number of range pixels per map cell over the entire field of view of a single image. The scanning density depends on the scanning pattern of the sensor as well as its position and orientation on the vehicle.

Real Numbers

For the ERIM scanner mounted on the **HMMWV** at 2.7 meters height and 16.5° tilt, using a map resolution of 1/6 m, the scanning density is 2.0. However, this figure is the average value so it does not reflect the considerable variation in density over the field of view. Recall from an earlier section that the maximum variation in density is the cube of the range ratio - or about 50.

The scanning density is purely a geometric matter. It is easy to compute as follows:

$$\sigma_s = \frac{\text{pixels}}{\text{Area}} = \frac{(\text{HFOV} \times \text{VFOV})}{[\text{IFOV} \times \text{IFOV}] \times (\text{AREA}/8)}$$

$$\text{AREA} = \frac{\text{HFOV}}{2\pi} \pi [R_{\text{max}}^2 - R_{\text{min}}^2]$$

$$\sigma_s = \frac{\text{HFOV} \times \text{VFOV}}{\text{AREA}} \times \frac{8}{\text{IFOV}^2}$$

$$\sigma_s = \frac{2 \times \text{VFOV}}{[R_{\text{max}}^2 - R_{\text{min}}^2]} \times \frac{8}{\text{IFOV}^2} = \frac{2h \times \frac{(R_{\text{max}} - R_{\text{min}})}{R}}{[R_{\text{max}} - R_{\text{min}}] [R_{\text{max}} + R_{\text{min}}]} \times \frac{8}{\text{IFOV}^2}$$

$$\sigma_s = \frac{2h}{(R_{\text{max}} R_{\text{min}}) [R_{\text{max}} + R_{\text{min}}]} \times \frac{8}{\text{IFOV}^2}$$

$$\sigma_s = 2 \left(\frac{h}{R_{\text{max}}} \right) \left(\frac{1}{\frac{R_{\text{max}}}{R_{\text{min}}} - 1} \right) \left(\frac{8}{R_{\text{min}}} \right)^2 \left(\frac{1}{\text{IFOV}} \right)^2$$

$$\sigma_s = 2h_{\text{min}} \left(\frac{1}{R + 1} \right) \left(\frac{8}{R_{\text{min}}} \right)^2 \left(\frac{1}{\text{IFOV}} \right)^2$$

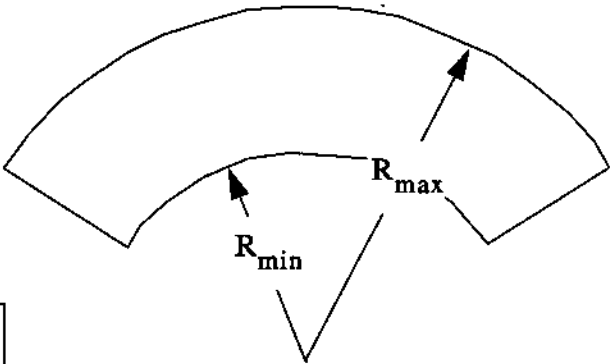


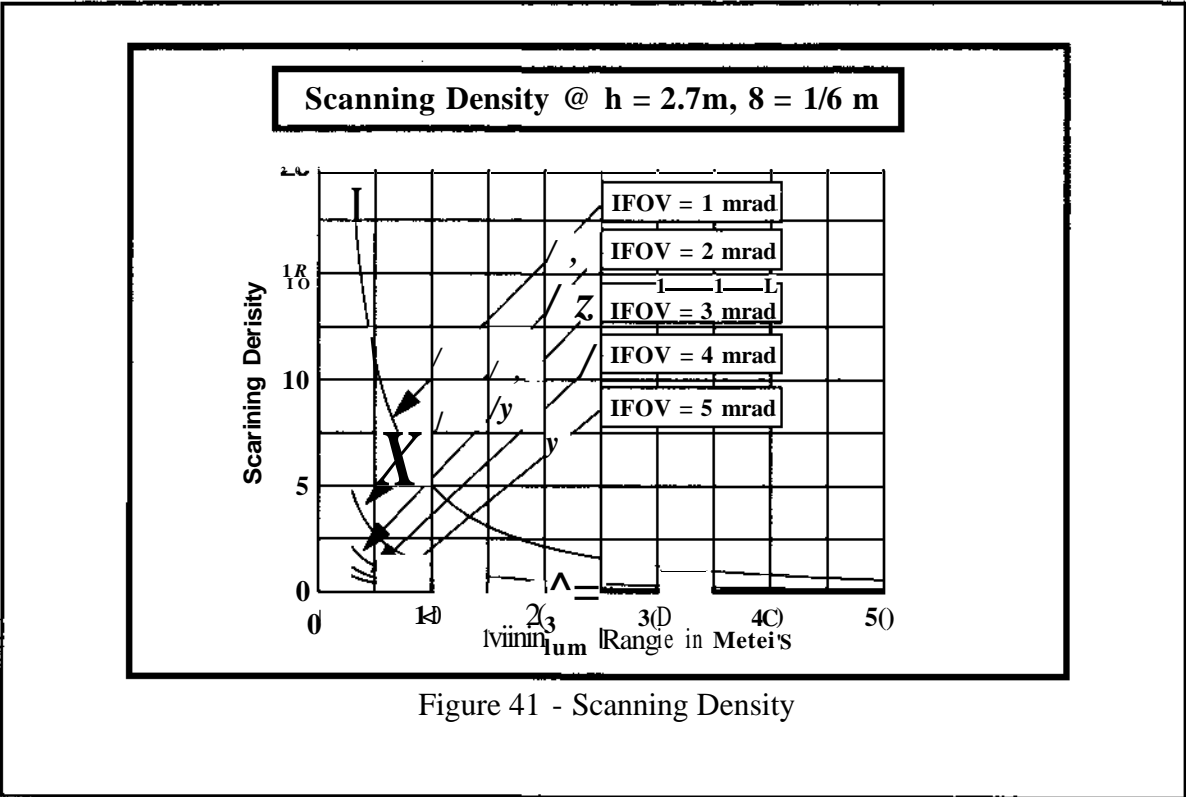
Figure 40 - Scanning Density Expression

For a given map resolution and IFOV, the scanning density depends only on the perception ratio and the range ratio and a new nondimensional ratio of map resolution to minimum range. Let this new variable be called the **normalized map resolution**.

$$\delta = \frac{\delta}{R}$$

The significance of the extra variable is that it is necessary to specify where the field of view is pointed in order to compute the area that it covers.

The scanning density is plotted below for a sensor height of 2.7 m, a map resolution of 1/6 m, and various values of the pixel size. The growth with IFOV is quadratic and the growth with decreased minimum range is also quadratic. Pixel sizes were chosen in the range necessary to identify step obstacles at the maximum range²⁵. Clearly, for any choice of parameters, there is a minimum range below which scanning density begins to increase very rapidly and significant computational bandwidth is wasted.



25. The situation is far worse if the beam dispersion is chosen to keep the spot size within the map resolution.

7.3 Imaging Density

It may be necessary to image each map cell more than once for several reasons:

- to ensure robust obstacle detection
- to track moving objects
- to track stationary landmarks for position estimation purposes

A quantity called the **imaging density** G_j can be defined which is the number of images that include a map cell in the field of view, averaged over all map cells. This quantity is independent of the scanning density by definition.

Real Numbers

For the ERIM scanner mounted on the HMMWV at 2.7 meters height and 16.5° tilt, at 5 m/s speed, the imaging density is about 6. So the vehicle sees an obstacle 6 times before it leaves the field of view of the sensor completely.

The imaging density depends on vehicle speed as well as the sensor configuration. A simple expression for it is available by considering linear trajectories of a given velocity. The vertical field of view of the sensor projects onto the groundplane to give a certain length in the downrange direction. Over a time period of the sensor frame period T_{image} (the inverse of the frame rate), the vehicle moves a distance given by its velocity. The governing relationship is given below:

$$\sigma_I = \frac{\Delta R}{VT_{image}}$$

but this is just the inverse of the **throughput ratio** as would be expected. The imaging density is plotted below versus speed for a sensor with a 0.4 sec frame period for some values of the vertical field of view.

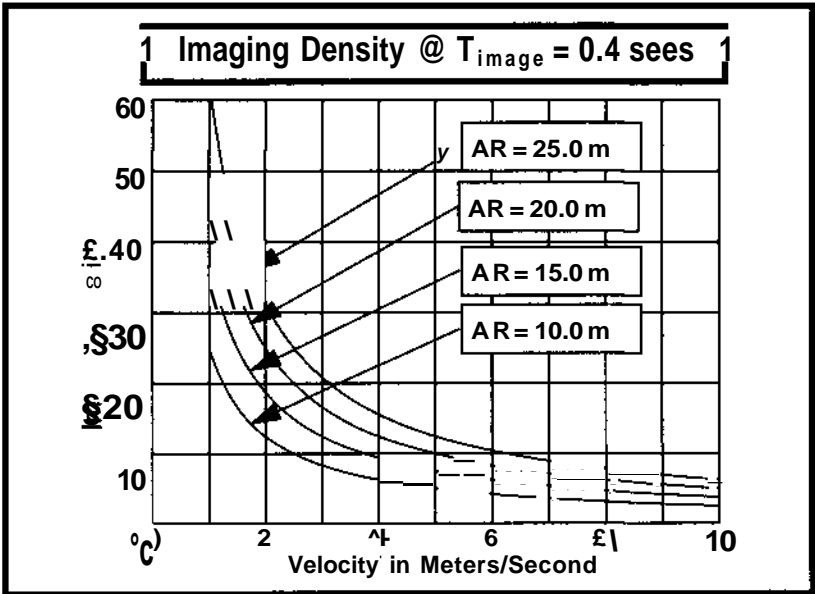


Figure 42 - Imaging Density

7.4 Geometric Efficiency

The scanning density and imaging density together give the average total number of image pixels that fall on a patch of terrain over several images. Putting these factors together, a **geometric efficiency** η_G can be defined as follows:

$$\eta_G = \frac{1}{\sigma_S \sigma_I}$$

In order to demonstrate the importance of the geometric efficiency, it is plotted below for the HMMWV using the ERIM rangefinder as currently configured. The configuration is maximum range 20 m, minimum range 4.5 m, beam dispersion 10 mrad, map resolution 1/3 m, sensor height 2.7 m:

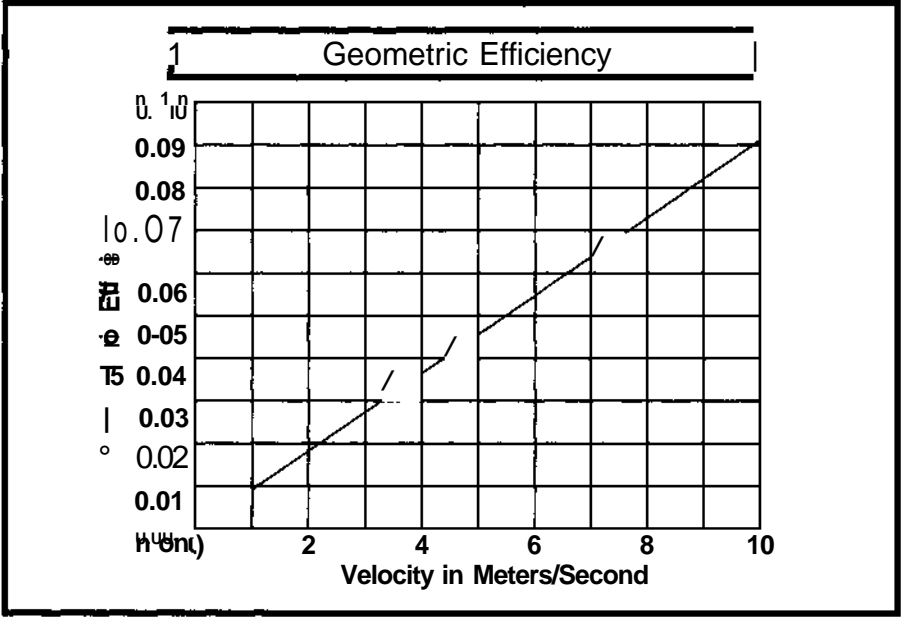


Figure 43 - Geometric Efficiency

Thus, at an operating speed of 3 meters/second²⁶, the **geometric efficiency** of the ERIM is a mere **2.5%**. *Equivalently, roughly 98pixels out of 100 are waste, providing no new information.*

26. This is the typical operating speed of contemporary off-road vehicles.

7.5 Adaptive Scan

Two techniques are available to alleviate poor geometric efficiency. **Adaptive sweep** was proposed earlier as a mechanism for minimizing the imaging density. Typical geometry implies that the area density of range pixels becomes more or less constant when this is done because the range ratio is fairly small. However, the shape of those uniformly distributed pixels is still very elongated. The mechanism for alleviating this problem, associated with the scanning density, is to elongate the pixel in the reverse direction in the image. This will be called **adaptive scan**. For contemporary sensors, adaptive scan can be implemented by filtering or subsampling the image in the azimuth direction. Ideally, a sensor would have nonsquare hardware pixels.

7.6 Acuity Nondimensionals

From the point of view of dimensional analysis, it is not so much the height of the sensor as the ratio of that height to the maximum sensor range that matters. This quantity was defined earlier as the **perception ratio**:

$$\bar{h} \equiv \frac{h}{R}$$

The quantity gives quick rules of thumb for computing laser spot aspect ratio and area. Let the ratio of maximum to minimum range be called the **range ratio**:

$$\bar{R} = \frac{R_{\max}}{R_{\min}}$$

Similarly, the **normalized range difference** is:

$$\overline{\Delta R} = \frac{R_{\max} - R_{\min}}{R_{\min}} = \frac{\Delta R}{R_{\min}}$$

These are obviously related by:

$$\overline{\Delta R} = \bar{R} + 1 \qquad \bar{R} = \overline{\Delta R} + 1$$

Recall that the quantity:

$$\eta_G = \frac{1}{\sigma_S \sigma_I}$$

is called the **geometric efficiency** and it expresses the average number of range pixels in a map cell when it is consumed by driving over it. The **scanning density** a_s is the average density of pixels per map cell over a single entire image. The **imaging density** a_x is the average number of images that fall on a patch of terrain.

Chapter 5: Fidelity

As was shown earlier, the **fidelity ratios** relate the configuration of the vehicle, the accuracy of its geometric models, and the density of hazards. This section analyzes these aspects of vehicle performance for typical vehicles.

Section 1: Modelling Dynamics and Delays

A software system is, of course, a disembodied entity which relies on sensors and effectors to connect it to, and permit it to influence its environment in robotics applications. In the context of high-speed motion, the time it takes to pass information into and out of the system becomes a significant factor. As a moving entity, any delays in time which are not accounted for are ultimately reflected as errors between both:

- what is sensed and reality, or
- what is commanded and reality

Time delays, also called **latencies**, may arise in general from several sources and all of these types of delay occur in a contemporary autonomous system:

- **Sensor dwell** is the time it really takes for a measurement to be acquired even though it is often a nominally instantaneous process.
- **Communication latency** is the time it takes to pass information between system processes and processors.
- **Processing latency** is the time it takes for an algorithm to transform its inputs into its outputs.
- **Plant dynamics latency** is the delay that arises in physical systems because they are governed by differential equations.

While delays affect response directly, they also affect the ability of the system to localize obstacles correctly. Delays themselves are not so much a problem as are *unmodelled delays*. Hence, *one of the impacts of high speed on the fidelity requirement is the need for high fidelity temporal models*. This implies several things:

- significant events must be time-stamped
- significant delays must be modelled
- i/o which is not correlated in time must be buffered

This section investigates these matters in the context of high-speed motion.

1.1 Latency Problem

Unmodelled latencies in both sensors and actuators cause the vehicle to both underestimate the distance to an obstacle and underestimate the distance required to react. This is indicated in the following figure:

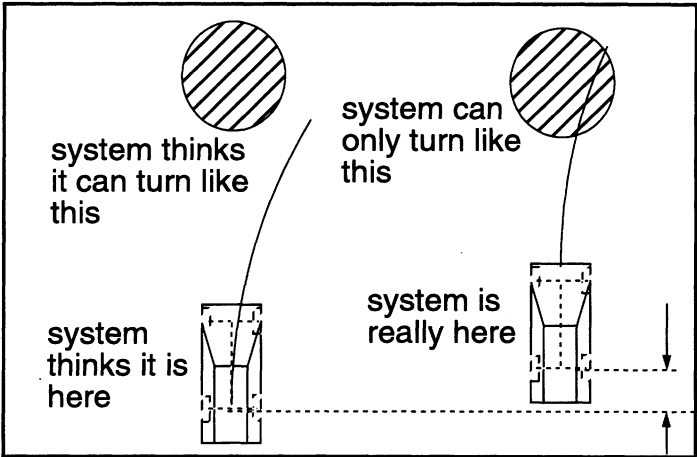


Figure 44 - Latency Problem

1.2 Minimum Significant Delay

Let a time delay of Δt occur which is not modelled by the system. If the vehicle travels at a speed V then the distance travelled is, naturally, $V\Delta t$. In order to guarantee correct localization of a range pixel, the **maximum fidelity** requirement must be met. The impact of any delay on the fidelity requirement can be expressed in terms of the **maximum fidelity ratio** thus:

$$\rho_{dx} = \frac{V\Delta t}{\delta}$$

and the **minimum significant delay** occurs when the ratio is 1, or when:

$$\Delta t = \frac{\delta}{V}$$

Real Numbers

At a mere 5 m/s, using a map resolution of 1/3 meter, this gives less than one-tenth of a second whereas the delays of real systems typically exceed this by a large factor.

1.3 Dynamic Systems

A matter similar to delay is the time constant of dynamic systems. Sometimes, a control algorithm actuates a derivative of the variable of interest and there is often a limit on the magnitude of the derivative that can be commanded. The net effect is a significant delay in the transfer function. In general, any system which commands a derivative of the controlled variable satisfies a differential equation at least as complex as:

$$\tau \frac{dy_{\text{response}}}{dt} + y_{\text{response}} = y_{\text{command}}$$

This is the equation of a **filter** - where T is called the **time constant**. When the time constant is the dominant component of actuator response time, it is also denoted as T_{act} .

Typically, the position and attitude of the vehicle are the variables of interest. For the Ackerman steer vehicle, the input commands are speed (the derivative of arc length) and curvature (the derivative of heading w.r.t. arc length) so *the Ackerman steering system is fundamentally a filter*.

1.4 Response Regimes

It is customary when describing dynamic systems to differentiate the time period when the system is attempting to track a command input from the time when it achieves the desired output. For a first order system, the typical response profile for a constant input command and zero initial conditions is as shown below. It can be shown that under these conditions, the output reaches about $(1-1/e)\%$ or 63% of the desired value after one time constant expires:

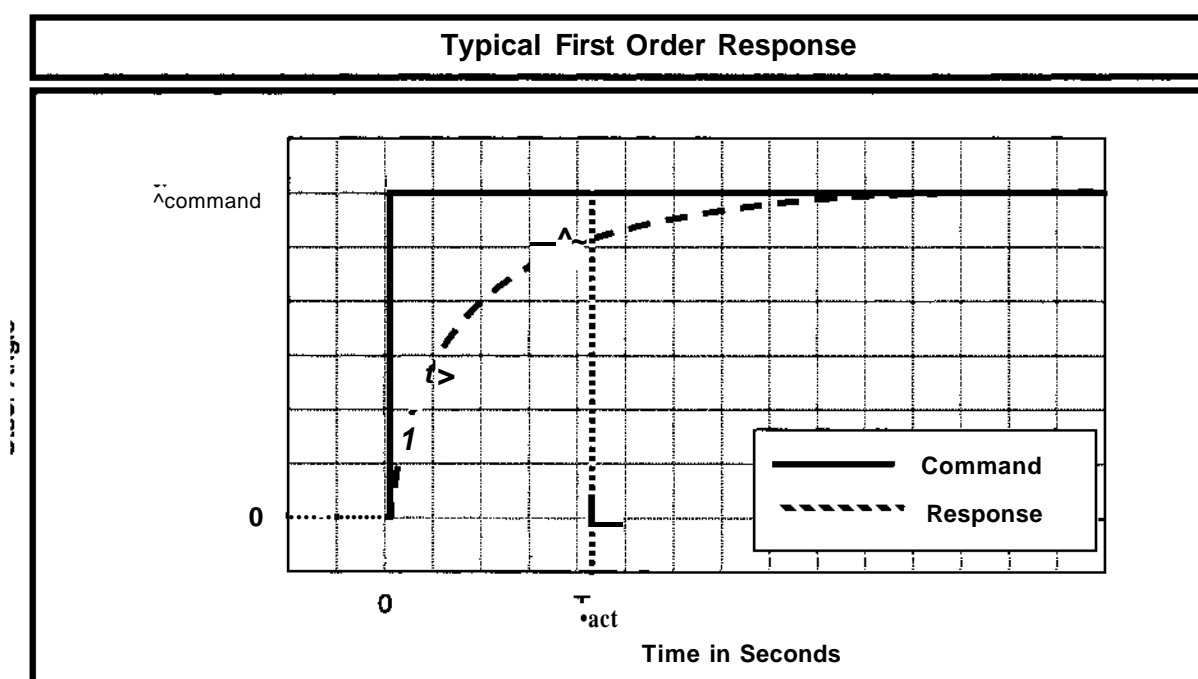


Figure 45 - Response Regimes

The time period when the system output is changing²⁷ is called the **transient regime** and the time period when it is not changing is called the **steady-state regime**.

1.5 Characteristic Times and Low Latency Assumption

The **characteristic time** of any element is the total delay, whatever its source, which relates the input to the associated correct, steady-state output. The total characteristic time of all information processing elements, hardware or software, and all energy transformation elements is the quantity which matters, so it is not correct to discount delays individually. To assume that delays are irrelevant is to assume that the characteristic time is zero. This **low latency assumption** is not correct for high-speed autonomy above some speed.

1.6 Normalized Time Constant

One of the most important quantities in the characterization of problems of planning and control is the **normalized time constant**. This quantity is a direct measure of the importance of dynamic modelling. It is typical of the planning approach to problems to abstract away the dynamics of the problem for reasons of efficiency. For example, in planning a path from one city to another, the time constant of the steering column can hardly be important. However, for obstacle avoidance purposes, the fact that the steering column will not even reach its commanded position before the obstacle is reached is a central concern. This spectrum can be formalized roughly with the normalized time constant:

$$\tau = \frac{\tau}{T_{\text{look}}} = \frac{T_{\text{act}}}{T_{\text{look}}}$$

where T_{look} is the **temporal planning horizon** or the amount of time the system is looking ahead in its deliberations.

When the normalized time constant is less than, say 0.1, dynamics are not important but when it approaches or exceeds unity, dynamics are a central issue. Perhaps the most important distinguishing characteristic of the problem of autonomous mobility is the fact that the temporal planning horizon is roughly equal to the reaction time so the system operates almost entirely in the transient regime.

1.7 Low Dynamics Assumption

The assumption that dynamics can be neglected in any particular part of a system will be called the **low dynamics assumption**. This assumption is related to the low latency assumption but it is more general since latency is but one kind of dynamic behavior that may be important to the overall behavior of a system.

27. Here, not changing does not mean constant because a sinusoidal output of constant amplitude and frequency is considered steady state.

1.8 Transience in Turning

When a vehicle executes a reverse turn, the actuator response can be divided into a transient portion and a steady-state portion as shown in the following figure. During the transient portion the steering mechanism is moving to its commanded position at a constant rate. This portion of the curve in the groundplane is a **clothoid**. During the steady-state portion, the curvature is constant, and the curve is a circular arc.

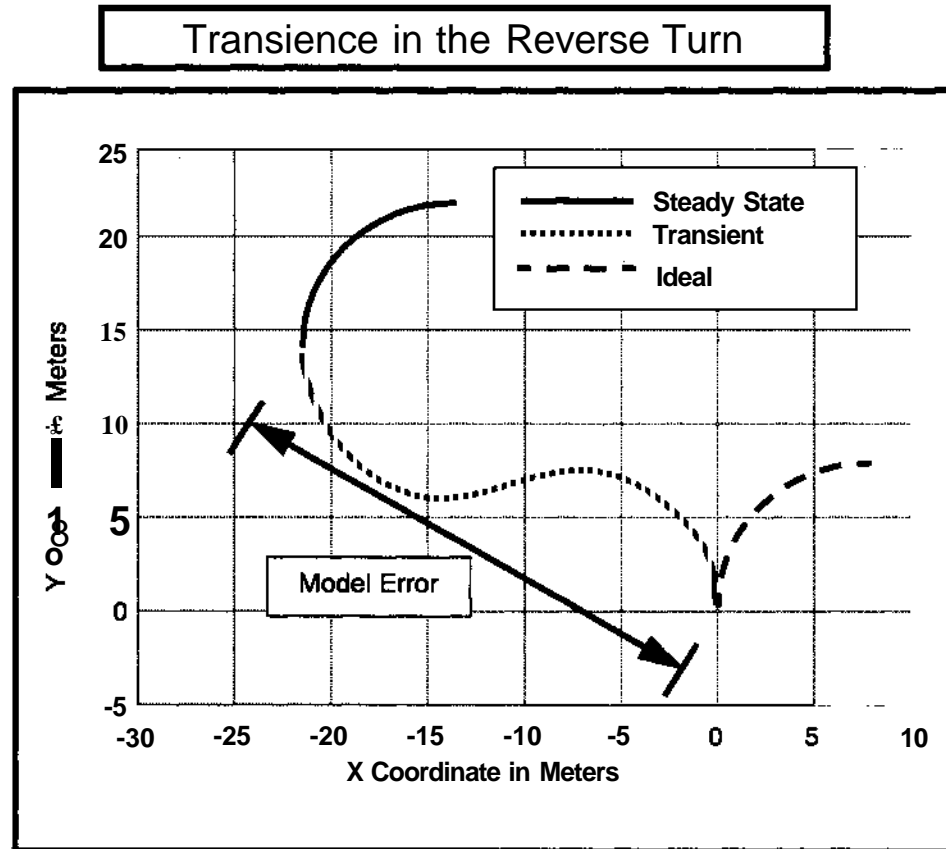


Figure 46 - Transience in the Reverse Turn

1.9 Heading Response

If the mechanism actuates curvature more or less directly²⁸, then the heading response curve is the direct integral of the steering mechanism position at constant velocity because yaw rate is given by:

$$\dot{\psi} = \kappa V$$
$$\psi(t) = \psi_0 + V \int_0^{T_{act}} \kappa(t) dt$$

This implies that the heading will grow quadratically, reach a maximum and descend back to zero exactly as the steering mechanism reaches its goal because the area under the curvature signal is zero as shown below:

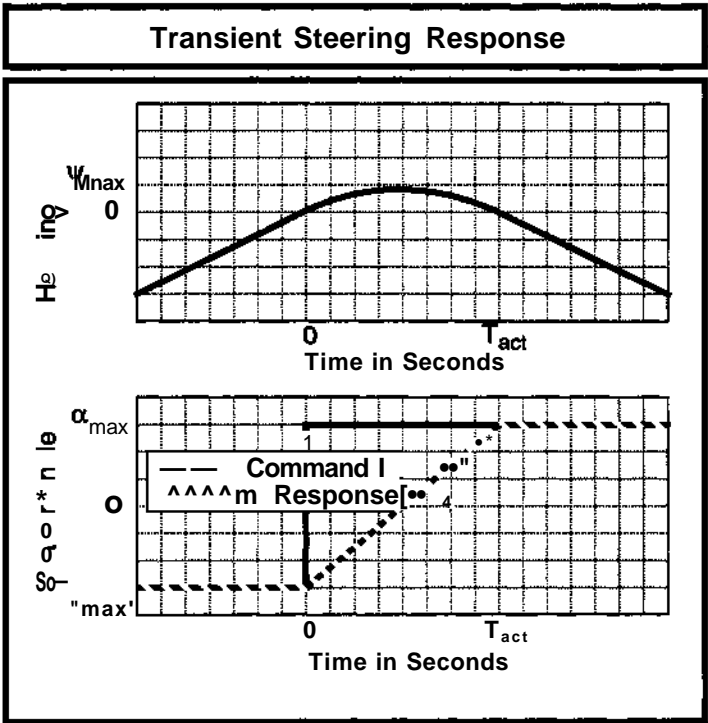


Figure 47 - Transient Steering Response

28. Ackerman steering actuates curvature more or less directly because although the relationship between steer angle and curvature is an arctangent, it is mostly linear over the domain of interest.

1.10 Nondimensional Transient Turning

Recall that the actuator reaction time is given by:

$$T_{\text{act}} = \frac{\Delta\alpha}{\dot{\alpha}_{\text{max}}} = \frac{2\alpha_{\text{max}}}{\dot{\alpha}_{\text{max}}}$$

Real Numbers

For the HMMWV, it takes 3 seconds to turn the front wheels through their 60° of travel.

The time required to turn through an angle $\Delta\psi$ at constant curvature is:

$$T_{\text{turn}} = \frac{\Delta\psi}{\dot{\psi}_{\text{max}}} = \frac{\Delta\psi}{\kappa_{\text{max}} V} = \frac{\Delta\psi \rho_{\text{min}}}{V}$$

Thus, a **transient turning coefficient** can be defined as the ratio of these two:

$$t_t = \frac{2\alpha_{\text{max}}}{\dot{\alpha}_{\text{max}}} \frac{\Delta\psi}{\psi} = \frac{2\alpha_{\text{max}} V}{\dot{\alpha}_{\text{max}} \Delta\psi \rho_{\text{min}}} = \frac{T_{\text{act}} V}{\Delta\psi \rho_{\text{min}}}$$

This nondimensional is a particular instance of the **normalized time constant**. It provides a measure of the importance of turning dynamics in a sharp turn. When it exceeds, say 0.1, it becomes important to explicitly consider turning dynamics. Note that the number increases for smaller constant curvature turns.

Real Numbers

For the HMMWV, this number is 1.27 at 5 m/s for a 90° turn. Hence, it is imperative to consider turning dynamics.

1.11 Command Following Problem

Another important aspect of the high curvature turn at speed is the raw error involved in assuming instantaneous response from the steering actuators. The difference between the two models is illustrated in the previous figure. The length of this vector can be approximated by:

$$s_{\text{error}} = T_{\text{act}} V$$

Thus, the modelling error associated with an ideal model of steering is equal to the reaction distance of the steering actuator.

Real Numbers

For the HMMWV, the model error of the ideal arc response is 15 meters at 5 m/s. Hence, the use of "arc" models of steering are fundamentally wrong at surprisingly moderate speeds.

To cast this result in terms of the fidelity ratio, consider the minimum fidelity ratio for an acceptable model error on the order of the wheel radius. Let this be called the **turning fidelity ratio**:

$$\rho_t = \frac{dx}{(r - W)} = \frac{T_{act} V}{(r - W)}$$

Real Numbers

For the HMMWV, the model error of the ideal arc response is 15 meters at 5 m/s, the wheel radius is 0.475 meters, and the vehicle width is 2.5 meters. This gives a fidelity ratio on the order of 7.5 whereas the requirement is that it be less than unity. The fidelity ratio indicates the importance of dynamic modelling.

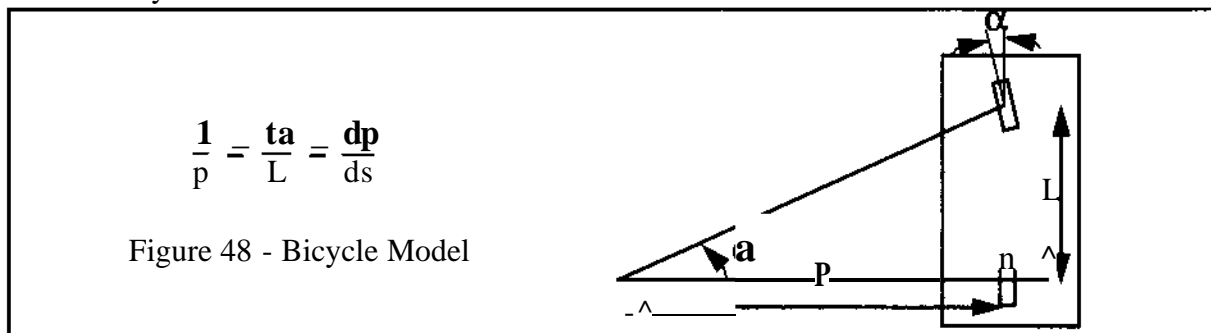
Section 2: Dynamics of Steering

An earlier section presented an analysis of the relative importance of computational reaction time and vehicle maneuverability on the **response ratio**. In that analysis, actuators were considered to respond instantaneously and perfectly to an input command - after some time delay had elapsed. While this is a useful theoretical approximation, and while it is a good model of braking, the same is not true of turning. Steering dynamics can only be modelled correctly by a differential equation. This section presents an accurate steering model for an Ackerman steer vehicle.

While this section is written specifically for the Ackerman steer vehicle, many of the conclusions apply in general because high speeds and rollover hazards limit the curvatures that a vehicle can safely sustain. There are three principle problems associated with the Ackerman steering mechanism. The first is steering dynamics, a **latency problem**, and the second is the nonintegrability of constraints, a **nonholonomy problem**. The third is the fact that the conversion from steer angle to curvature is **nonlinear**. A further problem is that attitude rate is *coupled* to both speed and steer angle.

2.1 The Bicycle Model

It is useful to approximate the kinematics of the steering mechanism by assuming that the two front wheels turn slightly differentially so that the instantaneous center of rotation can be determined purely by kinematic means. This amounts to assuming that the steering mechanism is the same as that of a bicycle. Let the angular velocity vector directed along the body z axis be called $\dot{\beta}$. Using the bicycle model approximation, the path curvature K , radius of curvature p , and steer angle α are related by the wheelbase L .



Where $\tan \alpha$ denotes the tangent of α . Rotation rate is obtained from the speed V as:

$$\dot{\beta} = \frac{d\beta}{ds} \frac{ds}{dt} = \kappa V = \frac{V \tan \alpha}{L}$$

The steer angle α is an indirect measurement of the ratio of $\dot{\beta}$ to velocity through the measurement function:

$$\alpha = \arctan\left(\frac{L \dot{\beta}}{V}\right) = \arctan(KL)$$

When the dependence on time of inputs and outputs is represented explicitly, this steering mechanism is modelled by a coupled nonlinear differential equation thus:

$$\frac{d\beta(t)}{dt} = \frac{1}{L} \tan[\alpha(t)] \frac{ds}{dt} = \kappa(t) \frac{ds}{dt}$$

2.2 Fresnel Integrals

The **actuation space** (A-space) of a typical automobile is the space of curvature and speed since these are the variables that are directly controlled²⁹. The **configuration space** (C-space) on the other hand is comprised of (x, y, heading) or perhaps more degrees of freedom in cartesian 3D. The mapping from A-space to C-space is the well-known **Fresnel Integrals** which are also the equations of **deduced reckoning** in navigation. For example, the following equations map A-space to C-space in a flat 2D world:

$$\begin{array}{ll} x(t) = x_Q + \int_0^t V(t) \cos(\psi(t)) dt & \psi(t) = \psi_Q + \int_0^t \kappa(s) ds \\ Y(t) = y_0 + \int_0^t V(t) \sin(\psi(t)) dt & \psi(t) = \psi_Q + \int_0^t \kappa(s) ds \\ \psi(t) = \psi_0 + \int_0^t \kappa(s) ds & \psi(t) = \psi_Q + \int_0^t \kappa(s) ds \end{array}$$

The inverse mapping is that of determining curvature $\kappa(t)$ and speed $V(t)$ from the C-space curve. Notice that C-space is three-dimensional while A-space is two-dimensional. Not only is the problem of computing this mapping a nonlinear differential equation, but it is underdetermined or **nonholonomic**. This is a difficult problem to solve and, from a mathematics standpoint, there is no guarantee that a solution exists at all. Practical approaches to the C-space to A-space mapping problem often involve the generation of curves of the form:

$$\kappa(s) = \kappa_0 + as$$

where s is arc length³⁰. These curves are linear equations for curvature in the arc length parameter and are known as the **clothoids**. There is a growing body of literature on the generation and execution of clothoid curves in indoor, non-omnidirectional autonomous vehicle applications. The generation of clothoids can be computationally expensive. Their generation can also be unreliable if the algorithm attempts to respect practical limits on the curvature or its derivatives.

29. The choice of what is to be considered the controlled variable depends on the level of abstraction. At some boundary in the system, a speed is asked for and an error signal on speed is formed. Lower level loops may form errors on other quantities.
 30. Mathematically, the clothoid generation process is equivalent to the power series solution of a differential equation. The idea is to assume a power series solution, substitute it into the original differential equation and then solve the simpler recurrence relation which results. This is how many higher transcendental functions (the Bessel function, for example) are defined.

2.3 Dynamics of the Constant Speed Reverse Turn

The limited rate of change of curvature for an Ackerman steer vehicle is an important modelling matter at even moderate speeds. A numerical feedforward solution to the dead reckoning equations was implemented in order to assess the realistic response of an automobile to steering commands. It was used to generate the following analysis. The maneuver is a reverse turn. The following figure gives the trajectory executed by the vehicle at various speeds for a 3 second actuator delay.

For a vehicle speed of 5 m/s, a kinematic steering model would predict that an immediate turn to the right is required to avoid the obstacle. However, the actual response of the vehicle to this command would cause a direct head-on collision. Any planner must explicitly account for steering dynamics - even at low speeds, in order to robustly avoid obstacles.

There are two fundamental reasons for this behavior. First, steering control is *control of the derivative of heading*, and any limits in the response of the derivative give rise to errors that are integrated over time. Second, curvature is an arc length derivative, not a time derivative. Hence the heading and speed relationships are *coupled differential equations*. The net result is that *the trajectory followed depends heavily on the speed*.

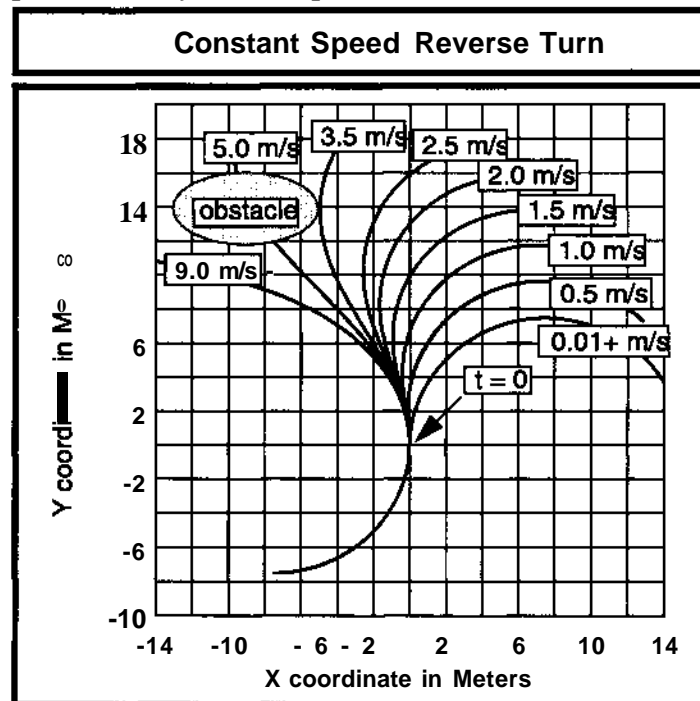


Figure 49 - Constant Speed Reverse Turn

Feedforward of dynamics is necessary for stable control. If the vehicle decided to turn slightly right at 5 m/s speed, position feedback would indicate that the vehicle was not turning right. Any feedback control law which attempted to follow the ideal commanded arc would continue to increase the turn command while the steering servo tries to turn right. This overcompensation will eventually lead to the maximum turn command being issued³¹ although a slight turn was commanded. Acceptable control is not possible without knowledge of these dynamics.

31. This is a major reason for the poor reliability of kinematic arc-based planners at higher speeds.

2.4 The Clothoid Generation Problem

The previous graph investigated the variability of the response to a steering command at various speeds. Consider now the response at a single speed to a number of steering commands issued at a speed of 5 m/s. Again using the reverse turn at $t = 0$, the response curves for a number of curvature commands are as shown in the figure below:

The vehicle cannot turn right at all until it has travelled a considerable distance. Further, a configuration space planner which placed curve control points in the right half plane would consistently fail to generate the clothoid necessary, if it attempted to model the steering dynamics, *because the vehicle fundamentally cannot execute such a curve*³². If the clothoid generator did not model such limits, the error would show up as instability and ultimate failure of the lower levels of control to track the path. The x-y region bounded by the curves is the entire region that the vehicle can reach.

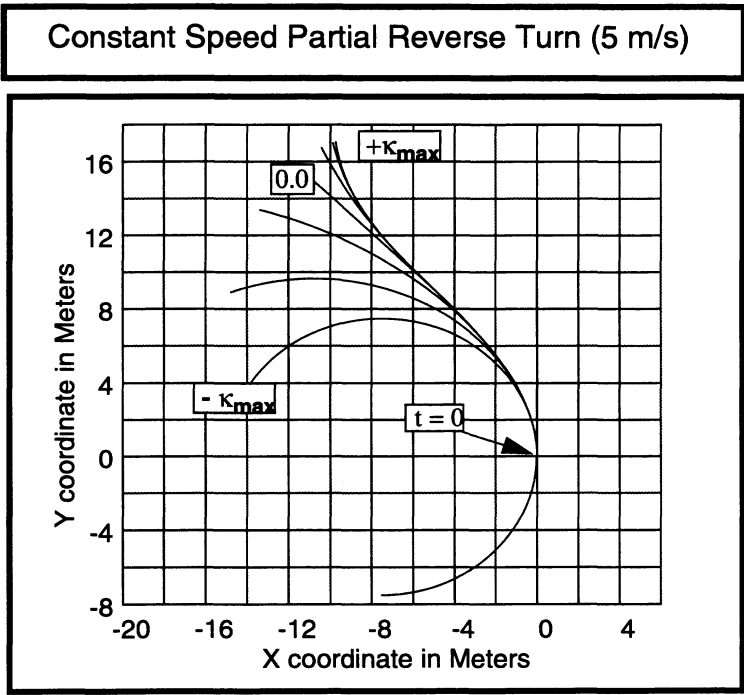


Figure 50 - Constant Speed Partial Reverse Turn

The only valid model of this system at even moderate speed is a coupled system of nonlinear differential equations. From the perspective of trajectory generation, it is advisable not to attempt the C-space to A-space transform in any form - particularly the generation of clothoids³³.

32. Thus, it is no accident that kinematic planners get “confused” as soon as a wide turn is executed. The incorrect low latency assumption in its particular form as a kinematic steering model is the reason for this behavior. The behavior will not arise if sharp turns are not attempted.
33. Of course, the fundamental issue is that of attenuation. Clothoid generation is feasible at low speed and for moderate turns at any speed.

2.5 Configuration Space

While it is difficult to compute the shapes of regions in configuration space in closed form, it is relatively easy to write a computer program to enumerate all possibilities and fill in boxes in a discrete grid which represents C-space at reduced resolution. The three dimensional C-space for an Ackerman steer vehicle for a 4.5 m/s impulse turn was generated by this technique.

The results are plotted below in heading slices of 1/16 of a revolution. Symmetry generates mirror images along the heading axis, so two slices are plotted on each graph. The maneuver is an impulse turn from zero curvature to the maximum issued at time $t = 0$. A dot at a particular point (x,y) in any graph indicates that the heading of the slice is obtainable at that position. There are 16 slices in total of which 6 are completely empty (i.e the vehicle cannot turn around completely in 20 meters). The total percent occupancy of C-space is the ratio of the total occupied cells to the total number of cells. This can be computed from the figure to be 3.1%.

So 97% of the C-space of the vehicle is empty if the limited maneuverability of the vehicle is modelled. The maneuverability is limited by both the nonzero minimum turn radius and the steering actuator response. The impulse turn is the best case. If the initial curvature was nonzero, the percentage occupancy is even less.

The occupancy of C-space does not account for higher level dynamics. There are severe constraints on the ability to "connect the dots" in these graphs, so the total proportion of feasible C-space paths is far lower.

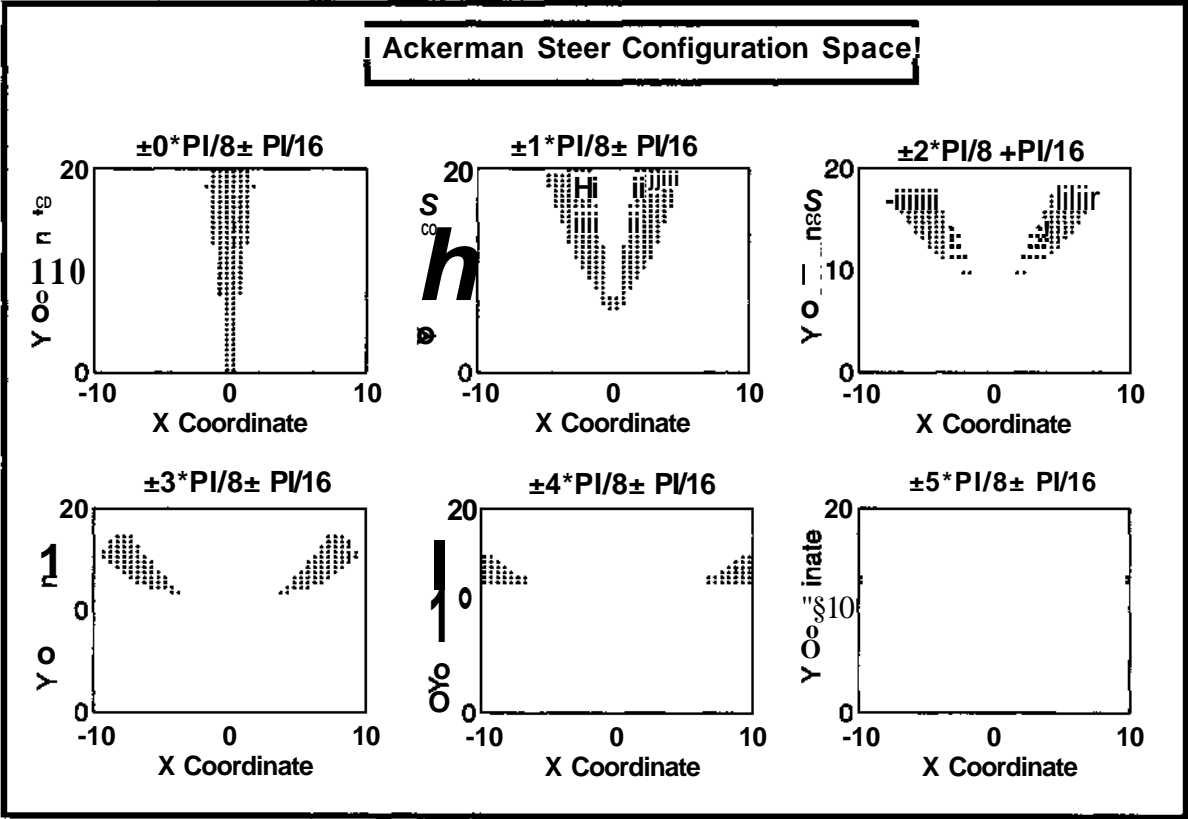


Figure 51 - Ackerman Steer Configuration Space

Section 3: Positioning Fidelity

In order to correctly localize hazards, the entire system geometric model must be sufficiently accurate. This implies that the vehicle location sensors and the perception sensors have accuracy requirements. This section investigates the first of these.

3.1 Absolute Attitude Accuracy Requirement

In order to compute static stability, a system fundamentally needs to know the projection of the body vertical axis onto the gravity vector, and this generates the need for:

- a sensor to measure 3D attitude at time t
- an *absolute angular accuracy* requirement on that sensor
- a 3D model of vehicle attitude and terrain following which allows the system to predict the presence of the hazard along candidate trajectories

A rough terrain system must both measure in 3D and think in 3D. An absolute accuracy on pitch and roll indications is necessary. As a rough rule of thumb, it must be an order of magnitude less than the minimum angle of rotation which brings the center of gravity outside the horizontal support polygon of the wheels. This arises again from the Nyquist sampling theorem and a requirement to distinguish hazard from non hazard as an angular proximity measurement. It can be called the **attitude acuity rule**.

3.2 Rigid Terrain Assumption

In order to predict tipover before it occurs, the system requires a model which maps the vehicle position in the terrain map onto an attitude. This requires some assumptions about the ability of the terrain to sustain compression and shear loads as well as a model of the vehicle suspension (which may be a trivial, rigid model). If the terrain is assumed to always provide the support loads required, without deformation, a **rigid terrain assumption** is being adopted.

3.3 Rigid Suspension Assumption

The suspension model allows the system to associate a unique vehicle attitude with each position, and permits the computation of any volumetric intersection between the vehicle body and the terrain. If it is assumed not to deflect at all, a **rigid suspension assumption** is being adopted.

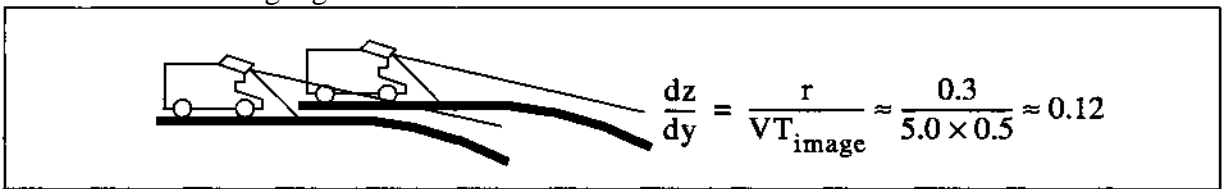
3.4 Image Registration Problem

A well-known problem in outdoor range image-based navigation is a special case of the **fidelity problem** known as the **image registration problem**. A fidelity requirement on the measurement of environmental geometry is that the relative error in elevation for the same spot on the ground between two consecutive images not exceed the acuity requirement. Otherwise, artifacts of systematic sensor error will appear as phantom obstacles in the terrain map. There are many potential sources for such systematic errors including:

- systematic range sensor errors
- range sensor miscalibration³⁴
- vehicle kinematic miscalibration
- systematic position and attitude sensor errors
- position and attitude sensor miscalibration

3.5 Linear Relative Accuracy Requirement

From the point of view of the impact of the fidelity requirement on linear position estimates, the relative accuracy required is related to the vehicle vertical error excursion between images as shown in the following figure:



Thus, it requires only 12% *relative* position accuracy to avoid an artifact that is within the maximum acuity limit at 5 m/s.

Hence the **image registration problem** ought not to exist at all (on this basis) unless there are gross systematic perception errors, overall miscalibration errors, or some inherent geometric sensitivity.

3.6 Angular Relative Accuracy Requirement

The relative angular accuracy follows directly from the acuity limit itself. The relative angular accuracy required of a position estimation system is on the order of the angle subtended by a pixel. These last two requirements are well within the performance specifications of available hardware.

34. Cartesian bias is the only form of miscalibration which can be ignored in a relative error model. Even small angular or range errors generate nonconstant errors in elevation over the sensor field of view.

Section 4: Perceptual Fidelity

Correct localization of hazards also depends on the accuracy of the range image itself. This section investigates the effect of imaging geometry on the accuracy of a sensor.

4.1 Incidence Sensitivity Problem

The small incidence angle associated with large lookahead distance creates a severe sensitivity problem in the **localization** of obstacles. Consider again the downrange projective differential imaging kinematics expressions.

$$dy = \frac{R^2 d\theta}{h} \qquad \therefore \frac{dy}{d\theta} = \frac{R^2}{h}$$

Therefore range resolution grows with the square of range for typical imaging geometry. This fact is related to the minimum acuity requirement. The footprint of a range pixel is elongated in the downrange direction, and the returned range value will have a large random component when the range itself is large.

Real Numbers

For the HMMWV, a 30 meter range pixel of 10 mrad width is elongated to 3.3 meters. Range measurement noise can be expected to be some fraction of this.

4.2 Attitude Sensitivity Problem

Notice that the vertical error excursion of a pixel due to a small error in vehicle pitch indications is linear in the measured range because:

$$dz = R d\phi$$

Unless the relative angular attitude accuracy is sufficiently high and the positioner bandwidth is sufficiently high, phantom obstacles will be generated. Another source of these errors is relative motion between the vehicle body and the sensor. Sensor shock mounts should be designed to remove high frequencies, particularly in pitch, from this relative motion.

Chapter 6: Interactions

The fundamental requirements of timing, speed, resolution, and accuracy are largely independent in concept but they become related as soon as it is insisted that they be met *simultaneously*. This section will investigate the interactions of requirements at a more fundamental level. In doing so, it will establish the need for:

- a feedforward control approach to trajectory generation and tracking
- an adaptive approach to environmental perception
- a real-time approach to obstacle detection

Section 1: Rationale for a Feedforward Approach

The **dynamics constraint** is a catchall term to express the fact that system behavior is governed by a differential equation and the realities of the latencies of the computer system on which it runs. Any response trajectory which satisfies these set of constraints is said to be **dynamically feasible**. This section discusses these issues and their impact on the problem of intelligent mobility.

For some actuators, such as steering, response cannot be modelled as a kinematic function of the input command. These systems require differential equation models. The solution of such equations for the purpose of predicting response is called **feedforward**. Feedforward is investigated in this section.

1.1 Command Following Problem

It is clear from Figure 45 that filters generally do not follow the commands that are given to them. This ultimately is a matter of physics and Newton's laws of motion because any mass filters input force through one time integral to get velocity and a second time integral to get position³⁵. Of course, the study of how to improve command following in various ways through the introduction of extra energy is the province of **feedback control**.

This problem of plants not doing what they are told will be generally referred to as the **command following problem**. In the case where the poor command following is due to latency, the problem has been called the **latency problem**. In plain terms, a dynamic system generally cannot do what it is told unless the agent generating the command understands the limitations of the plant being controlled.

1.2 Modelling Problem

Given that the response of a dynamic system is a matter of unalterable physics, there remains the question of whether or not a sufficiently accurate model can be used to predict response. The problem of the generation of such a model will be called the **modelling problem** and getting it calibrated to be sufficiently accurate has already been called the **fidelity problem**.

35. Though Newton certainly did not put it this way, *all the world's a filter*.

It is very important to distinguish the command following and modelling problems because useful controllers can be constructed which still inevitably exhibit poor command following provided they are sufficiently well modelled.

1.3 Dynamics Feedforward

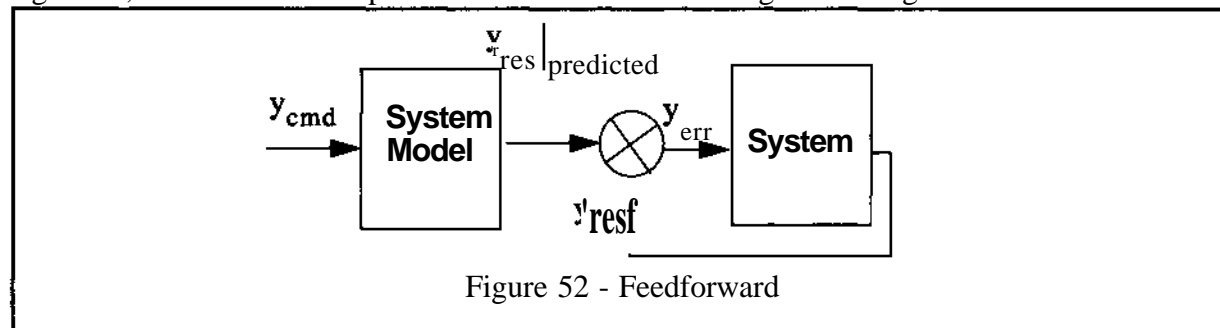
The control of vehicle speed and curvature for traditional steering mechanisms can cause massive errors in feedback controllers if they expect good command following because it has already been shown that these actuators operate solely in the transient regime. In the face of such dynamics, two general conclusions can be made:

- The system fundamentally cannot follow commands given to it above some frequency.
- Error signals formed directly on the controlled variable will cause control loop instability.

The traditional technique used to control highly latent systems is **feedforward**. The philosophy of feedforward is to:

- Accept that the system cannot be configured as a high-fidelity command follower.
- Form an error signal over *system response* and not commands by predicting the response to commands and comparing this to the feedback.

In general, a feedforward loop would look like the following block diagram:



where the system box may include lower levels of control as well as the real system.

1.4 Impact of Feedforward Dynamics on Computational Complexity

There is a positive side to the problem of controlling a highly latent system. While large delays cause problems that require feedforward, they are somewhat beneficial with respect to reducing the bandwidth requirements of control loops³⁶. In the context of the numerical solution of a differential equation, or in the context of a digital feedback control law, there is little to be gained by employing a high bandwidth servo to control a highly latent system, or at least, there are rapidly diminishing returns available as bandwidth increases.

The system will respond no faster than its time constant (related to the actuator power limit) predetermines. In control systems terms, the plant will not pass frequencies above its own natural frequency so high frequencies can be eliminated at the input side of the model. Increasingly higher loop bandwidth reaches an eventual point of diminishing returns in a throughput limited system where the cycles can be put to better use elsewhere³⁷.

36. Nyquist's sampling theorem of course says that sampling frequencies above twice the highest frequency in the sampled signal provide no new information.

In the context of obstacle avoidance planning, this issue appears in the form of **adaptive regard** because the limited ability of the vehicle to respond is exploited to reduce the complexity of motion planning search.

In the context of goal-seeking, this issue appears in the form of a relative soft goal-seeking loop. The fidelity of the feedforward transfer function is far more important to command following than the loop bandwidth because the plant will not pass high frequency command updates anyway.

1.5 Impact of Dynamics on Planner/Controller Hierarchy

The impact of these issues on the traditional control hierarchy is that the hierarchical view of planners commanding paths from controllers through some high-level interface will not work above some critical speed. The system will be unstable, and command-following fidelity will fail to meet the fidelity requirements of the problem as was shown earlier because the **turning fidelity ratio** significantly exceeds unity.

In high speed work, the emphasis must shift from the traditional AI hierarchy view to the tightly-coupled real-time control view. This point will be the central argument behind the design of the planner of the navigator where it will appear in the form of the tactical control layer to mediate between the other two layers.

1.6 A Real Time Control View of High-Speed Autonomy

In general, the path planning problem of autonomous navigation can be cast into one of searching some space of alternatives for those paths which meet two constraints:

- they must be safe from dangerous hazards (satisficing)
- they must be reachable by the vehicle (feasible)

On serial computers, these two constraints must inevitably be applied in the order stated or the reverse order. However, while the local result is the same, the order of application of the constraints can have a large impact on both the efficiency of the computation and the overall robustness of the system. The *hierarchical view* of the robot navigation problem is to check for collision first and give responsibility for execution of the path to a control algorithm placed lower in the hierarchy. The *control systems view* of the problem is to generate feasible paths first and then check for collisions. Both views are equally valid and have their domains of applicability. However, the hierarchical view is not optimal in high-speed autonomy.

It is not efficient to plan in C-space because too many solutions will be generated that do not satisfy the actuator dynamics constraint. C-space is almost completely *degenerate* in the heading dimension³⁸.

37. Consider the solution to the first-order filter equation presented earlier, for time steps below about 10% of the time constant, the solutions are indistinguishable from a relative error standpoint.

38. It is not surprising that this has occurred. Most research on the AI approach to path planning has been geometric in character - the implicit assumption being that dynamics could be safely ignored. Indeed they can be at slow speed.

The search space for planning purposes is degenerate because heading is practically not an option at all and vehicle position is confined to a narrow cone. Thus *the configuration space for an Ackerman steer vehicle is degenerate above about 5 meters/sec speeds* when typical steering column response is considered. There is effectively no state space to be searched and search based planners would accomplish nothing useful.

Explicit enumeration of the few alternatives that exist is the efficient method of planning paths. The planning problem degenerates into a simple decision process, and quite often, even though some regions in front of the vehicle may be clear of obstacles, the vehicle cannot turn fast enough to reach them and the only alternative is to stop.

An efficient method of planning and evaluating paths would be based on, not candidate paths, but candidate command signals *expressed in actuation space* (in terms of curvature and speed as a function of time). Such a system would use a high-fidelity feedforward simulator to solve the differential equations of motion in order to determine the exact response to these candidate commands. This strategy has the following advantages:

- The paths generated *meet the mobility constraints of the vehicle by construction*, so the difficult and often impossible problem of conversion from C-space to A-space is completely avoided. Instead, the reverse process of dead reckoning is used in the simulator. This is a kind of feedforward.
- The paths coarsely span the entire region of space that the vehicle can reach, so no real alternatives are missed within the fidelity requirement.

Section 2: Rationale for Adaptive Perception

An autonomous vehicle must satisfy all component requirements of guaranteed safety simultaneously and these requirements are all interrelated. For example, sensor range depends on speed through the response requirement and resolution depends on range through the acuity requirement and throughput depends on resolution, so throughput depends on speed. This section will quantify this relationship between throughput and speed.

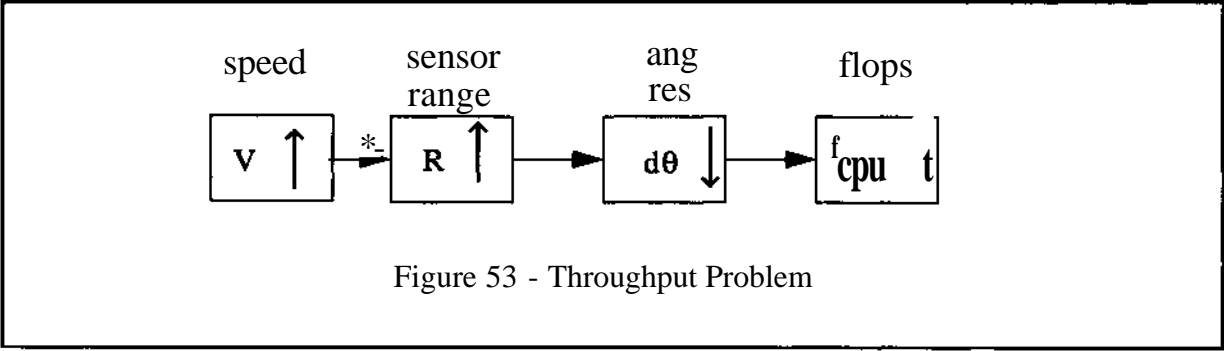
2.1 Throughput Problem

The throughput required to process an image depends on the number of pixels in the image. The number of pixels depends on the field of view and resolution. Resolution depends on the acuity requirement which implies it depends on range. The response requirement implies that range depends on speed so that resolution depends on speed. The throughput requirement also implies that field of view depends on speed. So ultimately, throughput can be expressed solely in terms of speed.

With an analysis of response and acuity it is possible to analyze the computational complexity of perception. In intuitive terms, guaranteed response implies that throughput is proportional to a high power of velocity because:

- Maximum range increases quadratically with speed (because braking distance does).
- Pixel size decreases quadratically with maximum range (in order to resolve obstacles).
- Throughput increases quadratically with pixel size (assuming fixed field of view).

This relationship is indicated in the following figure:



When throughput is limited, this relationship gives rise to a trade-off between speed and resolution. Naive analysis suggests that the problem of high-speed navigation is nearly impossible, because the necessary throughput is impractical. This will be called the **throughput problem**.

2.2 The Illusion

From an image processing perspective, the throughput problem appears to be impossible. Consider that contemporary rangefinders are 10 mrad resolution and many researchers believe that 1 mrad or so is needed to resolve obstacles. A ten-fold increase in resolution is a hundred-fold increase in pixels and a hundredfold increase in required throughput. Today, it is not possible to process 10 mrad images fast enough on a 10 Mflop processor. Therefore, if resolution were increased tenfold, it would be impossible to process 1 mrad images on a 1 Gflop processor. Brute force is not the elegant way to solve this problem.

On the other hand, the raw requirement is the throughput requirement, and this is trivial to meet. Consider that a 5 m/s vehicle covers about 6 map cells between images at 2 Hz, so there is band in the image about six pixels wide which would supply exactly the needed steady state throughput.

This section will show that *the throughput problem is an illusion* which arises from an image processing view of the problem and that simple adaptive techniques can solve it completely at contemporary sensor resolutions.

2.3 Adaptive Perception

It is possible to solve the throughput problem while simultaneously guaranteeing safety as efficiently as is theoretically possible by employing three principle mechanisms:

- **Adaptive Lookahead** is a mechanism for guaranteeing that the vehicle can respond to any hazards that it may encounter at any speed.
- **Adaptive Sweep** is a mechanism for guaranteeing barely adequate throughput and the fastest possible reaction time. In this way, speed is maximized.
- **Adaptive Scan** is a mechanism for ensuring barely adequate resolution that is as constant as possible over the field of view. In this way, speed is maximized without compromising robustness of the system.

Together, these mechanisms can increase the efficiency (measured in terms of range pixel throughput) of a system by four orders of magnitude at 20 mph while simultaneously making it considerably more robust.

2.4 Assumptions of the Analysis

The following subsections will analyze the throughput problem in terms of the design of a fixed vehicle which is optimized for some maximum speed. The pixel size is permitted to change with speed, so the graphs represent the variation of system designs versus speed and not the throughput requirement for a single design as it drives faster.

The estimates that are produced are underestimates for many reasons:

- They are based on an oversampling factor of 1. A practical factor is at least 3. This implies that the results must be multiplied by the square of 3, or 9.
- Minimum acuity will be used because this is actually the most stringent requirement beyond some range.
- The maximum range that is chosen is based on the stopping distance. Actually it is the minimum range which should be set to the stopping distance, but the approximation is useful.
- Braking is chosen as the obstacle avoidance maneuver. This is viable for a system which stops when a hazard is detected. However, when a vehicle turns to avoid obstacles, sensor lookahead must exceed the stopping distance by a large factor.
- The processor load is assumed to be 50 flops per pixel when experience suggests that up to ten times this is required in a practical system.
- The graphs estimate perception geometric transform processing only. Planning, position estimation, and control are not included at all.
- Horizontal field of view is fixed at 80°, 120°, 170°, and 215° for each increasing reaction time respectively, based on earlier analysis.
- Frame rate is set to 2 Hz.

2.5 Common Throughput Expression

Recall from previous analysis that the throughput required from the computer for perception processing can be written as:

$$f_{\text{cpu}} = \frac{f_{\text{pixels}}}{\eta_s} = \frac{1}{\eta_s} \frac{\Psi}{(\text{IFOV})^2}$$

This can be written in terms of field of view and frame rate as follows:

$$f_{\text{cpu}} = \frac{1}{\eta_s} \frac{\text{HFOV} \times \text{VFOV} \times f_{\text{images}}}{(\text{IFOV})^2}$$

When it is necessary to employ a nonsquare pixel size, the horizontal and vertical pixel dimensions can be differentiated as follows:

$$f_{\text{cpu}} = \frac{1}{\eta_s} \frac{\text{HFOV} \times \text{VFOV} \times f_{\text{images}}}{\text{IFOV}_H \text{IFOV}_V}$$

2.6 Basic Mechanism

The basic mechanism for generating a complexity estimate is as follows:

- Choose an angular resolution that is consistent with the need to resolve obstacles at the maximum range (guaranteed detection).
- Choose a maximum range consistent with the need to stop if necessary (guaranteed response).
- Choose a fixed field of view and frame rate (because sensors are designed that way).
- Throughput is then simply the number of pixels generated per second times the cost of processing one pixel.

Guaranteed detection is enforced by substituting for the IFOV from the minimum acuity rule developed earlier:

$$\text{IFOV} = \frac{1}{2} \frac{Lh}{R^2}$$

Guaranteed response is enforced by substituting for the maximum range based on the expression derived in an earlier section for the stopping distance in terms of the braking coefficient:

$$R = s_B = \tau_B V [1 + b]$$

Complexity is estimated by noting that the braking coefficient does not approach 1 for the speed regimes of current research, so it can be neglected. Under this kinematic braking regime assumption, the stopping distance is the product of speed and reaction time. This is a characteristic vehicle distance - the distance required to stop. All complexity results will be polynomial in this distance.

The resulting complexity estimate represents the minimum computational throughput necessary in order to meet guaranteed response, throughput, and detection simultaneously. It will be called the **throughput performance limit**. Any system which cannot supply this throughput must either:

- reduce resolution and violate guaranteed detection
- reduce field of view and violate guaranteed throughput
- reduce lookahead and violate guaranteed response

2.7 Complexity of Constant Flux Range Image Processing

It turns out that, because a real sensor usually has a fixed field of view and fixed frame rate, it is possible to compute, not the underlying requirement, but the complexity of processing all of the data that the sensor generates if it is designed to properly resolve obstacles. This is the historical view of the problem - that of processing images at some arbitrary rate in order to resolve obstacles. The whole image is evaluated and geometry and perhaps an obstacle map is later passed to a planner which determines an appropriate response.

The basic throughput expression under guaranteed detection is.:

$$f_{cpu} = \frac{f_{pixels}}{\eta_s} = \frac{1}{\eta_s} \left(\frac{4R^4}{(Lh)^2} \right) \Psi$$

Substituting the stopping distance gives:

$$f_{cpu} = \frac{f_{pixels}}{\eta_s} = \frac{1}{\eta_s} \left[\frac{4 \left(T_{react} V [1 + b] \right)^4}{(Lh)^2} \right] \Psi$$

In the kinematic braking regime, the following result for the computational complexity is obtained:

$$f_{cpu} \sim \frac{1}{\eta_s} O \left([T_{react} V]^4 \right)$$

The following graph indicates the variation of throughput with speed when square pixel size is chosen to satisfy the maximum acuity resolution requirement at the maximum range. This paints a bleak picture. However, later sections will show that this graph arises from a design decision and is not intrinsic to the problem of autonomous mobility.

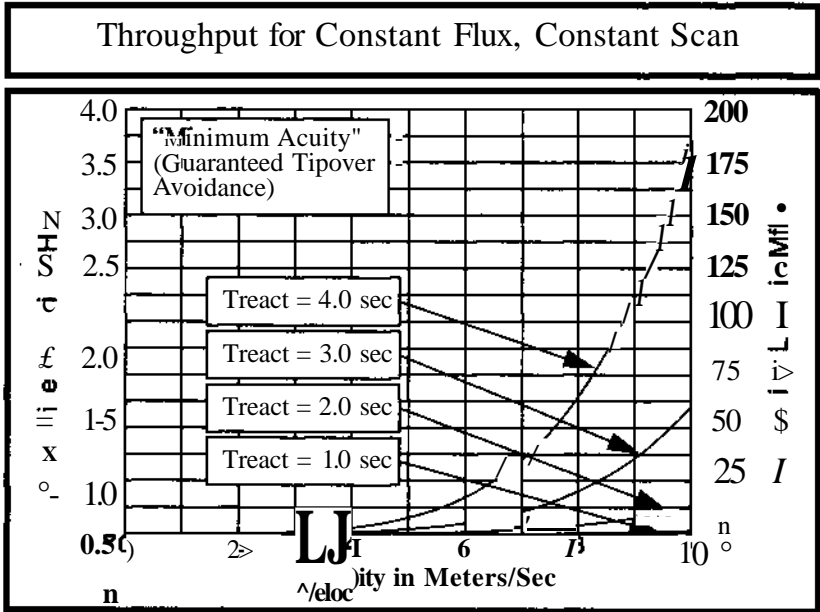


Figure 54 - Throughput for Constant Flux, Constant Scan

2.8 Complexity of Adaptive Sweep Range Image Processing

By analogy to the last section, it is possible to compute the processing requirements associated with guaranteed response for a constant flux sensor if only the requirement for guaranteed throughput is met. The basic idea is similar to the previous section except that the sensor vertical field of view is chosen so that it barely meets the guaranteed throughput requirement. This does not compromise guaranteed response and it leads to significant improvement.

The basic throughput expression under guaranteed detection is:

$$f_{cpu} = \frac{f_{pixels}}{\eta_s} = \frac{1}{\eta_s} \left(\frac{4R^4}{(Lh)^2} \right) \Psi$$

The sensor flux is, again, the solid angle measured per unit time. Thus:

$$\Psi = VFOV \times HFOV \times f_{images}$$

where the product of the vertical field of view and the frame rate is a measure of the angular velocity of the beam, and is known as the **sweep rate**:

$$\dot{\theta} = vFov \times f_{images}$$

An expression which relates the vertical field of view to its projection on the groundplane was given earlier as:

$$VFOV = \theta_{max} - \theta_{min} = \frac{h}{R_{min}} - \frac{h}{R_{max}} = \frac{h}{R_{max}l, R_{min}} \left[\frac{R_{max}}{R_{min}} - 1 \right] = h \frac{(AR)}{R_{max}R_{min}}$$

Recall that the throughput ratio is given by:

$$\frac{VT}{K_c y^c AR}$$

Using this, eliminate all ranges but the maximum from the expression for VFOV, giving:

$$VFOV = \frac{f^c - \lambda^c}{R_{max}} \left[\frac{YT_c y^c / (R_{max} p_{cyc})}{(1 - v_{eye}^{max^c})} \right]$$

Guaranteed throughput is implemented by setting $p_{cyc} = 1$. Recall the definition of the **lookahead ratio**:

$$\rho_R = \frac{VT_{cyc}}{R_{max}}$$

This gives:

$$VFOV = (h/R_{\max}) \left(\frac{\rho_R/\rho_{\text{cyc}}}{\rho_R/\rho_{\text{cyc}} - 1} \right) = (h/R_{\max}) \left(\frac{\rho_R}{\rho_R - 1} \right)$$

which is an angle considerably smaller than that used in the previous graph. The complexity expression is now:

$$f_{\text{cpu}} = \frac{f_{\text{pixels}}}{\eta_S} = \frac{1}{\eta_S} \left(\frac{4R^4}{(Lh)^2} \right) (h/R_{\max}) \left(\frac{\rho_R}{\rho_R - 1} \right) (HFOV \times f_{\text{images}})$$

using the fact that:

$$\frac{1}{x-1} \sim -1 - x - x^2 - \dots$$

there results:

$$f_{\text{cpu}} = \frac{\text{pixels}}{\eta_S} \sim \frac{1}{\eta_S} \left(\frac{4R^4}{(Lh)^2} \right) \left(\frac{h}{R_{\max}} \right) \left(\frac{V}{R_{\max}} \right) (HFOV \times f_{\text{images}})$$

Assuming every image is processed:

$$T_{\text{eye}} = 1/f_{\text{images}}$$

so this becomes:

$$f_{\text{cpu}} = \frac{f_{\text{pixels}}}{\eta_S} \sim \frac{1}{\eta_S} \left(\frac{4R^4}{(Lh)^2} \right) \left(\frac{h}{R_{\max}} \right) \left(\frac{V}{R_{\max}} \right) (HFOV)$$

Substituting the stopping distance and noting that $R = R_{\max}$ gives:

$$f_{\text{cpu}} = \frac{1}{\eta_S} \left[\frac{4 \left(T_{\text{react}} V \sqrt{1 + b_J^2} \right)^2}{L^2 h^2} \right] (V) (HFOV)$$

In the kinematic braking regime, the following result for the computational complexity is obtained:

$$f_{\text{cpu}} \sim \frac{1}{\eta_S} O \left([T_{\text{react}} V]^2 [V] \right)$$

which is less than the previous result by the factor $T_{\text{react}}^2 V$ or roughly two orders of magnitude.

This result leads to the conclusion that, in general, if the vertical field of view were adjusted for speed, computer requirements would be reduced significantly. This idea will be called **adaptive sweep** because it arises fundamentally from the direct relationship between throughput and the sensor sweep rate. An alternate name is **computational stabilization** which borrows terminology from strapdown inertial navigation.

The following graph indicates the variation of throughput with speed when the vertical field of view is computed from the above expressions and square pixel size is chosen to satisfy the resolution requirement at the maximum range.

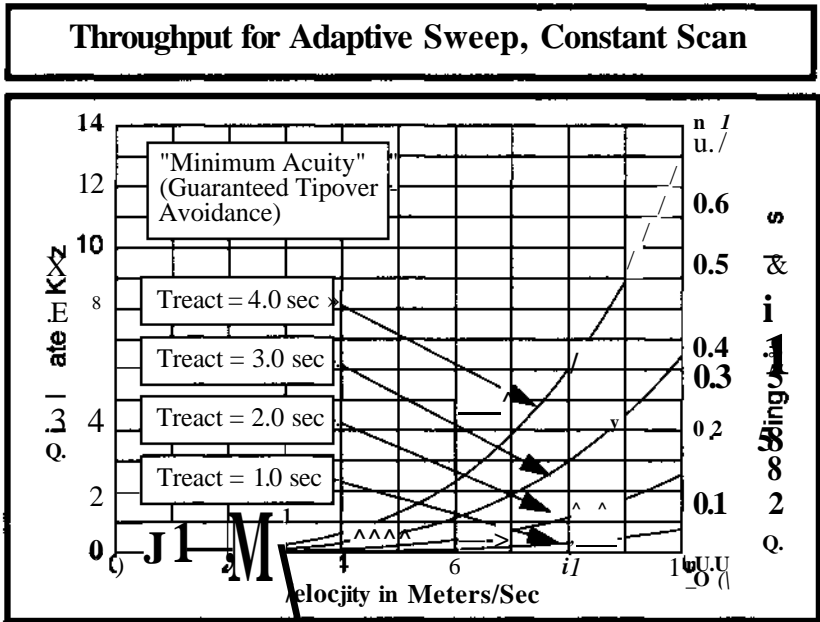


Figure 55 - Throughput for Adaptive Sweep, Constant Scan

This graph demonstrates that the problems of the previous graph were an illusion. In fact, if the vertical field of view is chosen such that it satisfies the guaranteed throughput requirement, the throughput requirements are 300 times lower than constant flux at 10 meters/second speed.

Thus, while the logic of decreasing pixel size for higher speeds is inescapable, equivalent logic leads to a reduced vertical field of view requirement. Adaptive sweep is the key to keeping the navigation problem tractable because it is worth two orders of magnitude improvement in efficiency.

The key to this complexity difference is the redundant measurement of the same geometry that happens when images overlap on the groundplane. The average number of images that fall on a given patch of terrain is called the **imaging density** c_i .

2.9 Complexity of Adaptive Sweep/Scan Range Image Processing

The complexity analysis of this section gives the best complexity that can be practically achieved using a conventional fixed field of view sensor where the pixel shape is optimized in order to match the groundplane resolution requirement.

Although adaptive sweep is a significant improvement in complexity, it addresses only the optimum size of the image. It does not address the fact that the density of pixels on the groundplane and the aspect ratio of a pixel are grossly suboptimal. If the sensor pixel size and shape were optimal, then the throughput would be exactly equal to the theoretical minimum possible throughput and another two orders of magnitude improvement would be possible. While there are limits to what can be done, the idea of dynamically adjusting pixel size and shape, called **adaptive scan**, is an important one.

It is difficult to construct a system with perfect density on the ground. However, it is easy to subsample an image in azimuth to remove the effects of poor aspect ratio. Further, guaranteed response already requires that high depression scanlines (where the density is too high) be avoided. These two approaches can eliminate the effects of poor geometric efficiency. Recall that the cpu load required to process all sensory data is given by:

$$f_{\text{cpu}} \sim f_{\text{pixels}} = \frac{\Psi}{(\text{IFOV})^2}$$

In practical adaptive scan, the pixel aspect ratio is adjusted to be a constant over the field of view and equal to the perception ratio, and this gives the opportunity to cancel one of the ranges. The vertical and horizontal instantaneous field of view now have different expressions at minimum acuity:

$$\text{IFOV}_V = \frac{1Lh}{2R^2} \quad \text{IFOV}_H = \frac{1Lh\left(\frac{R}{h}\right)}{2R^2} = \frac{1L}{2R}$$

thus, the throughput required to process all sensory data is given by:

$$f_{\text{cpu}} = \frac{f_{\text{pixels}}}{\eta_S} = \frac{1}{\eta_S} \left(\frac{4R^3}{L^2h} \right) \Psi$$

As before, the vertical field of view necessary for guaranteed throughput is:

$$\text{VFOV} = \left(\frac{h}{R_{\text{max}}} \right) \left(\frac{1}{\rho_R} \right) \left(\frac{1}{\rho_{\text{cyc}} - 1} \right) = \left(\frac{h}{R_{\text{max}}} \right) \left(\frac{\rho_R}{\rho_R - 1} \right)$$

The complexity expression is now:

$$f_{\text{cpu}} = \frac{f_{\text{pixels}}}{\eta_S} = \frac{1}{\eta_S} \left(\frac{4R^3}{L^2h} \right) \left(\frac{h}{R_{\text{max}}} \right) \left(\frac{\rho_R}{\rho_R - 1} \right) (\text{HFOV} \times f_{\text{images}})$$

which is, assuming every image is processed:

$$f_{cpu} = \frac{f_{pixels}}{\eta_s} \sim \frac{1}{\eta_s} \left(\frac{4R^3}{L^2 h} \right) \left(\frac{h}{R_{max}} \right) \left(\frac{V}{R_{max}} \right) (HFOV)$$

Substituting the stopping distance gives:

$$f_{cpu} = \frac{1}{\eta_s} \left[\frac{4 \left(T_{react} V [1 + b] \right)}{L^2} \right] (V) (HFOV)$$

In the kinematic braking regime, the following result for the computational complexity is obtained:

$$f_{cpu} \sim \frac{1}{\eta_s} O \left(T_{react} V^2 \right)$$

which is less than the previous result by the factor $T_{react} V$ or roughly one order of magnitude. The following graph indicates the variation of throughput with speed when vertical field of view is computed from the above expressions, nonsquare pixel size is chosen to satisfy the resolution requirement at the maximum range, and system cycle time is set to the frame rate.

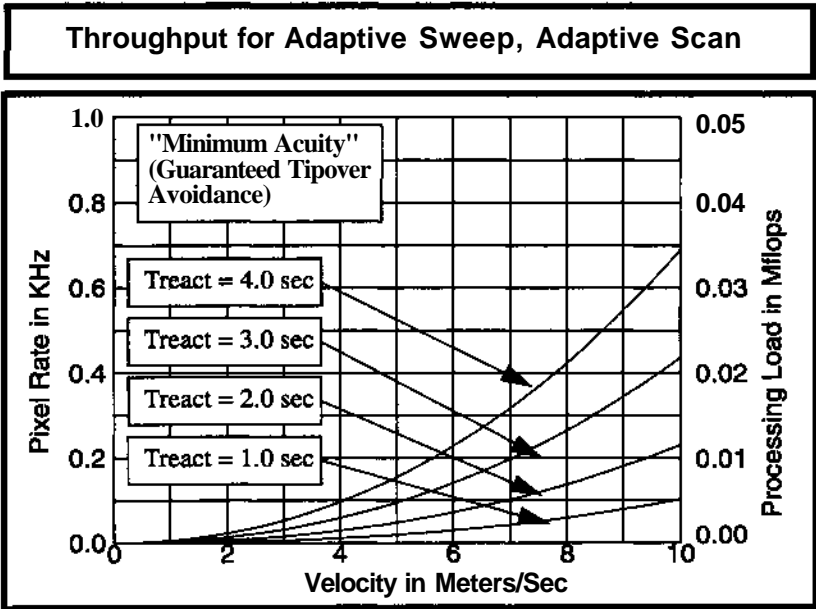


Figure 56 - Throughput for Adaptive Sweep, Adaptive Scan

It is clear from this graph that a problem which at first appeared impossible, requiring supercomputers, is in fact *theoretically trivial*. In fact, by using simple adaptive techniques, the throughput requirements are 20 times lower at 10 meters/second speed than adaptive sweep and *5,700 times lower than constant flux at 20 mph*.

2.10 Complexity of Adaptive Sweep, Uniform Scan Image Processing

The following analysis considers the fundamental acuity and throughput requirements of perception in terms of a terrain map but it does not necessarily imply that a map is explicitly formed. As a minimum requirement, any sensor must generate map cells at a rate that is consistent with the rate at which the vehicle consumes them through motion. For now, consider that the motion of the vehicle consumes a swath of map cells directly in front of it as shown below:

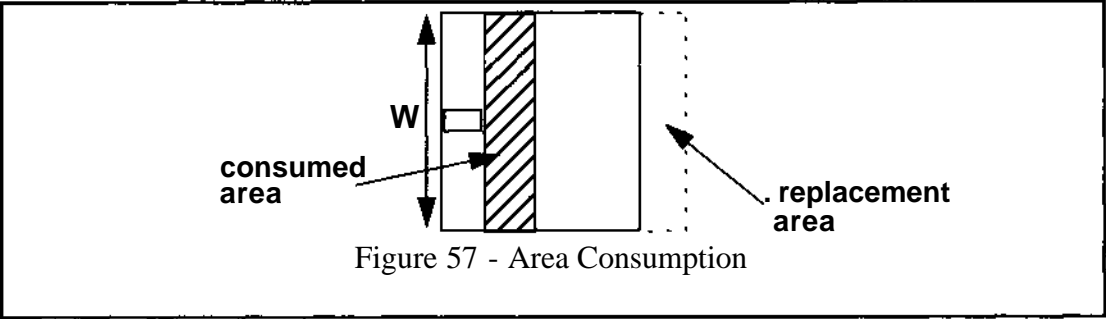


Figure 57 - Area Consumption

In the simplest case, this consumed area must be replaced by adding new information to the map shown to the right. In general, the new information need not be at the end of the map, but could be anywhere where an unknown cell exists. This is a pure throughput argument. In fact, it has already been argued that guaranteed response places limits on how close a measured pixel can be before it becomes useless.

Also, although the information immediately to the left and right of the vehicle is not useful, the system did not know this at the time it was measured. Further, guaranteed response requires that the vehicle measure geometry a long way off, so, for now, there appears to be no solution to this problem of measurement of useless lateral geometry³⁹.

The area consumed per second, expressed in map pixels, is the required absolute minimum throughput of a perception system. This quantity is the minimum rate at which new geometric information must be generated, regardless of the scanning pattern of any sensor, or the vehicle must either

- drive over unknown terrain and violate guaranteed throughput
- accept inadequate resolution information and violate guaranteed detection

Let the width of the map be W , the velocity of the vehicle be V , and the map resolution (unit length per cell) be δ . This minimum rate is given by:

$$f_{\text{cells}} = \frac{WV}{\delta^2}$$

In previous sections it was shown that, under guaranteed response, the maximum range can be determined from the stopping distance. Let L be the vehicle wheelbase. Setting the width of the map to twice the maximum range gives:

$$f_{\text{cells}} = \frac{2s_B V}{L^2} = \frac{2T_{\text{react}} V^2 [1 + b]}{L^2}$$

39. Something, at least, can be done, if the sensor field of view can be pointed.

Putting all of these results together, gives the following expression for the processing load:

$$f_{cpu} = \frac{1}{\eta_s} f_{cells} = \frac{i}{\eta_s} \frac{2 T_{react} v^2 [1 + b]}{L^2}$$

In the kinematic braking regime, the following result for the computational complexity is obtained:

$$f_{cpu} \sim \frac{1}{\eta_s} O(T_{react} V^2)$$

which is, in complexity terms, *equal to the adaptive sweep, adaptive scan expression*. This is the fundamental complexity of the problem⁴⁰. There is a multiplicative constant difference of 2 x HFOV between this minimum requirement and the adaptive sweep, adaptive scan expression because the whole image is processed at the same nonsquare pixel resolution in adaptive scan.

This relationship is plotted below for minimum acuity map resolution of 3.3 meters versus vehicle velocity for various values of system reaction time. Again, for consistency reasons, 50 flops per cell are assumed. The analysis does not assume a uniform scan on the ground. It only assumes that the average density of the scanning pattern is one. This average is accumulated up to the time the vehicle drives past a particular point in the map.

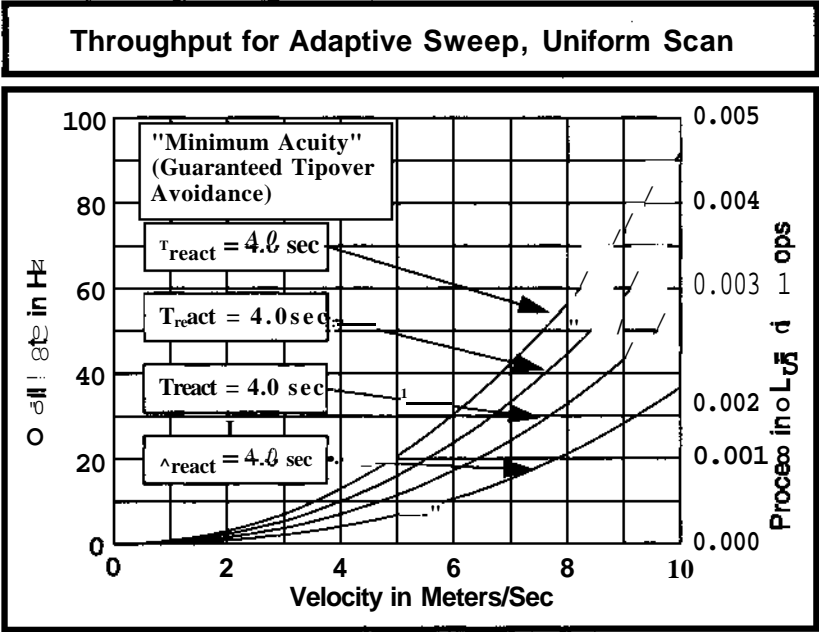


Figure 58 - Throughput for Adaptive Sweep, Uniform Scan

40. This is also why autonomy based on line scanners is feasible and so very efficient. Effectively, adaptive sweep and adaptive regard convert an imaging sensor into an ideal computationally-stabilized line scanner.

2.11 The Fundamental Speed/Resolution Trade-off

Recall that the complexity estimates are all consistently based on a kinematic braking regime assumption. The true power of velocity is actually squared as speeds increase. Identical resolution assumptions have led to the following throughput estimates for different image processing algorithms:

Table 2: Throughput Estimates

Algorithm	Estimate at Minimum Acuity, 4 second Reaction Time, and 10 m/s speed	Complexity
constant flux	250 Mflops	$O(T_{\text{react}}^4 V^4)$
adaptive sweep	0.7 Mflops	$O(T_{\text{react}}^2 V^3)$
adaptive sweep, scan	0.035 Mflops	$O(T_{\text{react}} V^2)$
ideal	0.0045 Mflops	$O(T_{\text{react}} V^2)$

The actual data for all 4 second reaction time curves is plotted below on a logarithmic vertical scale.

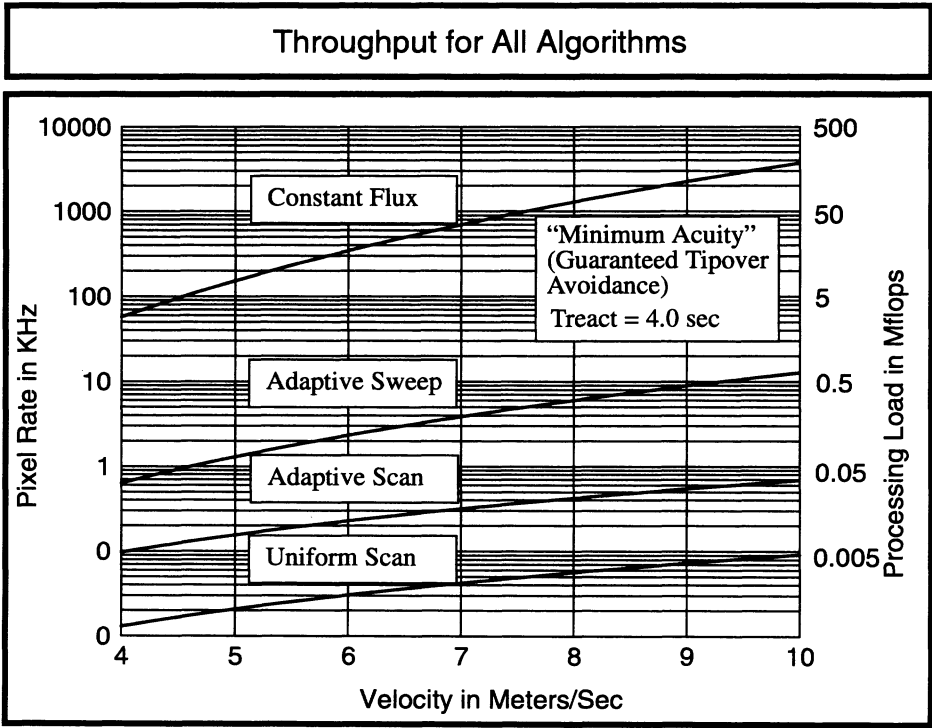


Figure 59 - Throughput for All Algorithms

The logic of decreasing pixel size for higher speeds is inescapable, but equivalent logic leads to a reduced vertical field of view requirement, so if the vertical field of view is not reduced, extreme throughput waste is being tolerated. Further, because pixel aspect ratio is extremely elongated at high ranges, the density of measurements in the crossrange direction is grossly suboptimal unless it is managed.

Notice that the complexity in either of the above cases contains a constant times a power of the product $T_{\text{react}} V$. That is:

$$f_{\text{cpu}} \sim \frac{1}{\eta_s} O\left([T_{\text{react}} V]^N [V]^M \right)$$

This will be called the **fundamental trade-off** because it indicates that the trade-off of finite computing resources is one of reliability for speed. This is a basic trade-off of speed and resolution which always arises from a system throughput limit. Computing resources establish a limit on vehicle performance which can be expressed as either high speed and low reliability or vice versa.

There are a few ways to read the result. If throughput is fixed, then speed is inversely proportional to reaction time. If speed is fixed, throughput required is the n th power of reaction time. If reaction time is fixed, throughput is the $(n+m)$ th power of speed.

Section 3: Rationale for A Real-Time Approach

The previous sections have shown that the throughput required of a system is proportional to high powers of both speed and reaction time by substituting the dependence of maximum range on speed, and the dependence of pixel size on range into the throughput expression. This was called the **throughput problem**. The resulting throughput depended on both reaction time and speed.

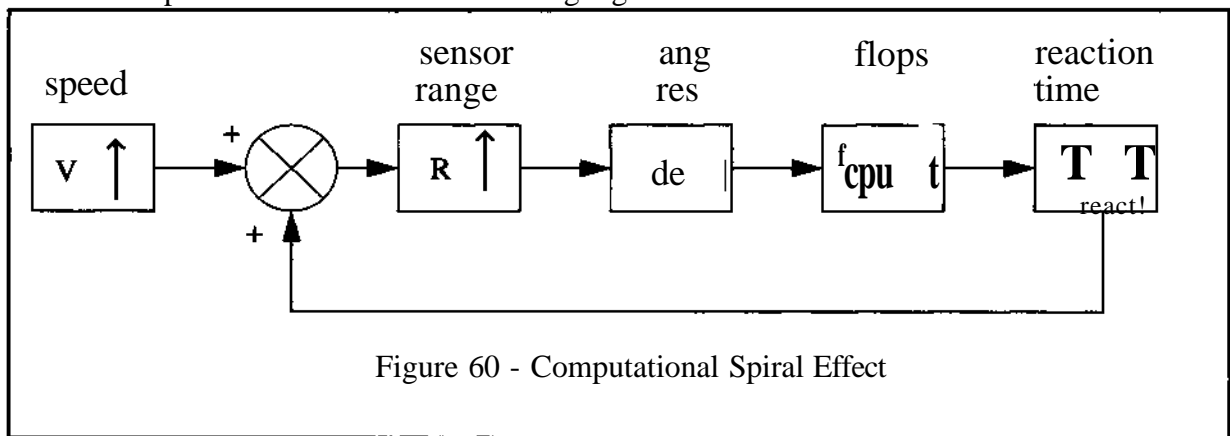
3.1 Computational Spiral Effect

There is another dependence which has not been resolved. Reaction time itself depends on throughput. It increases as throughput increases for a fixed computer system. The intuitive logic behind this is:

- Driving faster, or reacting slower, requires looking farther ahead.
- Looking farther ahead requires increased sensor angular resolution in order to resolve hazards at the lookahead distance.
- Increased resolution generates higher processing loads.
- Higher processing loads cause longer reaction times.
- Longer reaction times require looking farther ahead.

Therefore increased reaction time causes increased reaction time because computer cycle time is a component of reaction time. That is, as speeds increase, the computer has to do even more work per cycle than what would be expected based on the previous analysis. Effectively, *reaction time itself is a function of throughput* and therefore it is a function of velocity ⁴¹. This will be called the **computational spiral effect**.

The spiral effect arises fundamentally from the policy of guaranteed safety because it closes a relational loop. It is indicated in the following figure:



41. Note however, for fixed sensor angular resolution, throughput decreases as speeds increase because the region of interest subtends a smaller angle in the image. If resolution is allowed to increase with speed, however, pixels decrease in size at the same rate as the angular width of the region of interest. As a theoretical analysis, the latter case is considered here.

3.2 Impact of the Spiral Effect

This spiral effect gives rise to a trade-off between throughput and reaction time. Conceptually, throughput and reaction time are completely independent. However, in practice, any control system will cycle at some finite rate and this causes the two to be related. Consider the following simplification of the throughput expression:

$$f_{\text{cpu}} = \frac{K_1}{\eta_s} [T_{\text{react}} V]^N [V]^M$$

for some constant K_1 . Thus, the change in required minimum throughput due to a change in reaction time is given by:

$$\Delta f_{\text{cpu}} = \frac{K_1}{\eta_s} N T_{\text{react}}^{N-1} V^{N+M} \Delta T_{\text{react}} = \frac{N f_{\text{cpu}}}{T_{\text{react}}} \Delta T_{\text{react}}$$

A simple model of the dependence of reaction time on throughput is as follows:

$$T_{\text{react}} = T_{\text{fixed}} + T_{\text{variable}} = T_{\text{fixed}} + K_2 f_{\text{cpu}}$$

where the fixed component is due to hardware and the variable component is due to software. It has been assumed here that software cycle time will increase proportionally to the amount of data processed. Thus, the change in reaction time due to a change in throughput is:

$$\Delta T_{\text{react}} = K_2 \Delta f_{\text{cpu}} = \frac{(T_{\text{react}} - T_{\text{fixed}})}{f_{\text{cpu}}} \Delta f_{\text{cpu}}$$

If this is substituted back in to the differential throughput expression, a difference equation results:

$$[\Delta f_{\text{cpu}}]_{i+1} = \frac{N f_{\text{cpu}}}{T_{\text{react}}} \Delta T_{\text{react}} = N \left(1 - \frac{T_{\text{fixed}}}{T_{\text{react}}} \right) [\Delta f_{\text{cpu}}]_i$$

The **reactive ratio** was defined earlier as the ratio of the software component of reaction time to the total reaction time, thus:

$$[\Delta f_{\text{cpu}}]_{i+1} = \frac{N f_{\text{cpu}}}{T_{\text{react}}} \Delta T_{\text{react}} = N \rho_{\text{react}} [\Delta f_{\text{cpu}}]_i$$

In order to understand this expression, it is illustrative to assume $\rho_{\text{react}} \sim 1$. This amounts to an assumption that most of the reaction time is due to software. Under this assumption, it has been shown that if a system processes slightly more data than absolutely necessary, the time it takes to do this actually increases the minimum throughput required by N times the amount added. Now in practical cases, the exponent, N , of reaction time in the throughput expression is greater than 1.

Thus, when a system can barely supply the required throughput, it is a losing battle to try to improve system reliability with more sophisticated processing.

The analysis suggests that optimal real-time performance is achieved by reducing the field of view and increasing the frame rate in order to improve reaction time, but to ensure that guaranteed throughput is still maintained. In theoretical terms, *the trade-off is managed by reducing throughput to a bare minimum and reducing reaction time as far as possible. This is the real time approach*² to the problem.*

There are at least two other reasons to reduce throughput to a minimum. The first is that it permits the highest possible vehicle speed. This is clear from the **throughput ratio**.

The second is that, because range data uncertainty increases quadratically with range, it is optimal to reduce the lookahead as far as possible without violating **guaranteed response**. In this case, the quality of the range data is as high as possible.

42. Guaranteed response implies that a real-time approach to the problem of high-speed mobile autonomy is needed simply because there exists stringent response requirements which must be met. Here, a real-time approach means simply that response is a consideration - it does not necessarily imply the need for a real-time operating system.

Chapter 7: Summary and Conclusions

This section summarizes all of the most important content of Part II: Analysis.

Section 1: Elements

The basic requirement of intelligent mobility is to maintain vehicle safety. This requirement can in turn be expressed in terms of timing, speed, resolution, and accuracy requirements on sensors, effectors, and computer models and algorithms.

Chapter 1 has expressed these four elements of the problem in nondimensional terms and identified four nondimensional variables which relate vehicle parameters such as speed, response time, wheelbase, and wheel radius, to sensor parameters such as lookahead and vertical field of view, and to software system parameters such as processing cycle time, and environmental model spatial resolution. The essence of the safety requirement is that the four requirements mentioned above must be met at all times and this gives rise to nondimensional relationships between the above variables which, it often turns out, must be managed actively as the system moves.

1.1 Summary of Elementary Nondimensionals

The following table summarizes these elementary nondimensionals:

Table 3: Elementary Nondimensionals

Symbol	Name	Expression	Symbol	Name	Expression
\hat{r}_{react}	Response Ratio	$\frac{VT}{r_{L}^{react}}$	\hat{r}_{eye}	Throughput Ratio	$\frac{v_{cyc}^T}{AR}$
P_{dy}	Minimum Acuity Ratio	$\frac{dy}{L}$	P_{dz}	Maximum Acuity Ratio	$\frac{dz}{r}$
P_{dx}	Minimum Fidelity Ratio	$\frac{dx}{(A-W)}$	P_{dx}	Maximum Fidelity Ratio	$\frac{dx}{\delta}$

1.2 Derivative Issues

Using these nondimensionals as a guideline, one can first of all establish natural thresholds on their values which correspond to safety thresholds and then characterize all of the major failure modes of autonomous navigators by progressively assigning blame to each element of each expression.

The **response problem** is that of responding fast enough. If a response failure occurs at some target speed, it is either because the lookahead was too small - a **myopia problem**, or the response time was too large - a **latency problem**.

The **throughput problem** is that of sensing fast enough. If a throughput failure occurs at some target speed, it is either because the sensor or computer was not fast enough, the sensor was not pointed appropriately - a **stabilization problem**, or the field of view was not large enough - a **tunnel vision problem**.

The **acuity problem** is that of representing the vehicle and its environment at sufficient resolution to allow the system to resolve important features in space and events in time. There is some resolution that is barely able to resolve the largest important feature - **minimum acuity** and some resolution that is barely able to resolve the smallest important feature - **maximum acuity**. System acuity failures can be related to inadequate spatial resolution - called the **sampling problem** and inadequate temporal resolution - called the **motion distortion problem**.

The **fidelity problem** is that of representing the vehicle and its environment sufficiently accurately to enable correct decisions. Models and associated measurements may be overly sensitive - a **sensitivity problem**, not repeatable as in the **image registration problem**, or not observable - an **observability problem**. An attendant issue of model fidelity is that inaccurate models can cause control instability - a **stability problem**.

1.3 Derivative Adaptive Rules

Using these nondimensionals as a guideline, it is possible to elaborate mechanisms that adapt system parameters that can be changed to those that cannot be changed in order to ensure safety. For example, when a response failure is imminent, one can reduce speed, increase lookahead, or reduce reaction time independently or in any combination in order to improve matters.

The top level system requirement was expressed as safety. This was converted into response, throughput, acuity and fidelity in this section. The remaining chapters of this part of the thesis elaborates these four requirements into the following fourteen aspects of performance before presenting the design of the navigator:

- vehicle configuration
- reaction time
- braking and turning maneuverability
- sensory lookahead
- depth of field
- field of view
- sweep rate
- perceptual bandwidth
- angular resolution
- positioning bandwidth
- sensor geometric efficiency
- dynamic modelling fidelity
- positioning fidelity
- perceptual fidelity

Section 2: Configuration

Many aspects of performance are intrinsically related to the dimensions of the vehicle - sensor system. For this reason, a set of nondimensionals are developed in order to reduce the dimensionality of the expression of vehicle shape.

2.1 Summary of Configuration Nondimensionals

The following table summarizes the vehicle configuration nondimensionals:

Table 4: Configuration Nondimensionals

Symbol	Name	Expression	Symbol	Name	Expression
L	Normalized Wheelbase	$\frac{L}{R}$			
E	Perception Ratio	$\frac{h}{R}$	^A L	Longitudinal Aspect Ratio	$\frac{h}{L}$
f	Normalized Wheel Radius	$\frac{r}{R}$	^A r	Wheel Fraction	$\frac{r}{L}$
c̄	Normalized Undercarriage Clearance	$\frac{c}{R}$	^A c	Undercarriage Tangent	$\frac{c}{L}$

of these variables, the **perception ratio** and the **undercarriage tangent** will be most important later.

Section 3: Response

The elements of the **response ratio** are reaction time, lookahead and speed. Taking speed as fixed, this section investigated the other two elements of the requirement.

The reaction time of an autonomous system includes many components including those of electronic, software, and mechanical systems and the total system response time is the sum of all of these. The total **system reaction time** is a big number and for moderate vehicle speeds, the vehicle travels a fair distance before the commands it generates are enacted in hardware. Rather than consider the actual numbers in detail, a nondimensional approach to the analysis has been taken.

3.1 Summary of the Response Nondimensionals

All of the nondimensionals given in this section are related to the response of the vehicle. They are summarized below:

Table 5: Response Nondimensionals

Symbol	Name	Expression	Symbol	Name	Expression
μ	Coefficient of Friction	numeric	ν	Coefficient of Lateral Acceleration	numeric
\bar{b}	Kinematic Braking Coefficient	$\frac{VT}{s_{react}}$	\bar{t}_k, \bar{t}	(Kinematic) Turning Coefficient	$\frac{P}{VT_{react}}$
$\bar{b}_{d,6}$	(Dynamic) Braking Coefficient	$\frac{V}{2 \wedge T_{react}}$	\bar{t}_d	Dynamic Turning Coefficient	$\frac{v^2}{vgp}$
AR_L	Normalized Incremental Lookahead Distance	$\frac{AR_L}{VT_{react}}$	$A_{\gamma L}$	Incremental Lookahead Angle	$\frac{AR_L}{P}$
R	Range Ratio	$\frac{R_{max}}{R_{min}}$	PR	Lookahead Ratio	$\frac{VT_{eye}}{R_{max}}$
\bar{r}_{react}	Reactive Ratio	$\frac{T_{sw}}{T_{react}}$			

3.2 Maneuverability and Response Time

It is computational reaction time and mechanical maneuverability together which determine the ability of the system to respond. In order to render the problem in a form suitable for analysis, a few simplifying assumptions are made. First, because throttle-generated speed cannot normally be changed much over the planning horizon, the obstacle avoidance maneuvers considered concentrate on the other two available actuators - on braking and turning.

Second, four classical maneuvers are distinguished which employ combinations of these actuators. These are the **panic stop**, the **turning stop**, the **impulse turn**, and the **reverse turn**.

3.3 Braking

In the case of braking, the **braking coefficient** represents how much of the distance in front of the vehicle is consumed in braking. It turns out that most of the lookahead distance is consumed in simply deciding to brake, so that stopping distance is mostly linear in initial speed.

This is good news in the sense that software is empowered to make a difference and bad news in the sense that it now has to do the job. This point is expressed more concisely with the **reactive ratio**. As a measure of software to total reaction time, it represents the relevance of software update rate to vehicle safety.

A major conclusion of this section is that because the reactive ratio approaches unity in the mobility application, it classifies the problem as one for which software update is indeed very important and therefore as a **real-time**⁴³ problem.

3.4 Turning

Turning requires much more space than braking in the sense that it takes more space to change direction than it does to stop. This is unfortunate because continuously moving navigators must be more sophisticated in order to stay in motion. The major conclusion of this section is that high amplitude turns such as the reverse turn cannot be modelled by assuming instantaneous response of the steering actuator as has been done classically at low speed. Indeed, this assumption is wrong at even moderate speeds and its invalidity gives rise instability of both obstacle avoidance and goal seeking behaviors.

3.5 Lookahead

An analysis of lookahead starts with the basic requirement of looking where you are going but an even cursory look at the matter indicates a few important subtleties. Given that response times are large, it seems more relevant to look, not where you are going, but where you will be when the commands you are now considering will reach the hardware. This observation and a few related ones are grouped in this section into a concept called **adaptive regard**.

In more plain terms, the issue under consideration is that the product of speed and reaction time, the **reaction distance**, is a large number, and this number represents the distance that the vehicle is already committed to travelling in any planning cycle. If the vehicle is already committed to this motion, then there is nothing useful that a planner can do about it and hence it is wasteful to think about the inevitable when in a survival situation.

43. By definition, a real-time problem is one for which response time is important. This classification has *nothing to do with* whether a real-time operating system is used.

The importance of adaptive regard is manifold. First, it explicitly computes where the vehicle will go when a command is issued - which later sections will show is not at all a trivial matter. Second, it generates a characterization of where the vehicle is already committed to going - the **dead zone**, where it cannot go - the **free zone**, and where it really has an option of going - the **detection zone**. Third, it provides the region of interest in the environmental model for planning purposes which can be used by the perception system to compute a corresponding region of interest in the image.

Section 4: Throughput

The elements of the **throughput ratio** in addition to speed are the environmental sensory throughput expressed in terms of field of view and image processing frame rate⁴⁴. Field of view is investigated here in terms of a depth of field - a minimum and maximum range - and then horizontal and vertical field of view.

4.1 Summary of Throughput Nondimensionals

The following table summarizes the throughput nondimensionals:

Table 6: Throughput Nondimensionals

Symbol	Name	Expression	Symbol	Name	Expression
HFOV	Horizontal Field of View	numeric	VFOV	Vertical Field of View	numeric
σ_I	Imaging Density	numeric	σ_S	Scanning Density	numeric

4.2 Depth of Field

From **adaptive regard** we know that range data inside the stopping distance is almost useless and the measurement of range data outside the distance required to turn can be postponed until it is really needed. Using this specification of the depth of field of a sensor leads to a system which first tries to turn to avoid obstacles but which stops if turning is unsuccessful, so it leads to two levels of defense against hazards.

When maximum range is quantified at high speed, it becomes clear that contemporary sensors have a **myopia problem** because there are real physical limits on how much response time can be reduced.

4.3 Horizontal Field of View

The horizontal field of view is quantified in terms of vehicle response and speed and the conclusion that results is that most contemporary sensors have a severe **tunnel vision problem**. Indeed, a system engaged in a sharp turn is often obliged to drive over unknown terrain and until the horizontal field of view is increased significantly, the only available alternative is to artificially limit the turning maneuverability of the vehicle.

4.4 Vertical Field of View

Requirements on the vertical field of view can be generated by several means. If a kinematic argument is used and the sensor is not stabilized, then the requirement is for a very wide field of view on rough terrain. Unfortunately, there exists a strong interrelation between vertical field of view, sensor throughput and the **stabilization problem**. That is, one can easily solve either the

44. This is the rate they are processed, not the rate generated in hardware. Few if any systems can keep up with a 30 Hz video camera.

throughput problem or the stabilization problem but apparently not both. In plain terms, one can either look everywhere - and process too much, or process as little as possible - and risk missing something.

The difficulty of satisfaction of the kinematic requirements on vertical field of view lead to an expression of the dynamic requirements in terms of a quantity called the **sweep rate**. This quantity removes the dependence on field of view and frame rate by basically working with their product.

Under the assumption that mechanical or software stabilization can be achieved, the sweep rate specification of throughput becomes the minimum throughput requirement. A mechanism called **adaptive sweep** is proposed which achieves this required stabilization and the attendant improvement in throughput. This notion is one of the most important elements of the navigator design.

Section 5: Acuity

The **acuity ratios** relate the configuration of the vehicle to the resolution required of the software world model. While these ratios contain no interesting subcomponents, it is interesting to analyze how world model resolution maps onto sensor resolution and what parameters affect this transformation. It is vehicle resolution and sensor configuration together which determine the ability of the system to resolve obstacles.

5.1 Summary of the Acuity Nondimensionals

The identified pi products for acuity are summarized below:

Table 7: Acuity Nondimensionals

Symbol	Name	Expression	Symbol	Name	Expression
$\mathbf{\overline{f_i}}$	Perception Ratio	$\frac{h}{R}$	\mathbf{R}	Range Ratio	$\frac{R_{\mathbf{max}}}{R_{\mathbf{min}}}$
σ_I	Imaging Density	numeric	Scanning Density	σ_S	numeric
\overline{AR}	Normalized Range Difference	$\frac{AR}{R_{\mathbf{min}}}$	$\nabla\delta$	Disparity Gradient	
$\mathbf{\overline{5}}$	Normalized Disparity	$\frac{d}{f}$	$\frac{b}{h}$	Normalized Baseline	
η_G	Geometric Efficiency	$\frac{1}{a^a S^a I}$			

5.2 Range Image Acuity

The acuity analysis generates several key nondimensionals. First, the **perception ratio** identifies a basic geometric fact of life for high speed autonomous navigators - stopping distance exceeds the height of the sensor. The assumption that this is so is called the **small incidence angle assumption** and it will be important later. The assumption has both advantages and disadvantages.

On the down side, because a sensor cannot practically be mounted over the region of interest looking down at it, the spatial resolution of the world model is neither homogeneous nor isotropic. Indeed, the size of a pixel can be expected to vary by up to three orders of magnitude (the cube of the inverse perception ratio) over the field of view and the aspect ratio can be as high as 10 (the inverse perception ratio). The up side will be discussed later in adaptive perception.

Independent of whether resolution is constant is the issue of whether it is adequate. It turns out that contemporary sensors can barely meet the **minimum acuity** requirement because a single range pixel at the maximum range projects onto an area on the groundplane that can exceed the size of the vehicle.

5.3 Range Acuity in Stereo Vision

The resolution of stereo vision range images is often poor for many reasons. In the case of range resolution, the quadratic growth of range resolution with range implies that contemporary systems cannot resolve a pitch hazard just as laser rangefinders cannot for other reasons. As a result, **subpixel disparity estimation** is called for but this is a process that tends not to be robust.

5.4 Angular Acuity in Stereo Vision

Stereo vision angular acuity is also poor due to a basic tradeoff of reliability for resolution or in more classical terms of detection for localization. Large correlation windows are more likely to generate a distinct correlation peak at the cost of putting it in the wrong place. This issue is a central limitation of stereo vision that is connected with its current inability to account for **disparity gradients** across the correlation window.

5.5 Positioning Bandwidth

A matter of acuity concern that relates to time is the bandwidth of position estimates. This section shows that removal of the distortions due to motion can necessitate position estimates that exceed 100 Hz in frequency. A related issue is the degree of simultaneity of image capture for stereo vision. This section computes that no more than a few hundred microseconds is acceptable.

5.6 Geometric Efficiency

The **geometric efficiency** is defined as the inverse of the amount by which environmental geometry is oversampled with respect to any target spatial resolution. The practical implication of the **sampling problem** is that varying spatial resolution intrinsic to the sensor mounting geometry must cause either undersampling at long range or oversampling at short range or both.

The geometric efficiency of a sensor has two elements. The efficiency of a single image is related to the **scanning density** whereas the efficiency associated with image overlap is related to the **imaging density**.

In practical terms, the efficiency of the ERIM laser rangefinder is only 2% - implying that 98% of the geometry that it generates is redundant. The throughput problem is directly related to the geometric efficiency issue. In the response section, the **adaptive sweep** mechanism was proposed to reduce image overlap. In the language of this section it impacts the quantity called imaging density. Another mechanism is also proposed here called **adaptive scan**. This mechanism will reduce scanning density and attempt to remove the effects of the nonconstant mapping of resolution from image space to the world model.

Section 6: Fidelity

The **fidelity ratios** relate the configuration of the vehicle to the accuracy required of the software models of the world and the vehicle. While these ratios contain no interesting subcomponents, it is interesting to analyze how world model and vehicle model accuracy affect planning decisions in a more quantitative manner.

6.1 Summary of the Fidelity Nondimensionals

The identified pi products for fidelity are summarized below:

Table 8: Fidelity Nondimensionals

Symbol	Name	Expression	Symbol	Name	Expression
t	Normalized Time Constant	$\frac{T}{T_{act}}$	t _t	Transient Turning Coefficient	$\frac{T}{T_{act}} \frac{V}{V_{Pmin}}$
P _t	Turning Fidelity Ratio	$\frac{T_{look}}{T_{act}} \frac{V}{(r - W)}$			

6.2 Modelling Dynamics and Delays

This section considers dynamics from the point of view of their importance to modelling fidelity. Later sections will identify the impact of poor fidelity on the problem. Dynamics are defined here to include the pure delays inherent in digital controllers, the dynamics of actuator response, and the dynamics of vehicle - environment interaction. As a basic requirement, any system which attempts to avoid obstacles must understand the environment and its own motion sufficiently well. It must be able to interpret and localize latent sensory information sufficiently well and it must understand its own ability to maneuver sufficiently well.

Simple calculations indicate that 1 second of unmodelled latency in a 3 m/s vehicle will cause 3 meters or about 10 ft. of localization error unless it is modelled and these numbers are not atypical. If localization error is to be kept within the resolution of the world model, then the **minimum significant delay** is on the order of one tenth of a second. Any delays in excess of this must be removed through some sort of model of what is happening in time.

It was shown that because a conventional automobile actuates curvature, it is basically a filter. Indeed, for a moderate speed of 5 m/s the difference between an ideal model of steering and an accurate dynamic one is 15 meters⁴⁵.

45. Note that this is the error between the predicted an actual response of the vehicle and it is not to be confused with the errors in position estimation which are a separate matter, This 15 m error exists even if the position estimation is perfect.

The simplest expression of this issue is through a quantity called the **normalized time constant**. As the ratio of actuator characteristic time to planner lookahead time, it expresses the importance of dynamic models in the planning process. Its particular form for the steering actuator is called the **transient turning coefficient** which has a typical value that implies that at least turning dynamics must be considered for robust performance.

The major conclusion of this section is that a high speed navigator operates for local planning purposes entirely within its transient regime and hence entirely within the regime for which dynamic models of some form are necessary⁴⁶.

6.3 Dynamics of Turning

This section elaborates an accurate model of the Ackerman steering mechanism - the primary problem actuator - and assesses the impact of this model on the problem. There are three principle problems associated with the Ackerman steering mechanism. The first is steering dynamics, a **latency problem**, and the second is the nonintegrability of constraints, a **nonholonomy problem**. The third is the fact that the conversion from steer angle to curvature is **nonlinear**. A further problem is that attitude rate is **coupled** to both speed and steer angle.

What all this means in effect is that while a forward model of steering is trivial to compute an inverse model is almost impossible to compute. A forward model maps steering and speed commands onto position and attitude. An inverse model maps position and attitude onto steering and speed. There are at least three different reasons why an inverse model is so hard:

- Actuator Dynamics. While there exists a computable response for all commands, there is no command that can generate a response that does not satisfy the underlying actuator differential equation.
- Nonholonomic Differential Constraints. There are two inputs (speed and curvature) and three outputs (x, y, and heading) so there clearly is no inverse mapping from an arbitrary response to an associated command.
- Coupling. The path followed for any curvature command depends on speed and likewise the path followed for any speed command depends on curvature.

These issues are cast in the traditional robotics terms by computing the impact of the dynamics constraint on the size of the feasible subset of configuration space.

The conclusion is that for a typical maneuver, 98% of C space is not feasible. As in other matters concerning dynamics, this is only bad news if it is not understood. Once it is understood, its impact is that dynamic feasibility is a very powerful heuristic for reducing the search characteristic of motion planning.

6.4 Positioning Fidelity

This section considers the fidelity requirements on the position estimation system. It concludes that relative position accuracy on the order of 1% of distance travelled and absolute attitude accuracy on the order of 1° are required.

46. That form need not be an explicit differential equation. The main point is that the system must compute the correct system transfer function - however that is achieved.

Section 7: Interactions

While the previous sections have considered the four requirements of safety independently, this section takes the point of view that they actually must be met simultaneously and, in effect, can be substituted into each other. This leads to some general conclusions about the nature of the problem.

7.1 Rationale For a Feedforward Approach

The **dynamics constraint** is a catchall term to express the fact that system behavior is governed by a differential equation and the realities of the latencies of the computer system on which it runs. Any response trajectory which satisfies these set of constraints is said to be **dynamically feasible**.

The typical vehicle is fundamentally a filter and it operates entirely in the transient regime so poor command following must be expected. Given that the **command following problem** has no solution as a matter of basic physics, there remains the question of what can be achieved through an accurate model of its admittedly poor response.

The use of such a model has traditionally been called **feedforward**. It leads to an expression of motion planning for obstacle avoidance purposes in terms of a search over candidate commands expressed in **actuation space** for safe or collision-free alternatives. Such an approach uses **forward modelling** to map commands onto response, and introduces the following benefits:

- Vehicle control remains stable at high speeds.
- Higher fidelity models enable higher performance in hazard avoidance and goal-seeking because the system can very precisely predict its own motion. This leads to the ability to drive close to hazards when necessary and the ability to track paths with low error.
- System reliability is enhanced because dynamic feasibility is inherent in forward modelling approaches. No time is wasted generating infeasible alternatives so response time is enhanced.
- Computational complexity of planning is reduced because the dynamics constraint is a valuable heuristic to limit search. It is known *a priori* that the system cannot respond to frequencies above the natural frequency so such alternatives need not be considered.

The major conclusion of this section is that the traditional hierarchical robotic architecture consisting of a strategic planner connected directly to actuator level control will not work when the **reaction distance** is large relative to the fidelity requirements imposed by obstacle avoidance and goal seeking tasks. This conclusion is the organizing principle behind the design of the navigator's planning subsystem.

7.2 Rationale For Adaptive Perception

This section presents an analysis of the computational complexity of perception based on the observation that response, throughput, and acuity requirements must all be met simultaneously. By substituting these requirements into each other, it has been possible to show that the complexity of range image processing is basically polynomial in the **reaction distance**.

The policy of guaranteed safety implies that throughput is proportional to a high power of velocity because:

- Maximum range increases quadratically with speed (because braking distance does).

- Pixel size decreases quadratically with maximum range (in order to resolve obstacles).
- Throughput increases quadratically with pixel size (assuming fixed field of view).

This sensitivity of throughput to speed has been called the **throughput problem**.

There are four major conclusions of this section.

- It has been shown that to adopt a policy of guaranteed vehicle safety is to adopt a computational complexity of $O([TV]^N)$ for range image processing where T is the vehicle reaction time and V is the velocity.
- This result implies that increased vehicle speed will require nonlinear growth in computational bandwidth.
- This result identifies the fundamental trade-off of finite computing resources as one of speed for either resolution or reliability.
- Quantitative theoretical comparisons of various image processing approaches have identified that **adaptive scan** and **adaptive sweep** are very important mechanisms that promise to improve system performance.

This last conclusion is the organizing principle behind the design of the navigator's perception subsystem.

7.3 Rationale For a Real-Time Approach

This section identifies an effect referred to as the **computational spiral effect** which quantifies the cost of more sophisticated processing algorithms in terms of their effect on reaction time and therefore on the processing required. The essential argument here is that the problem has stringent real-time constraints and that trying to be overly smart in software is not so smart in the final analysis.

Luckily, there is a classical approach to this dilemma in real-time systems. The tradeoff between throughput and response is managed by reducing throughput to a bare minimum in order to reduce response time to a bare minimum. This **real-time approach** is the organizing principle of the entire navigation system.

PART III: Intelligent Autonomous Mobility

Performance of high-speed rough terrain autonomous vehicles is compromised by inadequate real-time response characteristics, poor models of vehicle dynamics and terrain following, and suboptimal allocation of limited computational resources. An approach to high-speed rough terrain autonomy is presented which solves these problems through a real-time predictive control approach and the adaptive perception that makes it possible.

RANGER is an acronym for Real-time Autonomous Navigator with a Geometric Engine. It is a computer program which allows a vehicle to intelligently drive itself over rugged outdoor terrain. This part describes the overall design of the navigator. The system is composed of:

- An adaptive perception algorithm which computationally stabilizes the sensor sweep and in doing so permits highest vehicle speeds. The algorithm supports either laser range-finder or external stereo vision inputs and an internal stereo vision system is also incorporated.
- A state space feedforward optimal controller, which implements obstacle avoidance, path planning, path generation, path following, goal seeking, and coordinated control of steering, brake, throttle, and sensor pan/tilt actuators.
- A navigation Kalman filter, which generates a vehicle position estimate based on all available information.
- A graphical simulation environment.
- A development support environment.

The simulation environment includes:

- Animated X11 based wireframe and image graphics
- Visible surface ray tracing sensor simulator
- State space vehicle kinetic simulator
- Fractal geometry terrain simulator

The development environment includes:

- General event server for ease of porting the system to different vehicles
- Data logger for off-line analysis of field tests
- C language interpreter for configuration and parametric modification
- Command line configuration parser

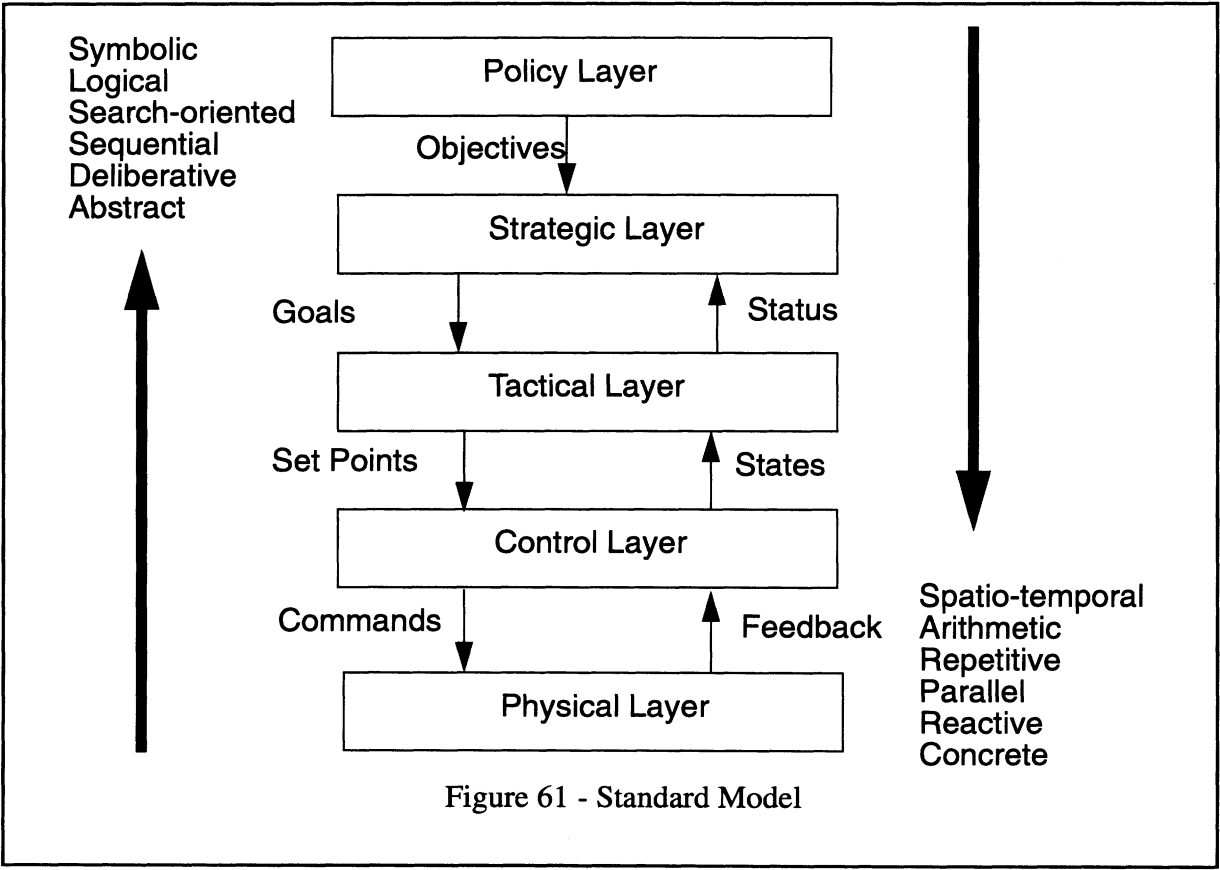
Chapter 1: Concept

This section discusses the problem of intelligent autonomous mobility from an architectural perspective.

Section 1: Standard Architectural Model

There is a tendency in the field of autonomous navigation at this point in its history to consider the entire problem from the perspective of control theory as well as an opposing tendency to consider the entire problem from the perspective of AI. This section considers some basic architectural matters in the context of high-speed autonomy and argues that both perspectives have their place.

Consider the following **standard model** for a class of architectures that are suitable for the implementation of intelligent autonomous mobility.



Generally speaking, this architecture is a hierarchy. This architecture is obvious and virtually universal and it provides a convenient perspective for organizing the exposition. It also provides a perspective on where the thesis fits into the larger picture of autonomous vehicles.

Higher levels of the hierarchy tend to be characterized by computation that is more symbolic, logical, search-oriented, sequential, deliberative and abstract than lower layers. Lower layers tend to be characterized by computation that is more spatial or temporal, arithmetic, repetitive, parallel, reactive, and concrete than higher layers. As a general rule, higher layers exhibit longer reaction times and longer cycle times.

The issues involved in assignment of system requirements to a particular layer are complex and interdependent. One useful principle for the solution of the problem of assignment of update rate might be called the **principle of characteristic times**. Here, this principle exploits the fact that although more abstract decision-making tends to require more time to accomplish, many problems can accommodate the time required.

Real-time analysis teaches us that event interarrival times are a basic consideration. If a system must respond to an event within some period, the system component that addresses that requirement must exist in the appropriate layer having an appropriate cycle time. Similarly, AI planning teaches us that there are times when the future consequences of actions taken now must be considered in order to avoid ultimate system failure. The design issue is to decide when these elements are and are not operative, in which parts of the larger problem they are operative, and how best to meet all requirements simultaneously and optimally.

1.1 Policy Layer

The **policy layer** is effectively omniscient. It enacts fundamental incontrovertible laws. It generates the mission objectives, collectively called policy - the overall raison d'etre of the system. These objectives may be encoded deliberately, they may emerge from the subsystems, or they may be supplied by human direction. The important distinction here is that the goals specified by this layer are highly abstract and not usually subject to much compromise. Examples of such goals are "stay alive", or "find the X".

Failure to meet these objectives are mission level failures. Policy does not change often and can usually be considered constant over the mission. Other mission failures may arise in the policy layer through the omission of objectives or the specification of inappropriate ones. In any case, because policy is not normally reversible over a mission lifetime, mission failures tend to be absolute. Techniques used for policy generation are elusive and poorly understood - largely because the important aspects of the problem itself are poorly understood. Examples of policy-generating mechanisms are^{1 2}**generate and test, mission analysis**, and the **minimum potential energy principle**.

Computations involved in policy generation, if any, are considered to take place outside the system at this level of conceptualization.

1. An example of the generate and test mechanism is biological evolution. Species tend to migrate with the seasons because it has some supposed evolutionary value. In human endeavors, technological evolution continues to refine solution concepts based on past experience. This permits incremental improvement even though the larger problem remains only partially understood.

2. Mission analysis is a human activity which would choose, for example, the area of a planet to be explored. The minimum potential energy principle is an organizing principle of thermodynamics which all systems tend to follow from an energy storage/conversion perspective. This principle governs everything from the deflection of a beam to the dynamics of a spacecraft to the climate of the earth.

1.2 Strategic Layer

The **strategic layer** corresponds to the deliberative, logical, goal-generating component of autonomous systems. It concerns itself with the larger picture within the confines of policy, with avoiding local planning minima, with overall optimality, and with modelling and memory of the environment. Relative to lower layers, this layer can afford (or is obliged) to consume more time in its deliberations.

Implementation of the strategic layer lends itself to the techniques of Artificial Intelligence and Operations Research. Problems solved at this level tend to be characterized by search of a large number of alternatives and by overwhelming computational complexity. Heuristic techniques are often used to attempt to reduce that complexity to manageable proportions. Two of the clear lessons of research on problems solved in this layer are that:

- An appropriate representation of the problem can make a major difference in the quality of the solution or in the time taken to compute it.
- The exploitation of constraints can be a very powerful heuristic.

Algorithmic results applicable to the strategic layer tend to concentrate on concerns of:

- **completeness** - Is a solution guaranteed to be found if it exists?
- **soundness** - Does the solution satisfy the constraints imposed?
- **optimality** - Will the chosen solution be the best one available?

One of the organizing principles of work on the strategic planning layer is the **principle of optimality** of dynamic programming which states, loosely, that an optimal solution to the whole problem must be composed of optimal solutions to its subproblems³.

1.3 Control Layer

The **control layer** corresponds to the real-time command following component of autonomous systems. It might also be called the **logistic layer**. It concerns itself with the immediate low level issues of how much current should be applied to actuators and generally, with doing exactly what it is told to do to the best of its ability. This layer has extremely limited autonomous authority which may be limited to such actions as cutting actuator power in an emergency when it has determined that it is unable to follow commands for a variety of possible reasons.

Implementation of the control layer lends itself to the techniques of automatic control theory. Problems solved at this level tend to be characterized by differential equation models of system dynamics and by issues of response time and control bandwidth. One of the basic lessons of research on problems solved in this layer is that feedback control makes it possible to alter system dynamics and cause a system to follow its commands faithfully in the presence of a large number of different nonidealities and disturbances.

Algorithmic results applicable to the control layer tend to concentrate on concerns of:

- **controllability** - Can the system do what it is told?
- **observability** - Can it know what its doing?
- **stability** - Will it lose its ability to control itself?

3. The A* algorithm of AI planning is related to dynamic programming through this principle. The principle is often not satisfied in practice but it is a very useful concept.

One of the organizing principles of work on the control layer is the **sampling theorem** which states, loosely, that both sensing and actuation bandwidth must be commensurate with the underlying frequency response of the plant being controlled.

1.4 Tactical Layer

One important aspect of the standard model from the perspective of the thesis is the introduction of the **tactical layer**. An upcoming section will be devoted to this layer.

Section 2: Bandwidth - Conceptualization Gap

A set of architectural design issues emerge for the problem addressed by the thesis which will be referred to as the **bandwidth-conceptualization gap**. The origin of this term is that the strategic layer usually cannot generate the bandwidth to respond to immediate concerns while the control layer cannot conceptualize the world in rich enough detail to enable useful strategic response. In other words, the strategic layer needs more *bandwidth* and the control layer needs more *time* and neither layer has the luxury of these resources in practice.

The real architectural design issue in a hierarchical software system is to allocate responsibility and authority appropriately to each layer. Some of these issues are outlined below by examining why it is difficult to merge the strategic and control layers into one layer.

2.1 Fast Execution Monitoring and Path Replanning

The historical technique of **execution monitoring** is to monitor the degree to which lower levels follow the plans given to them, and to react appropriately. **Path replanning** is an answer to the problem that the strategic layer may have limited or no prior knowledge of the environment at the resolution of obstacles. In the extreme the system can attempt to update the specification of the strategic plan at high enough rates to capture unknown obstacles and replan the task.

At some point execution monitoring and path planning begin to look like the feedback and control elements of feedback control, and at this point the strategic layer begins to do the job of a controller. There are several practical problems with the idea of fast replanning.

- It violates the **principle of characteristic times** because it is not often the case that a small amount of change in the execution of a plan will cause a large difference in optimality. Most problems are well-behaved so fast replanning is not necessary and therefore wasteful of computing resources.
- It is not often the case that an environment or task that is rich enough to require a strategic layer can also replan its actions at sufficiently high rates to respond to execution errors. The strategic layer cannot generate the necessary bandwidth anyway.

2.2 Smart Control

Instead of a faster planner, one might attempt to eliminate planning by adding smarts to the control layer. This idea also has historical precedent. A simple example of this approach is the notion of the **potential field**. In this approach, the world is modelled in terms that relate very directly to the control task. Some potential field model is generated and the controller is smart enough to follow its gradient in order to achieve its goal position. There are several practical problems with this approach.

- It cannot handle rich goal specification or even intelligently update its goal based on changes in the environment. It can be argued that the best a potential field can do is to follow a strategic plan, and the generation of that plan is outside the scope of the technique.
- It cannot handle rich worlds. The essential assumption of this approach is that the world will succumb to a locally linear model and a globally unimodal model. Yet, local minima exist in real applications and these are the historic failure models of such systems.

Section 3: Tactical Control Layer

While the two layer hierarchy can be justified with the preceding arguments, the practical fact of the matter is that the bandwidth-conceptualization gap still cannot be spanned by only two layers. The basic reason for this is the **command following problem** and the solution is the introduction of a third intermediate layer - tactical control.

3.1 Command Following Problem

If the strategic layer talks directly to control, the control layer is not able to execute its commands in the target application of high-speed autonomous mobility. A situation is created where the strategic layer cannot do its job of directing the machine and the control layer cannot do its job of following its commands faithfully.

It is tempting to propose that a system can live with the command following problem. The control layer could be made more cognizant of its own limitations and reject commands outright that cannot be executed. The major problem with this approach is the inefficiency inherent in **thrashing**. Considerable resources are typically expended in the generation of the strategic plan and it is not often possible to generate and test alternatives until the controller is satisfied. One can attempt to blame control for being stupid or strategic planning for being slow but this does not solve the practical problem.

3.2 Tactical Control layer

There are certainly times when the goals specified by the strategic layer must be ignored because it is not aware of the immediate environment - but the control layer is not smart enough to deal with the matter. There are also times when the control layer is unable to follow its commands - but the strategic layer is too slow to deal with the matter.

The solution to this problem is to have a layer smarter than control and faster than planning. This tactical layer⁴:

- Views the goals from the strategic layer as recommendations that it may be override when the situation demands it.
- Incorporates sufficient bandwidth to ensure vehicle survival at the coordinated actuator control level.
- Incorporates a sufficiently accurate model of vehicle dynamics that it understands and adapts to the inability of the controller to follow its commands.

This three layer architecture imparts a degree of autonomy to the layer below the strategic to allow it to implement basic survival without much regard for the mission imperatives. The basic assumption is that survival is not such a continuous issue that mission imperatives will be ignored for long because if that was the case, the system would not be viable anyway.

4. The title of the thesis derives from these properties of the tactical control layer. The navigator resides almost entirely in this layer. It is *intelligent* enough to understand vehicle dynamics and a local world model, *predictive* enough to see consequences as far ahead as a few vehicle response times, and based on multivariate *control* theory.

3.3 Hierarchical Time Spectrum

A useful way to conceptualize the three layers is in terms of their **lookahead** and **update rate**. Each layer is concerned with an amount of temporal and spatial lookahead and updates its output at some appropriate rate.

The strategic layer operates on a **strategic time** scale. It considers the future consequences of its actions up to the point where the policy objectives are predicted to be achieved. It updates its plan of action once over the minimum amount of time for which a change in the plan can significantly affect overall optimality. A rough measure of the magnitude of strategic lookahead is the mission duration.

The tactical layer operates on a faster **tactical time** scale. It considers the future consequences of its actions up to the point where steady-state response of the entire vehicle has been achieved. It updates its plan of action once over the minimum amount of time for which a change in the plan can significantly affect the vehicle response trajectory. A rough measure of the magnitude of strategic lookahead in the system reaction time.

The control layer operates on an even faster **control time** scale. It may or may not consider the future consequences of its actions up to the point where steady-state response of each actuator has been achieved. It normally updates its output at the highest frequency that the plant can pass. A rough measure of the magnitude of control lookahead, if any, is the actuator response time.

3.4 Uncertainty

It is important to distinguish several kinds of uncertainty in autonomous systems because there is a tendency to associate deterministic error that can be modelled well with random error that cannot. This distinction is raised here because the tactical layer models dynamic command following error that is often considered to be random uncertainty. Generally, information may be:

- random - inevitably random as a matter of physics
- unknown - not imparted to the system through sensing or encoded knowledge
- unknowable - outside of the parameters measurable or known

An example of random information is the fundamental physical noise present on a CCD array in a camera. Unknown information includes obstacles that cannot be predicted because they are not represented in the prior environment model and not currently in view, or because they have moved. Unknowable information includes information for which no sensing technology exists or no prior knowledge exists.

It is the case that techniques which deal with one form of uncertainty tend to be useful for other forms, but it is nonetheless important to understand which form of uncertainty is operative. For example, while models of random uncertainty may perform usefully in accounting for actuator dynamics, the underlying process is a deterministic one completely governed by differential equations. A properly calibrated dynamic model would be expected to outperform a random model. Indeed, one of the most basic premises of calibration is to remove all systematic error possible before modelling the remainder as random.

For this reason, the view taken here is that command following error is not random and the tactical control layer can deal with it very effectively.

Chapter 2: Design

This section outlines an approach to intelligent autonomous mobility based on intelligent predictive control. Before elaborating on the design, a brief characterization of the problem is presented. Given this, the major design decision is to implement an intelligent predictive control approach.

Section 1: The Nature of High-Speed Autonomy

It has been shown that *high-speed autonomy is a real-time problem* because response time requirements are stringent and throughput and response time depend on each other. Contemporary sensor technology coupled with the latencies of sluggish massive vehicles and distributed control schemes limit vehicle speeds. Computational resource limitations are severe at even moderate speeds and a real-time approach is indicated which minimizes response time and maximizes sensor lookahead. Through the adoption of a managed minimum reaction time strategy, an optimal system becomes a real-time system.

It has been shown that *optimal high-speed autonomy requires adaptive, minimum throughput perception* because the computational complexity of range image perception is severe when measured against contemporary general purpose computing hardware and non adaptive image processing techniques are used. Indeed, up to 98% of the information provided by contemporary sensors is redundant.

It has been shown that the *high-speed autonomy problem is a tactical control problem* because the vehicle configuration space is degenerate at high-speed and the mapping from configuration space to actuation space is not defined over approximately 97% of the extent of configuration space. Search-based C-space planners based on many fine AI algorithms are brittle and waste resources because the clothoid generation problem is impossible in practical terms and most vehicle configurations in C-space are not dynamically feasible. Therefore, an optimal system considers path feasibility before obstacle avoidance. In optimization terms, such a system considers the constraints before the utility function, and in doing so, becomes a controller.

It has been shown that *high-speed autonomy is a feedforward control problem* because algorithmic stability of path tracking and obstacle avoidance can only be achieved by high fidelity models of vehicle actuator dynamics.

It has been shown that *high-speed autonomy is a state space control problem* because a high-speed vehicle satisfies a coupled nonlinear multidimensional differential equation. An optimal system models this equation explicitly or implicitly, and in doing so becomes a state space controller.

Section 2: Intelligent Predictive Control

One of the axioms of the thesis work is that an intermediate layer between strategy and control is both useful and required, and most of the thesis work falls into this layer. This layer exists to bridge the **bandwidth-conceptualization gap** between the strategic and control layers. These two layers speak different languages. They conceptualize the world on radically different time scales and spatial resolutions. An intermediate layer is needed to connect them together and translate information between them for the problem of high-speed autonomous mobility. Indeed, the introduction of this layer is the most basic aspect of the problem solution.

The navigator is a high-level intelligent predictive controller. That is, it closes the overall perceive-think-act loop for a robot vehicle. It is not concerned with the specific control of actuators themselves, so it is not in the control layer. Further, while it can accept a specification of a strategic goal, such as a path to follow, or a direction to prefer, it cannot generate its own strategic goals, so it is not in the strategic layer.

The navigator solves the **local planning problem** for autonomous vehicles. That is, the problem of deciding what to do based only on what can be seen at the moment within the field of view of the environmental sensors. As a tactical layer entity, the system incorporates elements that might normally exist in one of the other two.

2*1 Deliberative Elements

Like a strategic planner, the system performs an amount of search and heuristics are employed to reduce that search. Like a strategic planner, the system models the environment and responds to it, so it merits the designation *intelligent*. Like a strategic planner, the system considers alternatives in an abstract space that is a transformation of reality - in this case, actuation space instead of the configuration space commonly used in strategic motion planning.

Like a strategic planner, the system considers the consequences of its actions. However, unlike the logical rules of precedence typical of at least symbolic planners, the system employs time continuous precedence information in the form of a feedforward system dynamics model. The state space model employed is more like the kinds of models used in controllers.

Like a strategic planner, the system employs some memory of the state of the environment. However, the planning horizon is limited to the distances that can be traversed within the vehicle reaction time, so the map used merits the designation *local*.

2.2 Reactive Elements

Like a controller, the system models the vehicle with a differential equation. However, because it concerns itself with the coordinated control level, the equation is multidimensional. This level considers the overall impact of the response of all actuators simultaneously and is typically far less intelligent in a manipulation system.

Like a controller, the system is very concerned with response time and throughput management as is common of real-time systems. However, the response and throughput issues are considered at the level of the entire vehicle system instead of individual actuators. Like a controller, the system is concerned with latencies and time tags and the precise timing of events.

Like a controller, the system concerns itself with command following. However, the commands it accepts are from the strategic layer, and are therefore specified abstractly - in terms of a path to follow or a direction to prefer. These commands are taken as recommendations and may be temporarily overridden by the tactical layer. The control layer could not accept such commands unless it maintained a coordinated control function that knew how to transform actuator states into vehicle states.

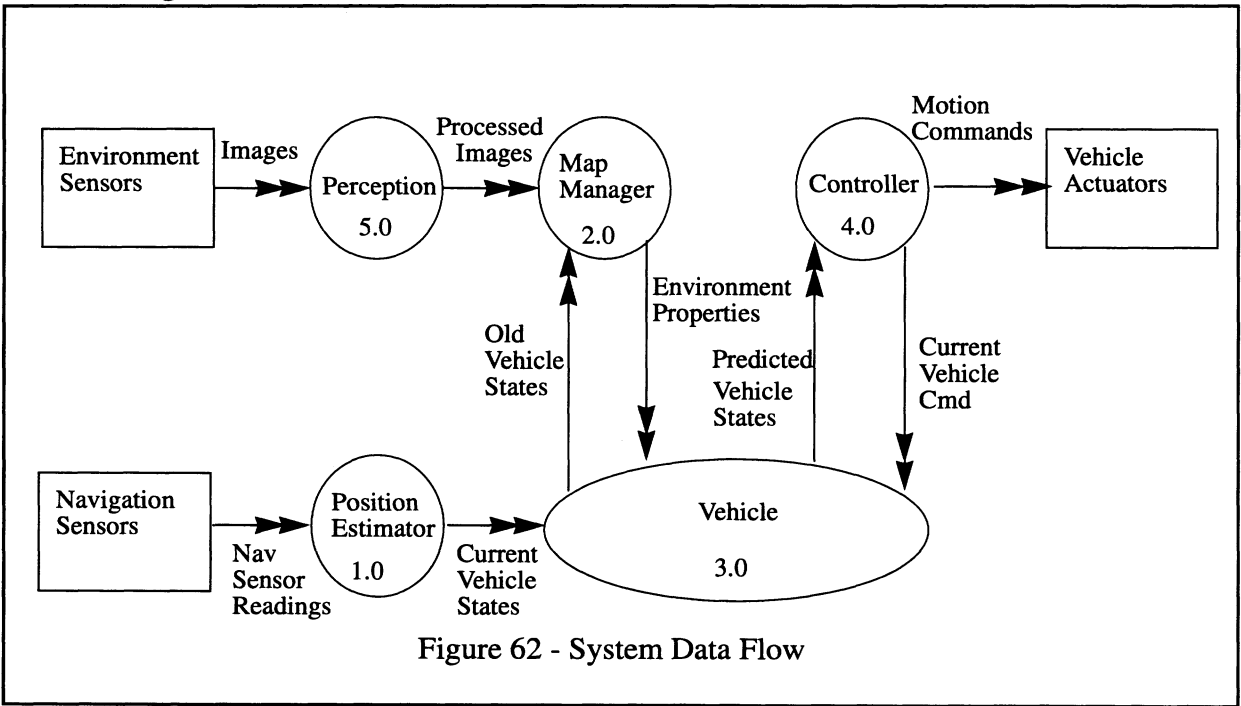
Chapter 3: Implementation

When the various development features are removed, the remaining system modules are suitable for embedded real-time environments. These core modules are called the **real-time kernel**. This section provides an overview of the real-time kernel which follows the data flow diagram convention of Ward and Mellor for real-time systems [81].

The diagram convention used here and throughout is the **data flow diagram**. In these diagrams, the bubbles represent transformations of signal flows, and arrows signify data. Each diagram has an implicit update frequency or characteristic cycle time associated with it. For the present purpose, consider that double arrows indicate time-continuous data and single arrows indicate intermittent data flows. Rectangles indicate external entities and parallel lines are called stores. A store indicates memory of data beyond a single characteristic cycle time.

Section 1: Real-Time Kernel

At the highest level, the system can be considered to consist of 5 modules as shown in the following data flow diagram:



1.1 Position Estimator

The **Position Estimator** is responsible for integrating diverse navigation sensor indications into a single consistent indication of vehicle state. Vehicle state information includes the positions of all actuators and some of their derivatives, and the 3D state of motion of the vehicle body. This module may be the built-in navigation Kalman filter or another system which generates the same output.

1.2 Map Manager

The **Map Manager** integrates discrete samples of terrain geometry or other properties into a consistent terrain map which can be presented to the vehicle controller as the environmental model. It maintains a current record of the terrain immediately in front of the vehicle which incorporates all images necessary, and which automatically scrolls as the vehicle moves.

The map manager forms an abstract data structure when combined with the terrain map itself. Images are transitory input flows at this level of conceptualization. Consumers read or write the map in navigation coordinates and are ignorant of the underlying transformation and scrolling operations.

1.3 Vehicle

The **Vehicle** object is both the control loop feedforward element and an abstract data structure which encapsulates the vehicle state. This module incorporates FIFO queues which store a short time history of vehicle states and vehicle commands. Old state information is required by the map manager in order to register images in space. The current vehicle state is used as the initial conditions of the feedforward simulation. Old commands are used in the feedforward simulation as well. This module also provides daemons which compute derived properties of vehicle state such as the planning range window and the maximum safe speed and curvature. Another set of daemons is used to convert curvature, steer angle, and turn radius to any of the other two based on the vehicle state and the wheelbase.

1.4 Controller

The **Controller** object is responsible for coordinated control of all actuators. This module includes regulators for sensor head control, an obstacle avoidance tactical controller and a path following strategic controller. It also incorporates an arbitrator to resolve disagreements between the latter two controllers.

1.5 Perception

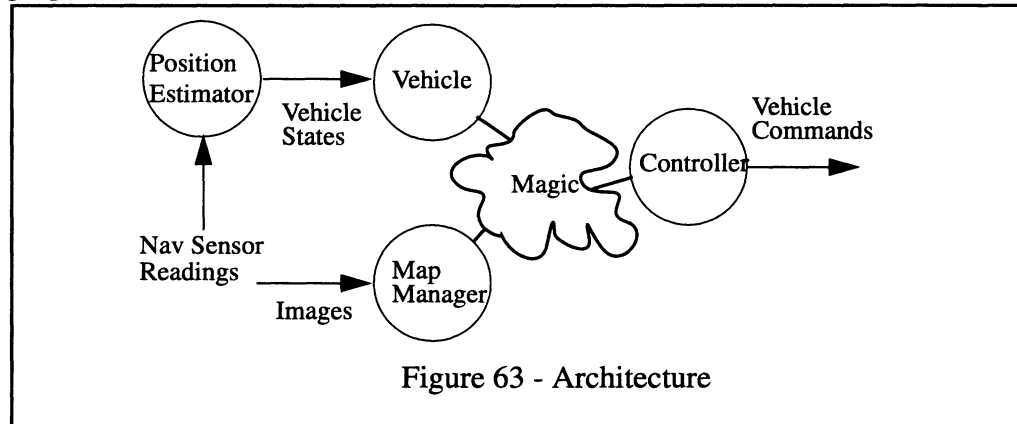
The **Perception** module is responsible for understanding or interpreting input images and putting them in a form suitable for the Map Manager to process. Examples of perceptual preprocessing include stereoscopy (stereo vision) which computes range from two or more images taken from disparate viewpoints, and terrain-typing which labels each pixel in an image as rock, road, shrubbery or tree. Stereo vision is the only perception that is currently supported.

Section 2: Rationale - Bandwidth Requirements

The system is divided into these high-level objects because each has different response and throughput requirements. Each can be implemented as a separate process, and all can run asynchronously. Three main data streams connect the system to the physical world. These are:

- image input stream
- vehicle state input stream
- vehicle command output stream

Additionally, there may be sensor head state input streams and sensor head command output streams. Perception and the Map Manager are serial transformations that may be treated as a unit. The real-time clock is also considered to be an external device. The architecture is indicated in the following figure:



2.1 Image Input Stream

The frequency requirement of the image input stream is the rate at which images are generated. However, because images generally overlap on the ground, it is not necessary to process all of them. The adaptive perception mechanisms will minimize the amount of image information processed if they overlap so the practical frequency requirement of the image input stream is the guaranteed throughput requirement converted back into image space.

2.2 Vehicle State Input Stream

The frequency requirement on this stream comes from the need to remove image distortions due to vehicle motion both during and since images were captured. Normally, this frequency is much higher than the image frequency requirement so *it is not feasible to consider image processing to be indivisible* and measure vehicle state only between images.

2.3 Vehicle Command Output Stream

The frequency requirement on this stream comes from the need to track commands faithfully. In practice, little is achieved against control fidelity by attempting to drive the vehicle faster than it can respond. Thus, the practical requirement on this stream is related to the vehicle natural frequency and the sampling theorem. Roughly ten times the vehicle natural frequency is appropriate for each controlled degree of freedom. The controller need not run any faster than this.

Section 3: Test Environment Layer

The **test environment** is the highest level of conceptualization of the system. It externalizes system interfaces to sensors, actuators, displays, and the file system in order to simplify porting to different environments. Ideally, the system is ported by changing only these interfaces. This layer provides a buffer layer which lies between the real-time kernel and the physical i/o drivers. The elements of this layer are:

- the main event loop and associated world tree
- the i/o filter which consists of:
 - simulators
 - data logger
 - physical i/o to the real vehicle
- development support tools such as:
 - graphics facilities
 - configuration parser
 - command line parser

The `main()` function may reside in this layer, if necessary. The main event loop is usually implemented in this layer.

It optionally:

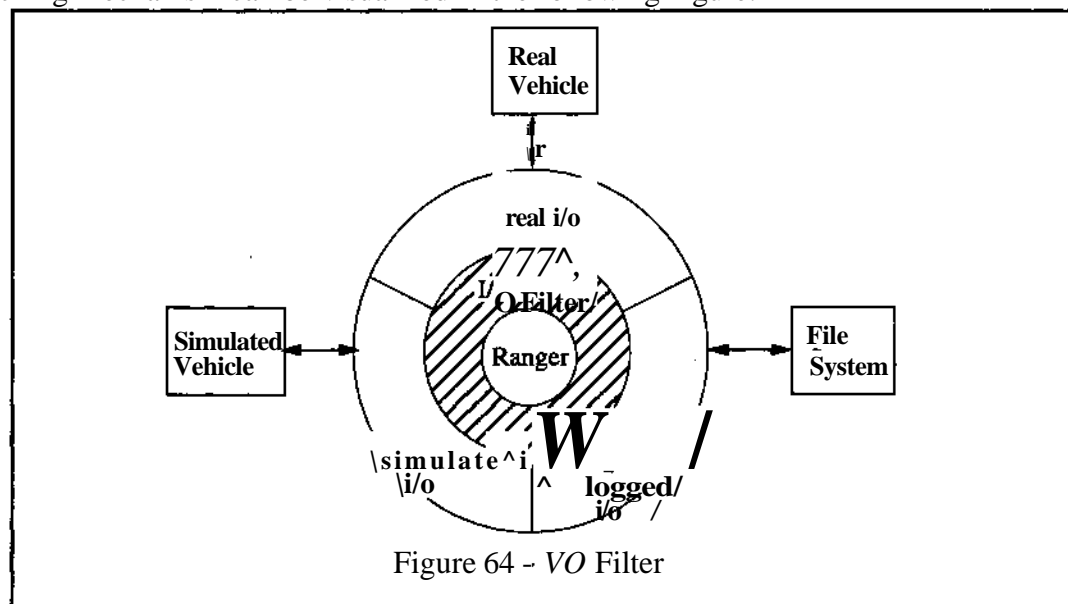
- intercepts data for viewing purposes
- generates simulated data for nonexistent hardware modules
- records data to the file system
- plays data back for off-line review and analysis

It is the responsibility of the test environment to supply sensor measurements and to dispatch commands. The existence of simulation, parameter editing, and data logging facilities is optional. None are needed to run the system.

This layer insulates the system from the knowledge of whether it is embodied in real hardware or executing in real time. It also insulates the system from the knowledge of whether or not it is the highest level control loop.

3.1.1 I/O Filter

Any external data interface can be implemented in simulation, from logged data on the file system, or from the hardware - and the knowledge of which is being used is very localized. This data dispatching mechanism can be visualized in the following figure:



3.1.1.1 Data Logger

The **data logger** is one service that may transparently intercede between the generation of an event and its dispatch. Its design is subject to four simplifying design decisions:

- determinism: because all input data including the clock is recorded, the system is guaranteed to produce the same output given the same input.
- zero state start-up: only the initial state is guaranteed to be repeatable so the logger cannot be enabled and disabled dynamically.
- parameter constancy: some mechanism outside the data logger must ensure that system parameters are consistent between record and playback.
- file layout constancy: changes to the layout of logger data files renders old data files unusable.

3.1.1.2 Simulator

The simulator is another service that may intercede between the generation of an event and its dispatch. If simulation is on, the physical i/o layer generates a pointer to an empty event which is the correct event in its simulation sequence. The simulator will fill in the event with simulated data. Various simulators are implemented inside their associated objects. The vehicle simulator is necessary to run the system, but sensor and terrain simulation are not.

3.1.1.3 Physical I/O Layer

If neither the simulator nor data logger intercedes, events are managed by the physical i/o layer. The physical i/o layer binds the real-time kernel to the physical sensors and effectors. The system does not know if network, backplane, or board-level communications are involved in this connection.

3.2 General Rationale for Physical I/O

There are a few general principles imposed on the input/output services. This section outlines these requirements.

3.2.1 Vehicle States

Image digitization for scanning laser rangefinders may take significant time and the vehicle moves up to 5 meters over this time at 10 m/s. Therefore, the vehicle state must be sampled while this is occurring and the state data must be supplied later on. Some agent close to the position and attitude sensors must write vehicle state information at high frequency to the state communication channel.

3.2.2 Queues

This data must be queued until the system has time to read it. The physical i/o layer must flush this state queue on every read and the Vehicle will store the data internally. The Vehicle queue may be used for this purpose by synchronous i/o channels because the queues can be accessed while the system is running concurrently.

3.2.3 Images

Image digitization is normally a synchronous process and although the Map Manager may not have time to process all images, they typically overlap on the ground so this is not a serious concern. Some agent close to the image formation hardware must write images to one of the image communication channels as soon as they are generated. The physical i/o layer must flush this image channel on every read and provide only the most recent image to the Map Manager.

3.2.4 Time Tags

All sensory inputs are ideally time-tagged at the source. A consistent system time standard is assumed, however it is generated. Consistency of time stamps is necessary within each data stream, and between data streams. This allows the system to register contemporary events in time even though they may have arrived at disparate times. It also allows the system to reason explicitly about delays and their effect on the requirement to maintain high-fidelity control.

Images are tagged with the time that image formation commenced. States are tagged with the instant in time to which the states correspond. Tagging agents must account for their own latency in their time tags.

3.2.5 Sample Rate

Sensory inputs must be sampled at the Nyquist rate, and this rate differs for different sensors. The system is not fast enough to do this itself, so i/o channels are responsible for a guarantee of high resolution sampling. Timers hidden in the physical i/o layer can accomplish this by sampling at regular frequency.

3.2.6 Concurrency

The map manager and controller objects copy the entire vehicle data structure at the start of their cycles, so it is admissible to queue data into the vehicle queues while these objects are running concurrently.

3.3 Main Event Loop

A central event loop, called the **main event loop**, performs the following functions:

- reads vehicle state information at high, roughly synchronous rate and stores it in the vehicle queues for later processing
- reads images from diverse environmental sensors as soon as their data is available
- reads the clock from time to time as needed

The event loop model is used for two reasons:

- Determinism can be guaranteed for data logging purposes because all system i/o must pass through it.
- Response time can be minimized by an event-reaction model.

When an image is available, the event loop calls the map manager to request integration of the image into a single consistent description of the local environment. Perhaps after the image is processed, or in general at any time, the event loop calls the controller to cause it to update the commands being sent to the vehicle.

3.3.1 World Tvee

The object library and all layers below it makes no assumptions about the configuration of the vehicle - although its dynamics are encoded in the state space model. Therefore, it is the responsibility of the test environment layer to specify the configuration of the world at start up. The initialization step involves the creation of a **world tree** that represents the physical connections of objects with each other and their associated attributes.

3.3.2 Events

To enhance portability, the physical i/o interface is defined in terms of a generic io_datum object which serves double duty as a place holder for events because, currently, all events are associated with external i/o. This permits automatic management of several sensors without having to assume a fixed number in the interface specification. It also permits the data logger to read the datum type from the log file and discover its size so that it knows how to read the data itself.

3.4 Development Support Tools

The following development support tools are available:

3.4.1 Parameter Editor

A C language interpreter is available as a general tool for parsing configuration command strings. This can be very useful for avoiding recompilation in the field and simplifies software configuration control. A special set of built-in functions permit the specification of simulated terrain of arbitrary complexity.

3.4.2 Graphics Tools

Graphics tools are available for displaying images, maps, and vehicles. These are not necessary to run the system. They include image, spline, wireframe surface objects and general 2D and 3D rendering primitives.

Section 4: Computational Hardware

The design of the system makes it retargettable on an entire spectrum of computational engines.

4.1 Engineering Workstations

The entire system, including perception, planning, and control, runs in less than 50 msec on a SPARC 10 platform and this is already a very high cycle rate for the entire perceive-think-act loop. The Kalman filter runs at a rate of 100 Hz in dead reckoning mode with periodic updates from sensors at a rate of 10 Hz.

4.2 Implementation in Special Purpose Hardware

Higher performance can be achieved by porting the system to special purpose hardware. The control laws are implementable solely in terms of matrix algebra, trigonometric functions and a few signal operators - because they really are just one big differential equation. For example, a graphics rendering pipeline would perform all of the necessary geometric transformations on image sensory data to generate the map data structure. The system model is literally a discrete matrix differential equation, and the hazard model can be implemented as a matrix. The arbitrator could then be implemented in a digital signal processor.

4.3 Implementation in Parallel Hardware

In the case of the perception system, the entire processing cycle from image to map can be performed in parallel at the pixel level. In the case of obstacle detection, each candidate command trajectory can be computed in parallel. The position estimation and goal seeking aspects of the system are not suitable for implementation in parallel hardware.

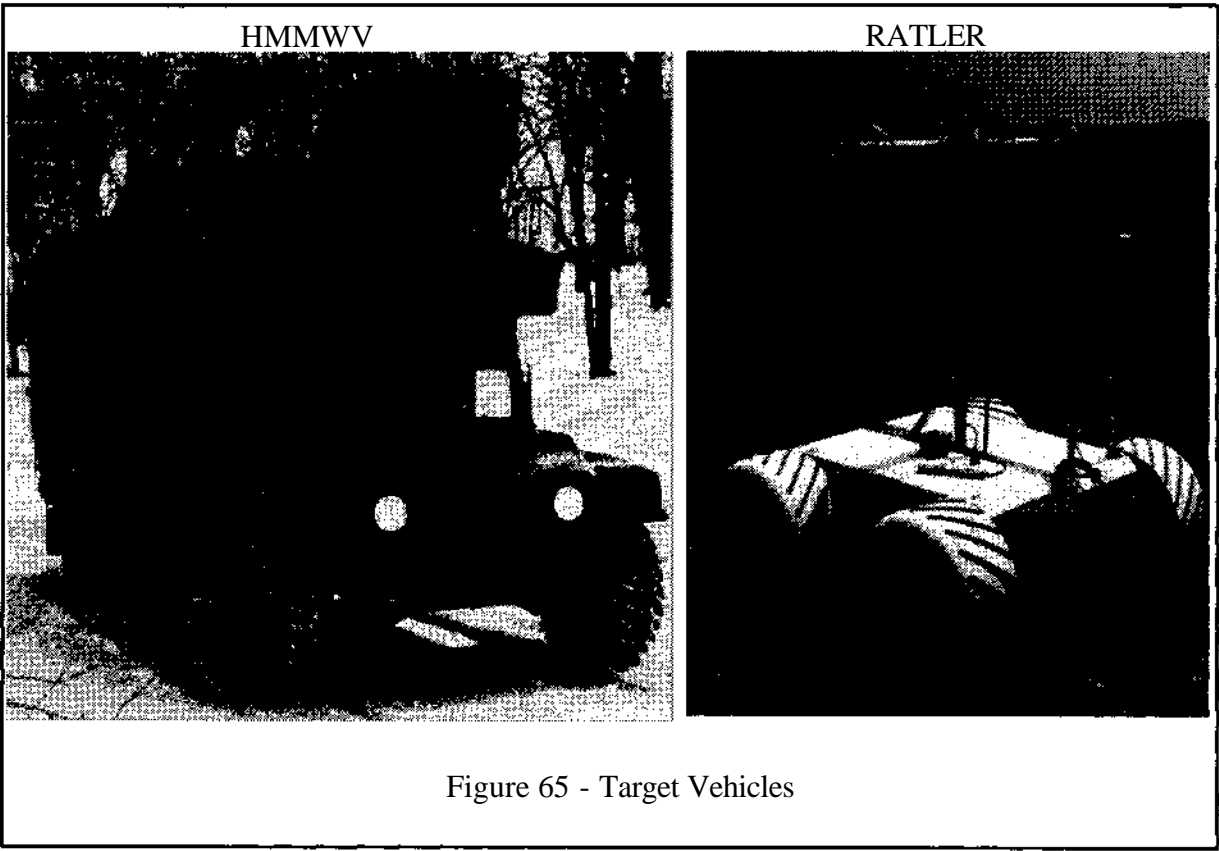
Chapter 4: Results

Section 1: Vehicles Automated

The navigator has been configured to run successfully on two very different types of vehicles:

- HMMWV (Highly Mobile Multi-Wheeled Vehicle) - a converted army jeep.
- RATLER (Robotic All-Terrain Lunar Exploration Rover) - a planetary rover. [50]

These vehicles are shown below:



Section 2: Test Run Excursion and Speed

The navigator has been used with laser range data on the HMMWV and with stereo range data on both vehicles. In the former case, excursions of 15 kilometers and instantaneous speeds of 15 km/hr have been achieved while tracking an externally supplied path that was specified at roughly 10 meter resolution. Average speed was on the order of 7 km/hr. For such long excursions, the goal seeking behavior of the navigator is used to seek some goal trajectory which is free from local minima - but not necessarily free of obstacles. While there was a detailed path for the vehicle to follow, errors in position estimation are such that the positioning system drifts a considerable distance from the path after only a few minutes of operation. This has meant, in practice, that the obstacle avoidance system is challenged regularly.

On the RATLER vehicle, the navigator is configured to run under human supervision. In this case, semi-autonomous excursions on the order of 1 Km have been achieved and high speed is not a goal of this work.

The Kalman filter has been implemented in stand alone mode and used as a basis for implementing autonomous vehicle convoys. The sensor suite used included an AHRS for indicating attitude and heading, a transmission encoder and doppler radar both for indicating vehicle speed, a video crosstrack indicator based on a roadfollowing system, and a GPS receiver. In this mode, the filter cycles at 100 Hz and provides sufficient accuracy for blind operation of the follower vehicle over excursions of about 1 kilometer.

Section 3: Real-Time Performance

The real-time performance of the system is demonstrated in the following tables. With adaptive perception enabled, using oversampling factors of 2 both horizontally and vertically⁵, the overwhelming computational cost is in the tactical feedforward associated with adaptive regard.

While adaptive perception resamples a range image for optimum coverage of the terrain, the specific attributes of the range sensor used for this run are given in the table below:

Table 9: ERIM Range Sensor Parameters

Attribute	Value
Image Rows	64
Image Cols	256
HFOV	80°
VFOV	30°
HIFOV (angular resolution)	0.3125°
VIFOV (angular resolution)	0.4688°
Frame Rate	2 Hz

Adaptive perception and adaptive regard are affected primarily by the following parameters:.

Table 10: Configuration Parameters

Attribute	Value
Vehicle Speed	3 m/s
Imaging Density	1.2
Column Oversampling Factor	2
Row Oversampling Factor	2
Map Resolution - x	0.75 m
Map Resolution - y	0.75 m.
# Turn Commands	15

5. Oversampling of 2 implies, in this case, that 4 range measurements fall into a 0.75 meters terrain map grid cell on average.

The run-time of the system under these configuration parameters is summarized below:

Table 11: Run-Time (SPARC 20)

Subsystem	Value
Perception	0.064 sees
Strategic	0.002 sees
Tactical - Command Generator	0.003 sees
Tactical - Feedforward	0.125 sees
Tactical - Arbitrate	0.014 sees
Total Runtime	208 msec.

The 64 milliseconds quoted for perception apply to range image processing only - the portion of the code enabled for a laser rangefinder. Stereo vision requires significantly more processing.

Chapter 5: Summary and Conclusions

This chapter summarizes the most important content of Part III: Intelligent Autonomous Mobility.

Performance of high-speed rough terrain autonomous vehicles is compromised by inadequate real-time response characteristics, poor models of vehicle dynamics and terrain following, and suboptimal allocation of limited computational resources. An approach to high-speed rough terrain autonomy is presented which solves these problems through a real-time predictive control approach and the adaptive perception that makes it possible.

RANGER is an acronym for Real-time Autonomous Navigator with a Geometric Engine. It is a computer program which allows a vehicle to intelligently drive itself over rugged outdoor terrain. The system is composed of:

- An adaptive perception algorithm which computationally stabilizes the sensor sweep and in doing so permits highest vehicle speeds. The algorithm supports either laser range-finder or external stereo vision inputs and an internal stereo vision system is also incorporated.
- A state space feedforward optimal controller, which implements obstacle avoidance, path planning, path generation, path following, goal seeking, and coordinated control of steering, brake, throttle, and sensor pan/tilt actuators.
- A navigation Kalman filter, which generates a vehicle position estimate based on all available information.

For development purposes, the system includes a simulation environment, visualization tools, and a data logger for off-line analysis of test results.

Section 1: Concept

1.1 Standard Architectural Model

A **standard model** for a class of architectures that are suitable for the implementation of intelligent autonomous mobility has been proposed. In this model, the **strategic layer** corresponds to the **deliberative**, strategic planning part of the system and the **control layer** corresponds to the **reactive** command following part of the system. For the general problem of autonomous mobility, both deliberative and reactive components are required.

The deliberative components achieve the larger goal through the use of representations of the world, perhaps memory, and certainly lookahead in order to avoid local minima and limit cycles and other undesirable global behavior. The reactive component achieves basic survival by reacting in unsophisticated ways to unknown or changing information that, for whatever reason, could not be dealt with by the deliberative components.

1.2 Bandwidth - Conceptualization Gap

The tendency to merge these layers shows up in many proposed additions to algorithms at each layer that attempt to avoid fundamental problems of architecture. This two-layer hierarchy can be justified in terms of a **bandwidth-conceptualization gap** for which two layers are necessary.

The origin of this term is that the strategic layer usually cannot generate the bandwidth to respond to immediate concerns while the control layer cannot conceptualize the world in rich enough detail to enable useful strategic response. In other words, the strategic layer needs more *bandwidth* and the control layer needs more *time* and neither layer has the luxury of these resources in practice.

It is tempting to try to use a single fast planning layer or a single smart control layer to solve the problem, but this approach is limited in practical scope.

There are several practical problems with the idea of fast replanning.

- Most problems are well-behaved so fast replanning is not necessary and therefore wasteful of computing resources.
- The strategic layer cannot generate the necessary bandwidth anyway.

There are several practical problems with the idea of a smarter controller.

- It cannot handle rich goal specification or even intelligently update its goal based on changes in the environment.
- Local minima exist in real applications and these are the historic failure models of such systems.

1.3 Tactical Control Layer

While the two layer hierarchy can be justified with the preceding arguments, the practical fact of the matter is that the bandwidth-conceptualization gap still cannot be spanned by only two layers. The basic reason for this is the **command following problem** and the solution is the introduction of a third intermediate layer - **tactical control** which is introduced by the thesis. This layer is smarter than control and faster than planning. This layer⁶:

- Views the goals from the strategic layer as recommendations that it may be override when the situation demands it.
- Incorporates sufficient bandwidth to ensure vehicle survival at the coordinated actuator control level.
- Incorporates a sufficiently accurate model of vehicle dynamics that it understands and adapts to the inability of the controller to follow its commands.

6. The title of the thesis derives from these properties of the tactical control layer. The navigator resides almost entirely in this layer. **It** is *intelligent* enough to understand vehicle dynamics and a local world model, *predictive* enough to see consequences as far ahead as a few vehicle response times, and based on multivariate *control* theory.

Section 2: Design

This section outlined an approach to intelligent autonomous mobility based on an intelligent predictive control architecture.

2.1 The Nature of High-Speed Autonomy

It has been shown that high-speed autonomy has the following problem characteristics:

- It is a real-time problem
- It requires adaptive, minimum throughput perception
- It is a tactical control problem
- It is a feedforward control problem
- It is a state space control problem

These problem characteristics lead to an intelligent predictive control approach to solving it.

2.2 Intelligent Predictive Control

The introduction of the tactical control layer implementing intelligent predictive control is the most basic aspect of the problem solution. The navigator closes the overall perceive-think-act loop for a robot vehicle. It is not concerned with the specific control of actuators themselves, so it is not in the control layer and it cannot generate its own strategic goals, so it is not in the strategic layer.

The navigator solves the **local planning problem** for autonomous vehicles. That is, the problem of deciding what to do based only on what can be seen at the moment within the field of view of the environmental sensors. As a tactical layer entity, the system incorporates elements that might normally exist in one of the other two.

The tactical control layer is somewhat deliberative because it:

- Performs an amount of search and uses heuristics to limit that search.
- Models and interprets the local environment with a world model.
- Considers the consequences of its actions.
- Employs some memory of the state of the environment.

The tactical control layer is also somewhat deliberative because it:

- Models the vehicle with a differential equation.
- Concerns itself with response time, throughput, latencies and time tags.
- Concerns itself with strategic command following.

Section 3: Implementation

This section provided an overview of the real-time kernel - the operational part of the system.

3.1 Real-Time Kernel

At the highest level, the system can be considered to consist of 5 modules:

- The **Position Estimator** is responsible for integrating diverse navigation sensor indications into a single consistent indication of vehicle state.
- The **Map Manager** integrates discrete samples of terrain geometry or other properties into a consistent terrain map which can be presented to the vehicle controller as the environmental model.
- The **Vehicle** object is both the control loop feedforward element and an abstract data structure which encapsulates the vehicle state.
- The **Controller** object is responsible for coordinated control of all actuators.
- The **Perception** module is responsible for understanding or interpreting input images and putting them in a form suitable for the Map Manager to process.

3.2 Rationale - Bandwidth Requirements

The system is divided into these high-level objects because each has different response and throughput requirements. Three main data streams connect the system to the physical world:

- image input stream
- vehicle state input stream
- vehicle command output stream

3.3 Test Environment Layer

The **test environment** is the highest level of conceptualization of the system. It externalizes system interfaces to sensors, actuators, displays, and the file system in order to simplify porting to different environments. Ideally, the system is ported by changing only these interfaces. This layer provides a buffer layer which lies between the real-time kernel and the physical i/o drivers.

Section 4: Results

4.1 Vehicles Automated

The navigator has been configured to run successfully on two very different types of vehicles:

- HMMWV (Highly Mobile Multi-Wheeled Vehicle) - a converted army jeep.
- RATLER (Robotic All-Terrain Lunar Exploration Rover) - a planetary rover. [50]

4.2 Test Run Excursion and Speed

The navigator has been used with laser range data on the HMMWV and with stereo range data on both vehicles. In the former case, excursions of 15 kilometers and instantaneous speeds of 15 km/hr have been achieved while tracking an externally supplied path that was specified at roughly 10 meter resolution. Average speed was on the order of 7 km/hr.

On the RATLER vehicle, the navigator is configured to run under human supervision. In this case, semi-autonomous excursions on the order of 1 Km have been achieved and high speed is not a goal of this work.

The Kalman filter has been implemented in stand alone mode and used as a basis for implementing autonomous vehicle convoys. The sensor suite used included an AHRS for indicating attitude and heading, a transmission encoder and doppler radar both for indicating vehicle speed, a video crosstrack indicator based on a roadfollowing system, and a GPS receiver. In this mode, the filter cycles at 100 Hz and provides sufficient accuracy for blind operation of the follower vehicle over excursions of about 1 kilometer.

4.3 Real-Time Performance

The real-time performance of the system has been demonstrated. With adaptive perception enabled, the overwhelming computational cost is in the tactical feedforward associated with adaptive regard. For laser rangefinder range images, the perception and planning systems can both run at 5 Hz on a single processor. Stereo vision requires significantly more processing.

PART IV: Perception

In Part II: Analysis, Chapter 6: Interactions, it was shown that an adaptive approach to perception based on the techniques of **adaptive sweep** and **adaptive scan** has several advantages. A more complete list of these advantages is as follows:

- It has the potential to solve the **throughput problem** which has tended to increase vehicle reaction time beyond the regimes of usefulness in real-time vehicle control.
- It amounts to a computational solution to the stabilization problem within the limits of the sensor field of view. It essentially converts any imaging sensor into an ideal adaptive line scanner.
- For limited field of view sensors, it provides an obvious basis for the generation of sensor pointing commands which keep a region of interest centered in the image.
- It solves the sampling problem for practical purposes because variation in the range ratio is very low over the small elevation width of the range window.
- It theoretically guarantees throughput, response, and acuity within the limits of the vehicle and the sensor.
- It adapts to vehicle speed, attitude and terrain shape implicitly.

This part concentrates on the adaptive perception algorithm which forms the basis of RANGER'S Map Manager and Perception objects. The techniques used should be applicable to any application that models the environment with a terrain map.

Chapter 1: Concept

An **adaptive perception** algorithm is the organizing principle of the design of the navigator's perception system. It confines the processing of range geometry to a **focus of attention** or **region of interest** which extends to the end of the detection zone - but whose size is governed by the distance moved since the last perception cycle. Thus, this mechanism applies to the image and the perception problem and is based on a throughput argument¹. Adaptive perception has three conceptual parts:

- **adaptive lookahead** - set the maximum range to the end of the detection zone
- **adaptive sweep** - set the region of interest width to the distance travelled since last time
- **adaptive scan** - manage image resolution to achieve uniform groundplane resolution

The motivation for this approach to perception can be expressed in familiar terms. Analysis suggests that much of the computational resources used to image and interpret the environment is a waste of resources in mobility scenarios. This waste occurs for two principle reasons:

1. See Part V: Planning/Control, Chapter 1: Concept for a definition of adaptive regard that distinguishes it from adaptive perception.

- The sensor vertical field of view is aligned with the direction of travel so that image sequences normally contain redundant information. Obstacles appear in the field of view long before they can be resolved, and long after they cannot be avoided.
- The projection of image pixels on the groundplane is elongated in the wrong direction for robust obstacle detection and minimum throughput.

This approach to perception has related precedents in other fields. Laser and video line scanners have been used in specialized applications for a long time. Synthetic aperture radar has used vehicle motion to provide the scanning motion of the sensor along the direction of travel. In a phrase, the essential principle involved is *why scan the sensor when the vehicle motion scans the field of view over the terrain anyway.*

The extreme instantiation of this approach would be to actually use line scanned sensors, but this approach suffers on rough terrain due to abrupt attitude changes of the vehicle body causing holes in the coverage of the sensor. Software adaptation provides the best of both worlds because it gives the ideally focussed attention necessary for high speed and the wide field of view necessary for rough terrain.

Section 1: Adaptive Lookahead

The design of **adaptive lookahead** is a straightforward matter. There is some maximum range beyond which it is unnecessary to look because there will be time to look there later. In detail implementation, the responsibility for computing this maximum range is left to adaptive regard and the results of this computation are used to bound the far end of the focus of perceptual attention.

Section 2: Adaptive Sweep

The design of **adaptive sweep** is also relatively straightforward, once a key assumption is identified, but its implications are more involved.

2.1 Selection Problem

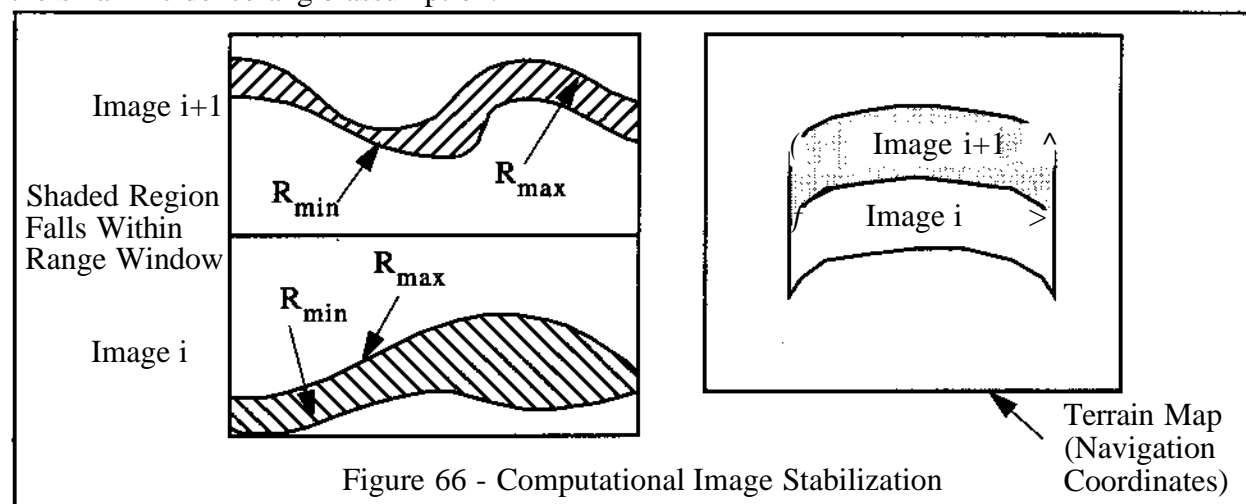
Removal of redundant computation is apparently not a straightforward problem to solve. The key issue is that the position of the end of a range pixel in world coordinates is unknown until it is computed, and the computation of its location is the largest element of the computational expense of a pixel. Any straightforward attempt to selectively process data in an area of interest apparently falters because *the problem of selection is as difficult as the problem of perception.*

Luckily, for high-speed autonomy, the **small incidence angle assumption** solves the problem of selection inexpensively by decoupling the problems of selection from perception.

2.2 Computational Range Image Stabilization

The central idea of adaptive perception is to specify a focus of attention in terms of the data in the image itself. That is, the focus of attention is specified as a **range window** - some maximum and minimum range within which the completely processed data must lie.

This simple mechanism computationally points the sensor vertical field of view based on the guaranteed response and guaranteed throughput requirements as shown in the figure below. Notice that while the shape of the range window may be very irregular in image space, it corresponds to a regular semi-annulus in the navigation coordinate system. This is directly due to the validity of the small incidence angle assumption.



2.2.1 Computational Cost

The processing of pixels outside the range window is confined to simply reading their values and comparing them to the range window interval. The cost of *finding* the range window is negligible. The cost of processing the range window is several coordinate transformations per pixel which was estimated in Part II: Analysis, Chapter 3: Throughput, at about 50 flops in theory and 500 in practice.

2.2.2 Physical Stabilization

If the vertical field of view is wide enough and the range sensor is fast enough in terms of range pixel rate, this software adaptation is superior to the technique of physically stabilizing the sensor because it responds instantaneously to both changes in the vehicle attitude and changes in the shape of the imaged terrain.

2.2.3 Monotone Range Assumption

The 2-1/2 D environment assumption that is involved in the use of a simple terrain map leads to an equivalent **monotone range assumption** - the assumption that range increases monotonically with image elevation angle. The validity of this assumption in the target environment makes the range window mechanism robust in practice.

2.2.4 Vertical Surfaces

The quest for the end of the range window will automatically process all data up to the top of the image if necessary and, as a side effect, it will discover the height of a near vertical surface as long as it remains within the range window². This mechanism is superior to simply processing a fixed subset of the vertical field of view (an **elevation angle** window).

2. Of course, it will also take more time to process a vertical surface, and this time is not reflected in any of the theoretical analysis of the thesis. Thanks to Martial Hebert for pointing this out.

2.3 Computational Stereo Range Image Stabilization

Adaptive sweep is also possible for stereo ranging systems. The basic principle of the range window can be converted to a **disparity window**³ for a stereo system because the two are related by the stereo baseline.

Traditionally, the stereo problem is cast as one of determining the range for every pixel in the image. Traditional stereo finds the range for each possible angular pixel position. Adaptive stereo finds the angular positions of each possible range value. An adaptive approach determines those pixels whose range value falls within a small range window, and it does so without computing the ranges of pixels which are not of interest. This principle is called **range gating** in laser rangefinders which employ it.

The motivation for the approach is that traditional mounting of sensors high on the vehicle, coupled with the nonlinear mapping of range onto disparity, implies that the region of terrain which is beyond the vehicle stopping distance corresponds to a very narrow region in stereo disparity space.

In a sense, the argument is that poor range resolution must be accepted anyway, so we might as well derive benefit from the reduction in computation that should go along with it. Analysis suggests that useful throughput improvements may approach two orders of magnitude.

Section 3: Adaptive Scan

The design of **adaptive scan** is a straightforward matter. Over the width of the range window, the aspect ratio of a range pixel typically varies little - again because of the small incidence angle assumption. Therefore, for the data of interest, a constant mapping from world model resolution to image resolution can be assumed and images can be subsampled by appropriate factors to achieve that resolution.

In the case of rangefinder images, a literal subsampling takes place. In the case of stereo vision, the situation is more complex because range resolution and angular resolution are coupled. Later in this part the implications of the use of a horizontal and a vertical baseline will be investigated in order to make efficient adaptive scan possible in stereo vision.

3. There is a slight difference in the geometry of a stereo range image compared to a rangefinder image. The first is based on perspective geometry and the second is based on spherical polar geometry. Therefore, a disparity window corresponds to a window on the y coordinate and not the true polar range. However, in most circumstances, this distinction can be safely ignored.

Chapter 2: Design

Section 1: Adaptive Perception From Range Images

In the case of laser rangefinders, a range image is generated in hardware. For this class of sensors, adaptive perception is limited to extraction of the range window from the complete image. The **adaptive perception** algorithm constitutes a simultaneous implementation of **adaptive lookahead**, **adaptive sweep**, and **adaptive scan** in one place. It proceeds in two phases: the computation of the range window based on the current speed and cycle time; and the mapping of this window into image space and the extraction of the data.

1.1 Range Window Computation - First Phase

Adaptive lookahead is implemented by computing the distance required to execute an impulse turn⁴ at the current speed. This gives the maximum range of the range window. Adaptive sweep is implemented by computing the distance travelled based on the current and last cycle distance travelled and subtracting this distance from the maximum range. This adaptive sweep algorithm will automatically adapt to any system cycle time, sensor frame rate, or vehicle speed - using smaller sweeps for faster sensors. The **planning window** is defined precisely to be measured with respect to the vehicle control point at the position of the vehicle when the command currently being computed reaches the output communication routines. The problem of finding this data in an image taken previously involves several aspects of time delays and geometric offsets.

First, the sensor is not mounted at the vehicle control point, so the planning window is adjusted for the offset of the sensor in order to project the window into sensor coordinates. Second, the vehicle is not itself a point, so if geometry under the front wheels must be known to evaluate hazards, then the vehicle wheelbase must be added to the range window. Third, there may be significant delay associated with the acquisition of an image, so the range measurements are adjusted for the age of the image. The most robust way to compute this age is to find the vehicle pose when the image was taken, as well as the most recent vehicle pose, and compute the distance travelled since image acquisition. Fourth, the most recent vehicle state estimate is itself somewhat old and it will take some time before the command computed this cycle reaches the output communications routines.

4. An impulse turn was chosen as the maneuver upon which to base adaptive lookahead because a vehicle which stops when it sees an obstacle (panic stop maneuver) is not useful when obstacles are dense. The impulse turn is a turn from zero curvature to the maximum. Other distinguished maneuvers are the panic stop, the turning stop, and the reverse turn.

So the distance that will be travelled by the time that the command reaches the output communications routines is dependent on the age of the state estimate and the current cycle time of the system. A conceptual C code fragment is shown below:

```

/*
** Plan Window
*/
rhomin = minimum safe turn radius
treact = impulse turn reaction time (turn only)
speed = most recent speed estimate
range_offset = distance from vehicle frame to sensor frame
Pmax = speed * treact + rhomin; /* adaptive lookahead */
/*
** Range Window
*/
img_dens = imaging density
cycle_dist = distance travelled since last cycle
Pmin = Pmax - img_dens*cycle_dist; /* adaptive sweep */
/*
** Convert to image space - range feedbackward
*/
cur_dist = distance of most recent state estimate
img_dist = distance travelled when image was taken
delay = estimated latency of most recent state estimate
delay += system cycle time
fut_dist = cur_dist + speed*delay;
Rmax = Pmax + (fut_dist-img_dist)+range_offset;
Rmin = Pmin + (fut_dist-img_dist)+range_offset;
/*
** Add the wheelbase
*/
Rmax += wheelbase

```

Figure 67 - Range Window Computation

1.2 Range Window Processing - Second Phase

The second phase of adaptive perception will be presented in two passes of increasing complexity. First, a simple implementation is presented that ignores the issue of adaptive scan.

1.2.1 Adaptive Sweep

By the **small incidence angle assumption**, the projection of the sensor range onto the groundplane is essentially the groundplane y coordinate. However, terrain roughness and nonzero vehicle roll mean that the position of the range window in the image is *different for each column*. Thus the range window is processed on a per column basis. In order to robustly find the range window, each column is processed in the bottom-to-top direction. The direction of horizontal scan is irrelevant.

A conceptual code fragment is as follows:

```

/*
** Process Range Window
*/
j = image->start_col;
while ( j < image->end_col )
{
    i = image->end_row;
    while ( i > image->start_row )
    {
        if (range(i,j) > Rmax ) break;
        else if( range(i,j) < Rmin ) {i--; continue;}
        else process_pixel_into_map();
        i--;
    }
    j++;
}

```

Figure 68 - Adaptive Sweep Implementation

The **monotone range assumption** appears as the break statement after the first conditional of the inner loop. The start_col and end_col variables implement a fixed azimuth and elevation angle window within which the range window always lies on typical terrain.

1.2.2 Adaptive Sweep/Scan

Now, a more detailed version would incorporate adaptive scan by intelligently subsampling the image in both directions in order to keep the pixel density on the groundplane uniform. This can be accomplished by invoking the **flat world assumption** as an approximation. From earlier analysis, the projections of a range pixel onto the groundplane in the crossrange and downrange directions are:

$$dy = \left(R^2/h \right) d\theta$$

$$dx = R d\psi$$

In the simplest form of adaptive scan, the number of pixels skipped in the horizontal and vertical directions can be set in the ratio of the average expected value of the perception ratio h/R at the maximum range. However, the differential relationships are also a function of range. Consider attempting to keep dx and dy constant by varying the column step dψ and row step dθ. The actual width of the range window is narrow in elevation angle, so the variation in ideal column step is small. Appropriate density is guaranteed by computing the column step at the maximum range because it is smallest here. Thus

$$d\psi = \left(\frac{1}{R_{max}} \right) dx$$

is the ideal column step after it is converted to pixels. The ideal row step varies greatly with range. It decreases quadratically with increased range.

$$d\theta = \left(\frac{h}{R^2} \right) dy$$

Luckily, however, it can be computed anew for each pixel because the range window is processed column by column. Although the flat world assumption may seem inappropriate on rough terrain, the use of it in adaptive scan works well in practice. Fundamentally, the most important factor affecting range pixel density is the local terrain gradient. If the range gradient is high in the image in the horizontal or vertical direction, then it is probably the case that the terrain is undersampled by the sensor, and the flat world assumption makes little difference because the data is of poor quality. Indeed, it may be more robust to ignore such data. If however, the gradient is small, the algorithm performs acceptably. A conceptual code fragment is given below:

```
/*
** Compute scan increments
*/
ds = map_resolution
col_skip = ds/Rmax*image->cols/HFOV;
/*
** Process Range Window
*/
j = image->start_col;
while ( j <= image->end_col + col_skip)
{
    i = image->end_row;
    while ( i >= image->start_row - row_skip)
    {
        if (range(i,j) > Rmax ) break;
        row_skip = ds*h/(range(i,j)*range(i,j))*image->rows/VFOV;
        else if( range(i,j) < Rmin ) {i -= row_skip; continue;}
        else process_pixel_into_map();
        i -= row_skip;
    }
    j += col_skip;
}
```

Figure 69 - Adaptive Sweep/Scan Implementation

One of the virtues of the algorithm is that it intelligently moves toward the bottom of the range window by subsampling in the region of the image where the data density is extremely high. In actual implementation, oversampling factors are introduced into both the row and column skip factors as a safety margin.

Section 2: Adaptive Stereo Perception

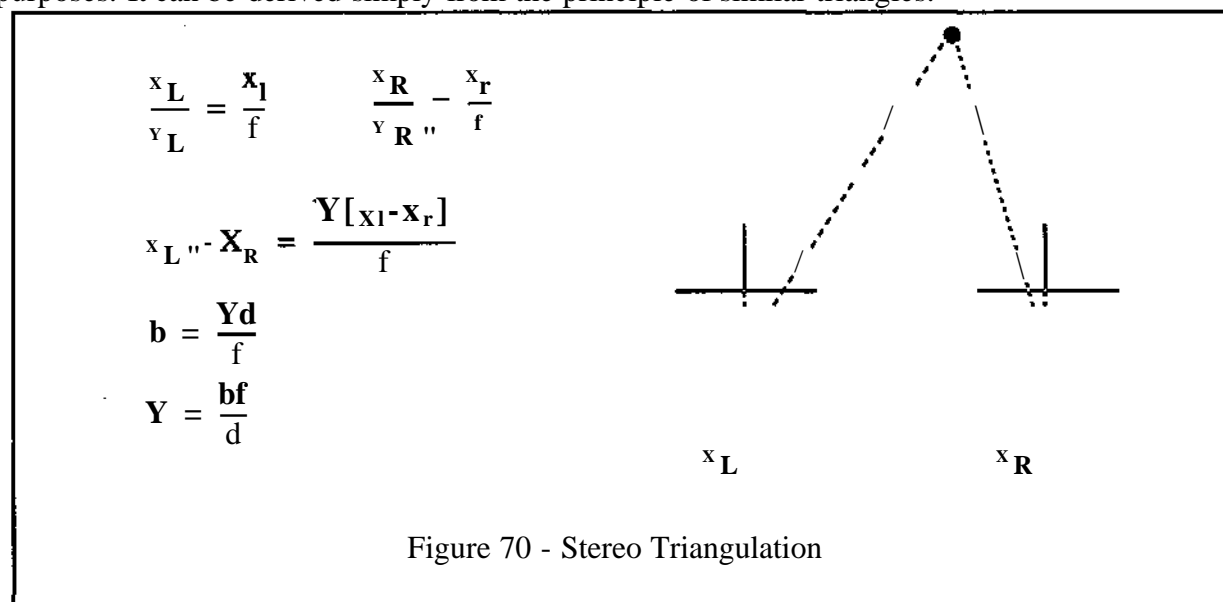
2.1 Basics of Stereo Perception

For outdoor terrain, area-based stereo algorithms are typically used because it is necessary to estimate the range of every pixel in the image. The typical steps in the process are:

- Preprocess the images to enhance texture and remove bias and scale variations across the image. The output of this process is a normalized image which corresponds to each input image.
- For each candidate disparity considered, for a window around each pixel in the first image, compute a measure of correlation between it and a window around the pixel in the second image which is displaced by the disparity considered. The output of this process is a cube of numbers of the form $\text{Corr}[i,j,d]$ which will be called the **correlation tensor**.
- The curve $\text{Corr}[d]$ obtained by fixing the row and column indices of the correlation tensor will be called the **correlation curve**. For each pixel in the first image, the correlation curve is searched to find its extremum value. The value of the disparity at the extremum value of the correlation curve for that pixel is the quantity of interest. The output of this process is a **disparity image**.
- For each pixel in the disparity image, convert disparity to range using the stereo baseline. The output of this process is the **range image**.

2.2 Stereo Triangulation

The basic stereo triangulation formula for perfectly aligned cameras is quoted below for reference purposes. It can be derived simply from the principle of similar triangles.



2.3 Width of Disparity Window

Using the previous result, consider the width of the disparity window which corresponds to a typical range window. It is most useful to remove the dependence on the focal length by expressing disparity as an angle, the **normalized disparity**, thus:

$$\delta = \frac{d}{f} = \frac{b}{Y}$$

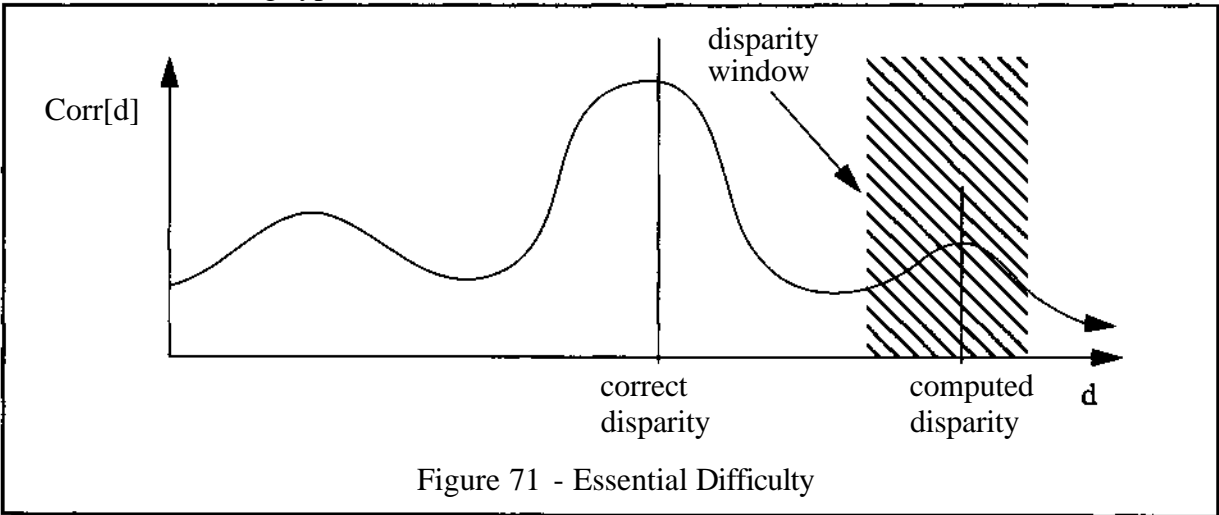
Then, for a range window between 25 meters and 30 meters, and a stereo baseline of 1 meter, the angular width of disparity window is:

$$\Delta\delta = \frac{1}{30} - \frac{1}{25} = 0.0067 = 0.38^\circ$$

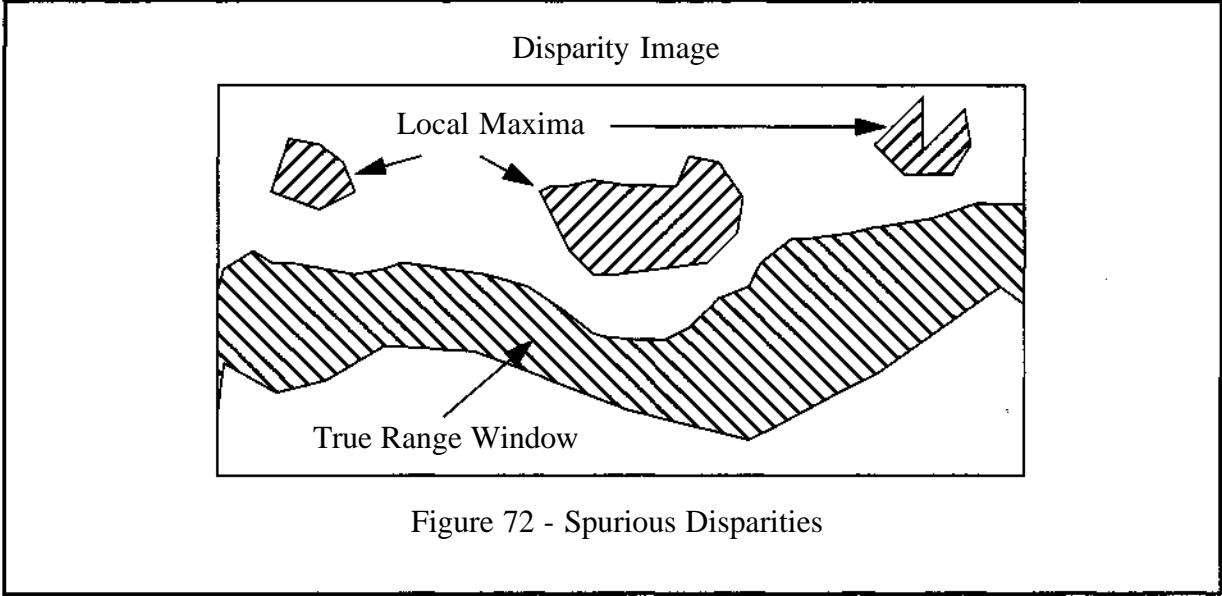
Thus, the range of disparities which corresponds to a typical range window is very small indeed. This implies that any process which robustly identifies global maxima of the disparity curve can generate the range window in an image. Again, because of the validity of the **small incidence angle assumption**, and because the window is based on the data itself, the process will automatically adapt for rough terrain and vehicle attitude. Such a process implements **adaptive sweep** inside the stereo algorithm.

2.4 Essential Difficulty

Consider the following typical correlation curve:



If the search for the maximum is confined only to the disparity window, then a local maximum will be found for the pixel, and not the correct global maximum. This will be called the **local correlation maximum problem**. This implies that a typical image will contain pixels and regions where the ranging is incorrect as well as the correct range window as shown below:



Spurious matches occur fundamentally because *regions which do not correspond physically actually look more or less the same*. This is called the **repetitive texture problem**. Several solutions to the repetitive texture problem help the situation somewhat but the simple technique of computing connected components and removing small regions [59] works effectively and is basically free because a disparity image cleanup pass is required even when a wide disparity range is searched.

Section 3: Refinements to Correlation-Based Stereo

This section introduces some refinements to correlation-based stereo vision that adapt the traditional algorithm to the specifics of the application.

3.1 Optimal Shape of the Correlation Window

An earlier section derived the geometric decorrelation for a correlation window of constant disparity gradient:

$$C_{\text{err}} = \frac{\delta_u U^2 V}{4} + \frac{\delta_v V^2 U}{4}$$

Notice that the decorrelation depends only on the disparity gradient and the window dimensions.

This expression can be optimized if another constraint equation is available. Let the area of the correlation window remain constant while the width and height vary dependently. Then, setting the derivative to zero, there results:

$$\frac{U}{V} = \frac{\delta_v}{\delta_u}$$

Thus *the optimum aspect ratio of the correlation window is the inverse of the ratio of the corresponding components of the disparity gradient*⁵. This is intuitively appealing because it requires that the extent of the window be minimized in the direction where disparity changes most. This result explains why outdoor stereo systems tend to perform better when the correlation window is wider than it is high. On average, the disparity gradient is mostly vertical in the image plane in outdoor settings.

3.2 Complexity of Stereo Vision

It is a straightforward matter to compute the complexity of stereo vision when implemented on a serial workstation by computing the complexities of the component processes⁶. A few assumptions will be made to simplify matters.

3.2.1 Assumptions

These assumptions do not affect the result - they only simplify the analysis:

- Scaling and rectification are performed as a single pass operation as they currently are in the navigator. This requires that scaling be done by subsampling, not filtering, and that rectification be implemented by inverting the rectification matrices.
- Let the image consist of R rows and C columns and let the angular resolution be precisely adequate to provide a pixel footprint satisfying the acuity requirement.
- Let the cost of multiplying a vector by a homogeneous 3 X 3 matrix be K₁ flops per

5. Bill Ross of VASC at CMU related this observation to me verbally several years ago. The above analysis simply proves why it is so.

6. The component processes are explained in the next chapter on implementation.

pixel.

- Let a moving average convolution operator cost K_2 flops per pixel.
- Let the normalization and correlation windows be the same size and call this size W columns by H rows.
- Let the number of disparities searched be a constant interval from 0 to some maximum value of D .

3.2.2 Complexity of Traditional Stereo Vision

The total complexity can be developed as follows:

- Scale and Rectify. This operation consists of a multiplication by a matrix and hence, for two images, it is $O(2K_1RC)$.
- Normalization. This operation is independent of the window size and can be implemented with a moving average operator, so, for two images, it is $O(2K_2RC)$.
- Correlation. This operation is also independent of window size and can be implemented with a moving average operator. Correlation must be performed for each value of disparity, so, for two images it is $O(K_3RCD)$.
- Disparity. This operation involves searching the disparity curve, containing D correlation scores, for every pixel. It is therefore $O(K_4RCD)$.
- Cleanup. This operation involves checking connectedness of every horizontal and vertical pair of pixels with some extra processing for relabelling that is not of substantial complexity, so this operation is (XK^4RC) .
- Triangulation. This operation consists of converting disparity to range with a simple formula and then multiplying a vector by a matrix, so it is (XK^4RC) .

The total complexity of traditional nonadaptive stereo vision is therefore (on a per frame basis):

$$f_{\text{stereo}} = (2K_1 + 2K_2 + K_5 + K_6)RC + (K_3 + K_4)RCD$$

3.2.3 Complexity of Partial Subsampling

Adaptive scan is somewhat problematic in stereo because the high angular resolution that is so computationally expensive is necessary to provide the texture necessary for accurate triangulation. Apparently, adaptive scan can be simply implemented in stereo by skipping columns in the input images. Note, however, that this technique can fail in practice unless one is careful because disparity resolution is unacceptably compromised.

It is possible to subsample the disparity image but the computations which generate the disparity image (except disparity itself) normally must be performed at high resolution. Adaptive scan can be implemented in the latter stages of stereo including the latter stages of correlation, disparity and triangulation. This technique will be called **partial subsampling** [48]. Supposing that the horizontal angular resolution is adjusted by a factor of the **perception ratio** relative to the vertical, the complexity in this case is:

$$f_{\text{stereo}} = \left(2K_1 + 2K_2 + \frac{h}{R}K_5 + \frac{h}{R}K_6 \right)RC + \left(K_3 + \frac{h}{R}K_4 \right)RCD$$

If one is willing to accept the complexity of implementation, one can subsample the correlation window in one of the images to further reduce processing⁷. The complete set of steps involved in this more efficient approach are:

Subsample one of the images in the normalization and rectification step. Then:

$$2 (K_x + K_2) \quad \text{becomes} \quad I \left(1 + \frac{h}{R} \right) (K_1 + K_2)$$

Subsample the correlation window and the correlation tensor. Then:

$$K_3 + \frac{h}{R} K_4 \quad \text{becomes} \quad \frac{h}{R} (K_3 + K_4)$$

The final complexity then becomes:

$$f_{\text{stereo}} = \left[\left(I + \frac{h}{R} \right) \langle K_1 + K_2 \rangle + \left[K_5 + \frac{h}{R} J \right]^{RC} + \langle K_3 + K_4 \rangle^{RCD} \right]$$

$$f_{\text{stereo}} = \left(\left(I + \frac{h}{R} \right) \langle K_1 + K_2 \rangle^{RC} + \left[\langle 2K_1 + 2K_2 + K_5 + K_6 \rangle^{RC} + (K_3 + K_4) RCD \right] \right)$$

3.2.4 Complexity of Complete Subsampling

Suppose there were a way to subsample the input images by the factor of the perception ratio. In this case, the overall complexity of stereo would be reduced to $\frac{h}{R}$ of its traditional value because the common factor of C in every term would be multiplied by this factor. This technique will be called **complete subsampling**. Its complexity is trivially:

$$f_{\text{stereo}} = \left[\langle 2K_1 + 2K_2 + K_5 + K_6 \rangle^{RC} + (K_3 + K_4) RCD \right]$$

Notice that this is less than the horizontal partially subsampled complextiy by the missing first term:

$$\left(I - \frac{h}{R} \right) (K_1 + K_2) RC$$

It will be shown that the use of a vertical baseline permits this form of complete subsampling.

7. Thanks to Larry Mathies for pointing this out.

3.3 Adaptive Scan in Stereo Vision

The techniques of partial and complete subsampling implement adaptive scan in stereo vision. This section investigates the differences between both approaches. The **perception ratio** is on the order of **1/10**, so *adaptive scan stereo vision can be an order of magnitude faster than traditional stereo* because the raw input images can be reduced in horizontal resolution by an order of magnitude.

3.3.1 Equivalent Distortion

Earlier sections have introduced a quantity called the **geometric decorrelation** as a measure of the local distortions of corresponding regions that is inherent to stereo vision. This quantity is dependent only on the disparity gradient over the correlation window and therefore only on the **scene geometry**. Hence, *the local distortions of horizontal and vertical baseline stereo systems are identical and distortion is not a material concern in choosing between the two*.

3.3.2 Resolution Coupling

Fundamentally, the difficulty of implementing adaptive scan in stereo vision arises from the relationship, intrinsic to triangulation, between angular resolution and range resolution. To see this, consider again the fundamental acuity relationships for a non-triangulated rangefinder. Recall the usual axis conventions where x points to the right, y forward, and z up. Then for horizontal and vertical pixels of sizes $d\psi$ and $d\theta$ respectively (see page 70):

$$dx = R d\psi$$

$$dz = R d\theta$$

$$dy = \frac{R^2}{b} d\theta$$

Solving these for the angular resolutions:

$$d\psi = \frac{dx}{R}$$

$$d\theta = \frac{dz}{R}$$

$$d\theta = \frac{b}{R^2} dy$$

In stereo vision, the range resolution relationships for horizontal and vertical baselines are as follows (see page 81) where the approximation $R \ll Y$ has been used:

$$dy = \frac{b}{R} d\psi \quad (\text{horizontal baseline})$$

$$dy = \frac{b}{R} d\theta \quad (\text{vertical baseline})$$

Solving these for the angular resolutions:

$$d\psi = \frac{R}{b} dy \quad (\text{horizontal baseline})$$

$$d\theta = \frac{R}{b} dy \quad (\text{vertical baseline})$$

3.3.3 Triangulation

These equations can be used to detennine angular resolution by fixing the spatial resolutions (dx, dy, dz) and solving for the angular resolutions (dψ, dθ) . The minimum value of each resolution variable is the requirement on the imaging system. Let all three spatial resolutions be equal (dx = dy = dz) . Then, assuming that the baseline is less than the height, the most constraining equations for both stereo pair orientations become:

$$d\psi = \frac{b}{R^2} dy$$

(horizontal baseline)

$$d\theta = \frac{z}{R^2} dy$$

$$d\psi = \frac{dy}{R}$$

(vertical baseline)

$$d\theta = \frac{1}{R} dy$$

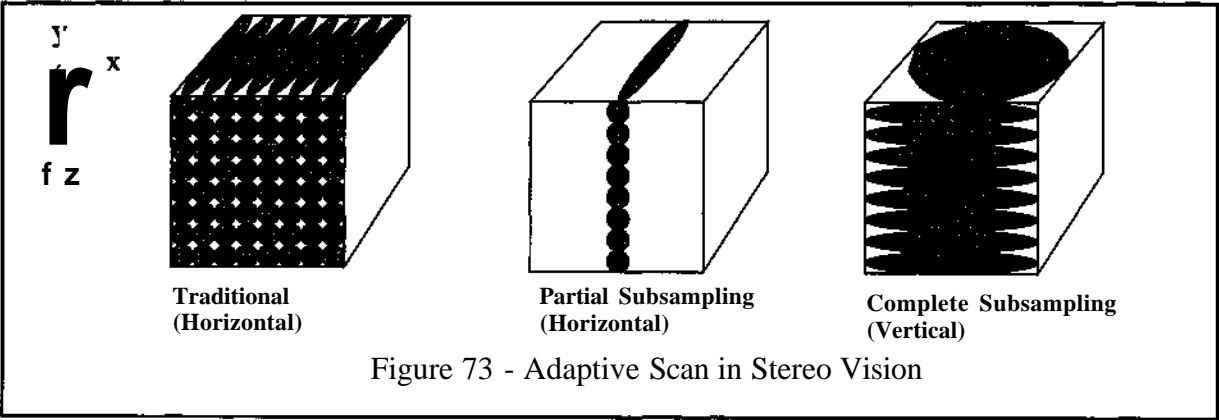
The reason for this asymmetry is that angular resolution requirements are already higher in the elevation direction for any type of sensor in high-speed autonomy.

3.3.4 Visualization

The points made above in algebraic terms can be easily seen with a picture. Let a cube exist in the environment whose dimensions are exactly the smallest dimension of interest (**minimum acuity**). Let the image pixels be such that they project onto precisely this dimension in the downrange direction. In the case of horizontally triangulated traditional stereo, the left figure results for a perception ratio of 8 and 72 range pixels fall on the cube.

The central point being made is that it is not possible to widen the image pixels in the x direction on the front surface because this also widens the pixels on the top surface of the cube in the y direction. In the horizontally triangulated adaptive scan case, the cube geometry is subsampled in order to reduce computation, and 9 pixels fall on the cube. In the vertically triangulated adaptive scan case, it is the z direction which is coupled to y, so it is possible to widen the pixels on the front surface and again 9 pixels fall on the cube. Note that:

- In all cases the minimum acuity requirement is barely met.
- The ratio of numbers of pixels from traditional to subsampled is precisely the inverse perception ratio of 8.



3.3.5 Advantages of Vertical Baseline

Vertical baseline stereo has several advantages. Some of these are:

- Complete subsampling is more straightforward to implement than partial subsampling, because this allows the requirement on the horizontal resolution to be relaxed at the input to the stereo algorithm. One simply runs the traditional stereo algorithm on rotated and scaled input images.
- It does not reduce the horizontal field of view (which is often already too small) but instead reduces the vertical field of view (which is often larger than necessary).
- It promises to permit extremely wide field of view stereo from a single camera pair because the distortions introduced by a wide field of horizontal view do not matter much when triangulating vertically.
- It permits a straightforward operation to remove the distortions due to the average expected disparity gradient. The vertical dimension of the bottom image is simply scaled spatially by the inverse of the disparity gradient.

3.4 Adaptive Sweep

A full implementation of adaptive perception requires that adaptive sweep be introduced. The implementation of adaptive sweep is based on the disparity window presented earlier as well as an associated computation of the azimuth window in the image to which the disparity window is likely to correspond. From the transformations provided in the appendices, the closed form stereo range image of flat terrain is:

$$Y = \frac{h}{\tan(\theta + \beta)}$$

where Y is the range measured normal to the image plane, θ is the pixel depression angle (positive downwards) and β is the tilt (pitch) angle of the camera (positive downwards). Clearly this relationship can be solved for the pixel depression of a given range. Hence, for a given range window, we have:

$$\begin{aligned}\theta_{\min} &= \operatorname{atan}\left(\frac{h}{Y_{\max}}\right) - \beta \\ \theta_{\max} &= \operatorname{atan}\left(\frac{h}{Y_{\min}}\right) - \beta\end{aligned}$$

In order to account for the fact that the flat terrain assumption is not entirely correct, this window is enlarged by an appropriate amount in either direction. The disparity window is computed in straightforward manner from the basic triangulation equation.

Section 4: Terrain Mapping

This section describes the terrain map data structure which is used to model the environment in rough terrain. Terrain mapping is the process by which surface descriptions, obtained from different vantage points, are accumulated into a consistent environmental model [31]. In order to provide the path planner with a single, coherent, uniform density data structure that may span several images, the approach taken is to transform images into a regularly-spaced cartesian grid called a Cartesian Elevation Map (CEM) or simply a **map**. In the map, each cell encodes z_{ij} where the z coordinate is unique for any pair i,j and is referenced to some fixed coordinate system called the navigation coordinate system with respect to which the vehicle moves. Individual elevation buckets in a terrain map are called **cells** to distinguish them from range image **pixels**.

Under some circumstances, natural outdoor terrain is well approximated by a surface expressed as $z = f(x,y)$ where the z axis is aligned with the local gravity vector. An important exception to this assumption is trees and other large vegetation, but barren terrain is assumed in this work in order to avoid the need to differentiate vegetation that can be traversed from vegetation that constitutes a hazard. Thus, the use of a terrain map normally means that the **2-1/2 D world assumption** is being adopted.

4.1 Wrappable Map

Consider the implications of satisfaction of guaranteed response and acuity on the memory and computation required to manage a traditional terrain map. Up to 30 meters of lookahead is not uncommon and resolutions on the order of 1/6 meter are theoretically necessary. Using only 20 bytes of memory per map cell, over 1/2 megabyte of memory is required to store a typical map. If this map is stored as a physically coherent block of memory, it must be physically shifted and copied after the acquisition of each image in order to account for the relative motion between the vehicle and the terrain. Using a realistic frame rate necessary to guarantee detection, the overhead involved in simply storing and managing the environmental model is not justified in a real-time system.

The solution to this problem is a classical one from computer science - the ring buffer. However, there is no intrinsic requirement to insert and delete nodes into a linked structure because the map is spatially coherent and of uniform resolution. Hence, a simple array accessed with modulo arithmetic suffices to *logically scroll* the map as the vehicle moves.

The operating principle of the wrappable terrain map is that the indices into the array are determined by modulo arithmetic as follows:

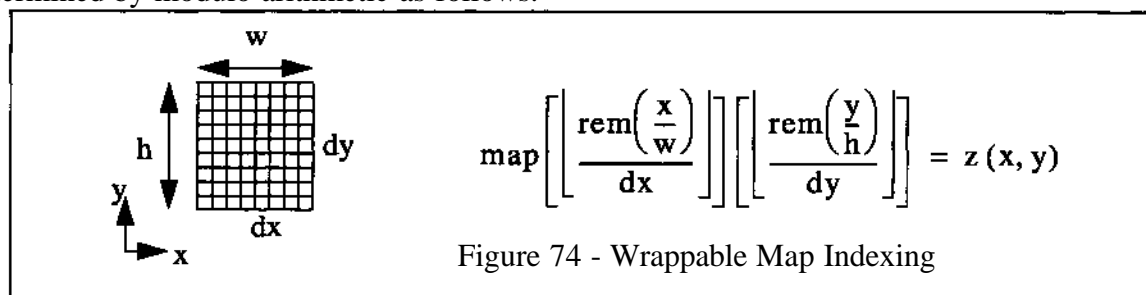
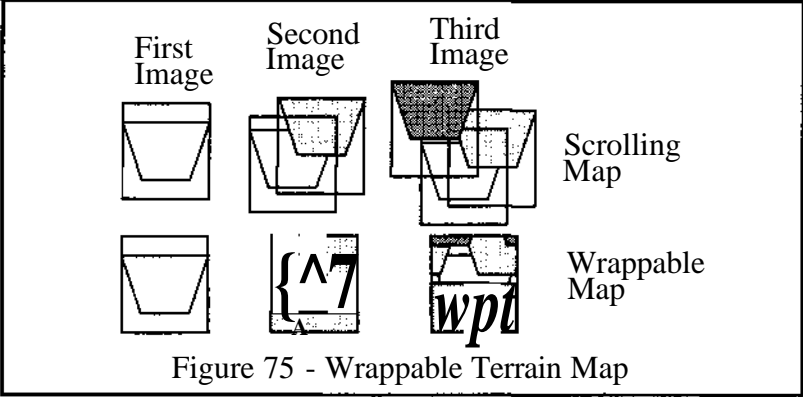


Figure 74 - Wrappable Map Indexing

The operator $[x]$ is the least integer function and $\text{rem}(x/y)$ is the floating point remainder function. These are implemented more efficiently than a function call in the code⁸. The operation of the technique when applied to three successive images is indicated below.



This approach creates new problems, the most serious of which is the condition that the mapping from world coordinates to map indices is multiply defined and therefore the inverse mapping is not a function. In mathematical terms, the coordinate transform is not **onto**.

An infinity of points in global coordinates correspond to a single cell in the map, so remnants of images of arbitrary age may remain in the map indefinitely. Suppose the elevation at the point (15, 25) is needed and the map is 10 by 10. Then the point (5,15) may also be in the map. A query for the elevation at (15, 25) may get the elevation at (5, 15) instead.

The system manages this problem in a very simple way. Although all data remains in the map until it is overwritten, each entry is tagged with the distance that the vehicle had travelled since the start of the mission at the time the pixel was measured. If the "age" of the last update is too old, the cell is considered to be empty⁹. This technique eliminates all of the overhead of map management except for the coordinate transformation necessary to access it. If pixels older than the length of the map are discarded in this way, it is impossible for old data to poke through the holes in new data.

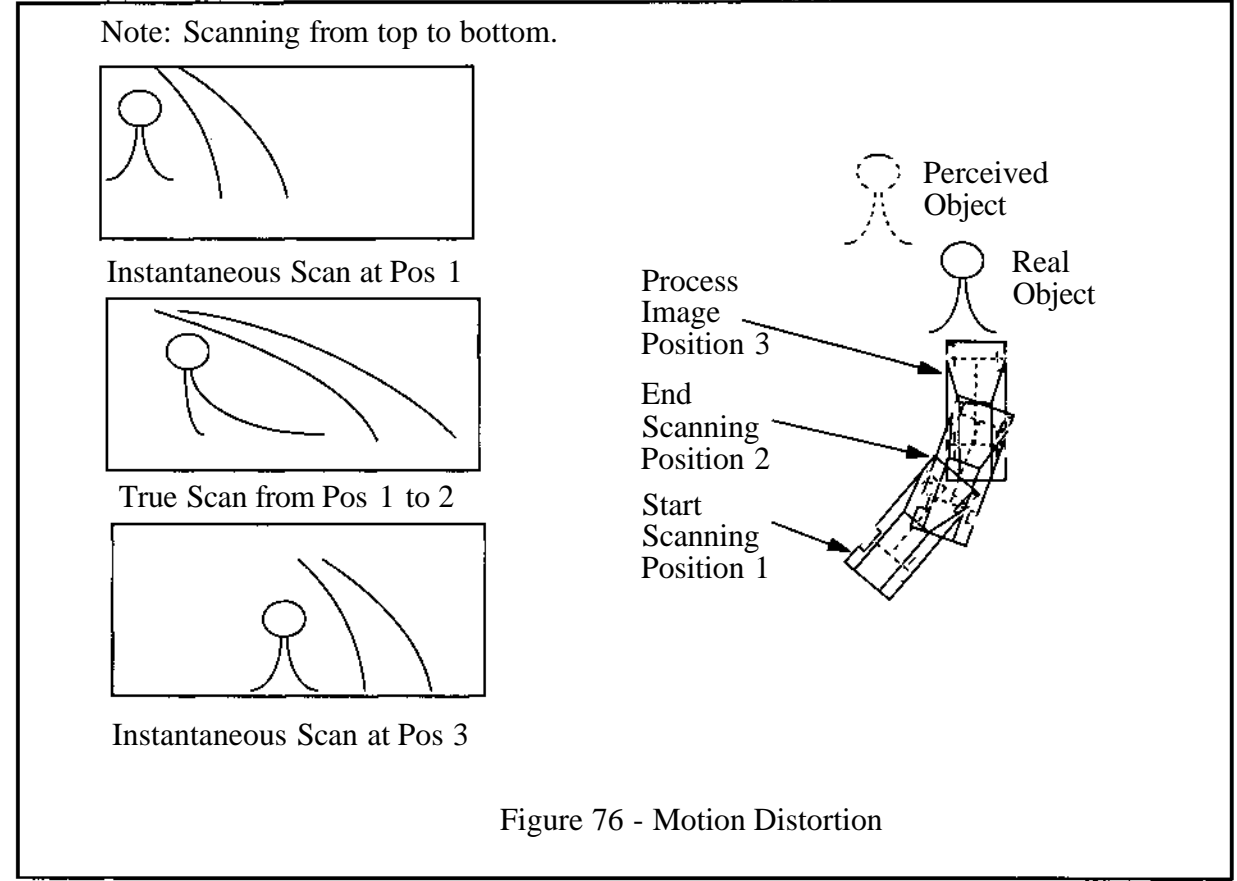
The impact of this data structure is that it is never necessary to perform a copy of the map data structure, or completely traverse it for any reason. Theoretically, the complexity of map management is squared in the resolution unless the map is wrappable. The cost of management of this data structure is independent of both its size and its resolution.

4.2 Motion Distortion Removal

By the time an image is received by the perception system, the vehicle may have moved a considerable distance since the image was acquired. So, the processing of the geometry in the image must account for the exact position of the vehicle when the image was taken. Further, some sensors such as scanning laser rangefinders may require significant time to scan the laser beam over the environment. In the worst case, there is a distinct vehicle pose associated with each pixel in a ladar image. If this motion distortion is not accounted for, the terrain maps computed from images will be grossly in error.

8. By assuming excursions are limited to a few times the distance to the moon!
 9. Age is measured by distance and not time because otherwise the geometry under the wheels will eventually disappear when the vehicle stops.

The worst case is a high angular velocity turn as indicated in the figure below. If rangefinder scanning takes about 0.5 sees and the vehicle is travelling at 6 mph and turning sharply, its angular velocity can be as high as 1 rad/sec, so an obstacle can be smeared by 30° in a rangefinder image at high speed. Similarly, if the input latency is 0.5 sees and it is not accounted for, objects will also be shifted by 30° in a rangefinder image at high speed. Of course, the range to an object will also be overestimated by the distance travelled in 1 second.



This distortion of range images is removed by maintaining a history of vehicle poses sampled at regular intervals for the last few minutes of execution and searching this list for the precise vehicle position at which each range pixel was measured.

4.3 Delayed Interpolation

The terrain map is not interpolated at all because interpolation requires a complete traversal which is too expensive to perform. Instead, the responsibility for interpolation is left with the users of the map. Spatial interpolation is wasteful because the ultimate use of the map is to evaluate vehicle safety and vehicle safety can be expressed as a time signal.

It is more efficient to *delay interpolation* until the point in the computation at which it is really needed, and this point occurs inside the path planner. Also, only a small portion of the map is actually used in some situations because the vehicle maneuverability is limited. Any interpolation of unused geometry amounts to a waste of resources.

Another aspect of the interpolation problem is that occlusion is inevitable anyway, so spatial interpolation can never succeed fully without unjustified and harmful smoothness assumptions on rough terrain.

Thus the path planner *interpolates in time instead of in space*. Throughout its internal processing, the central data structure is a time signal which may or may not be known at a particular point in time. The system is robust by design to unknown signal values and, as a by-product of its processing, computes an assessment of how much geometry is actually unknown and reacts accordingly. In this way, interpolation and occlusion are treated in a unified way and the system considers too much of either to be hazardous.

4.4 Image Registration

A simple image registration algorithm is used in situations where the imaging density is reduced below the amount necessary to ensure that the geometry under the rear wheels comes from the same image as the front wheels in the feedforward simulation.

The basic mechanism is to compute and remove the average elevation deviation between the overlapping regions of consecutive images. Currently, only the elevation (z) coordinate is matched and this works best in practice due to systematic errors which are, as yet, unidentified. When the z deviation of two consecutive images is computed, it is applied to all incoming geometry samples in order to remove the mismatch error.

4.5 Elevation Uncertainty

After the mean mismatch error is removed, there are still random errors in the elevation data. In order to represent the variation in geometry in a single map cell and to improve signal to noise ratios, a **scatter matrix** is computed [35] incrementally as each new range pixel is merged into the map. The scatter matrix is defined as:

$$S = \begin{bmatrix} X^{xx} & X^{xy} & X^{xz} \\ X^{yx} & X^{yy} & X^{yz} \\ X^{zx} & X^{zy} & X^{zz} \end{bmatrix}$$

The advantage of this incremental approach is that the mean and standard deviation of the evolving 3D distribution is available at any point in time from some simple formulas. Specifically, the deviation in z is useful for computing the uncertainty in the hazard estimates generated by the path planner.

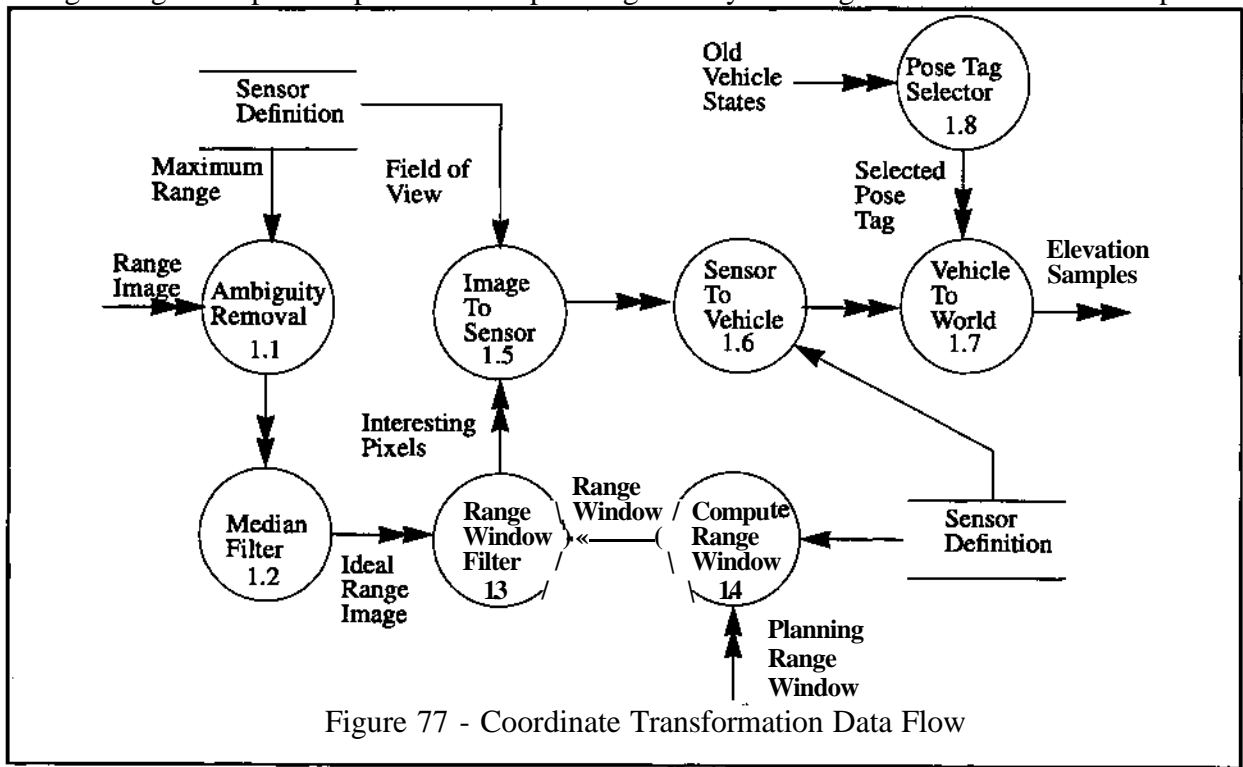
Chapter 3: Implementation

This section overviews the implementation of the perception system. Image generation may be as simple as a hardware function for color, intensity, or laser range images, or may involve nontrivial processing for stereo images. The perception system converts range image sequences into a large number of samples of the geometry of the vehicle's immediate environment specified in the navigation cartesian coordinate system. This subsystem incorporates an image processing element which converts a raw image into an ideal image, an adaptive perception element which determines which region of the image contains new information and therefore warrants the computational cost of coordinate conversion, and finally the coordinate transformation itself.

Section 1: Coordinate Transformation

The coordinate system conversion proceeds from image space all the way to global coordinates. Therefore the vehicle position estimate specified in the vehicle state is required in order to do this transformation. Old vehicle positions are used in the coordinate transformation in order to remove the distortion associated with the delay of the perception and communication electronics.

The data flow diagram for the coordinate transform component of the map manager object is given below. The mathematics of these transforms are documented in the appendices. This section takes a range image as input and produces samples of geometry in navigation coordinates as output.



1.1 Ambiguity Removal

For amplitude modulated sensors, range is available modulo some effective modulator wavelength. This module reconstructs an ideal range image from a range staircase generated by an AM laser rangefinder.

1.2 Median Filter

This module uses the well-known median filter algorithm to reduce image noise without smoothing the entire image as a side effect.

1.3 Range Window Filter

This module implements the adaptive perception algorithm. For each column of the image, the algorithm walks up from the bottom to the top to determine the range pixels which are within the current region of interest, and only those pixels are passed onto later stages in the processing.

1.4 Range Window Computation

This module transforms the planning range window, supplied by the vehicle object, into a range window in sensor coordinates. The transform accounts for the position of the sensor on the vehicle, the latency of the sensor, and the size of the vehicle wheelbase.

1.5 Image to Sensor

Based on the projection model for the sensor, which may be azimuth polar, elevation polar, or perspective, this module converts coordinates from (range, row, col) to (x,y,z) in a sensor-fixed cartesian coordinate system. A lookup table called the projection table is used which specifies the direction cosines of the ray through each pixel for efficiency reasons.

1.6 Sensor to Vehicle

Based on the position and orientation of the sensor frame on the vehicle, this module converts coordinates from the former to the latter via multiplication by the relevant homogeneous transform. If the sensor is mounted on a head, the head transform is incorporated in real time. If not, the transform is fixed and is computed once at start-up.

1.7 Vehicle to World

This module uses the position of the vehicle *when each pixel was measured* in order to convert coordinates into the global frame of reference via the relevant homogeneous transform. Elements of this transform change as the vehicle moves, so it is recomputed each cycle.

1.8 Pose Selector

This module interpolates linearly between old vehicle states based on the known physical scanning pattern of the rangefinder (or a single pose can be used for stereo) and assigns a (potentially unique) pose to every pixel in the image. In this way, both motion distortion and aliasing in the terrain map are avoided. The transforms required to form the range image (such as disparity to range for stereo) are considered to exist outside this part of the system.

Section 2: Map Manager

The Map Manager deals with the transformation of data into and out of the discrete representation of the terrain map. The data flow diagram for this component is given below:.

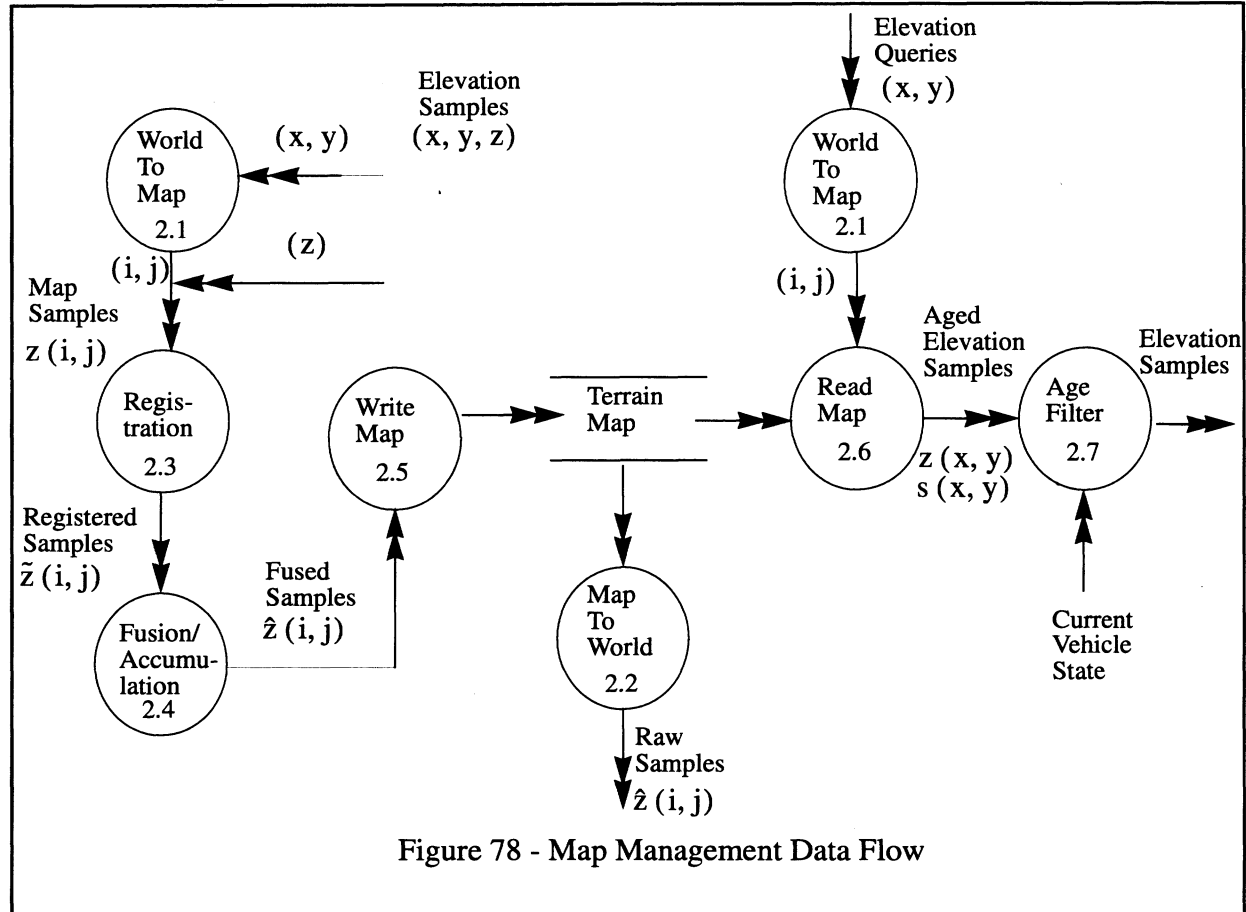


Figure 78 - Map Management Data Flow

2.1 World to Map

This module converts coordinates from (x,y) to (i,j) in order to process terrain samples into a map. A single block of memory is used to store elevation and the (i,j) coordinates of a point are computed by modulo arithmetic on the input (x,y) tuple.

2.2 Map to World

This provides the inverse transformation to the above for debugging purposes.

2.3 Registration

This module recovers the vehicle excursion between images by matching the overlapping regions of consecutive images. Only the elevation (z) coordinate is matched. When the z deviation is computed, it is applied to all incoming samples in order to remove the mismatch error.

2.4 Fusion/Accumulation

This module computes mean and standard deviation statistics for map cells in situations when two or more range image pixels from the same image (or optionally, from consecutive images) fall into the same map cell.

2.5 Write Map

This module performs the physical memory write operation. When pixels are placed in the map, they are tagged with the current distance travelled in order to implement a **cell aging feature** that prevents overlap of separate pages of geometry. This is a very efficient mechanism for avoiding the problems of page overlap. It is easy to implement because of the finite radius of curvature of the vehicle.

Aging is performed on a cell-by-cell basis in the map so that old pages do not poke through the holes in new images. They will be correctly discarded based on age even though they are still physically in the map. The mechanism works correctly for a stationary vehicle because distance is used instead of time as a measure of age.

2.6 Read Map

This module performs the physical read of memory when a query is received.

2.7 Age Filter

When the map is read, this module checks the age of the information against the current distance of the vehicle. If the information is too old, the cell is treated as unknown outside the map manager.

Section 3: Stereo Vision

The implementation of correlation-based stereo vision is a straightforward series of transformations of the input image based on the techniques used in [71]. The data flow diagram for this component is given below.

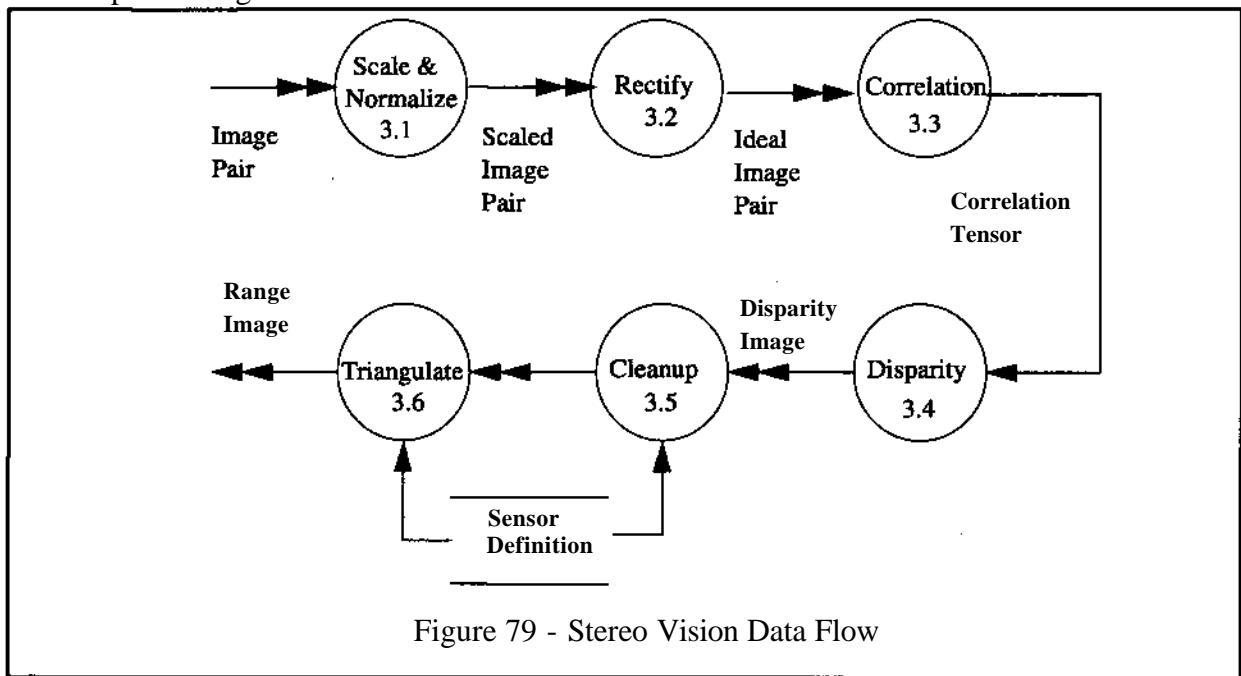


Figure 79 - Stereo Vision Data Flow

3.1 Scale and Normalize

This module replaces each pixel with its normalized value based on the mean and standard deviation of intensity over the correlation window. This operation enhances texture and removes localized bias from the image. An efficient moving average algorithm is used to compute the convolutions quickly.

3.2 Rectify

This module warps the input images into ideal epipolar geometry. This operation ensures that a match for a given region of the first image will be found along the same row (horizontal baseline) or same column (vertical baseline) of the second image. The inverse rectification matrix is used so that processing time is proportional to the number of output pixels rather than the number of input pixels. When speed is important, the scaling and rectification operations can be merged together very efficiently by using the inverse rectification matrix.

3.3 Correlation

This module computes the correlation between each window in the first image and each window in the second image within the disparity image. An efficient moving average algorithm is used to compute the convolutions quickly.

3.4 Disparity

This module finds the disparity value at which the correlation curve for each pixel in the first image is maximum. Optionally, a subpixel disparity estimation routine computes subpixel disparity. Pixels whose maximum correlation scores are low or whose maxima are not sufficiently unique are marked as bad.

3.5 Cleanup

This module uses a classical blob-coloring algorithm to remove those connected regions in the disparity image whose overall size is a small fraction of the total image.

3.6 Triangulation

This module uses the camera model of the first camera to convert disparity to (x,y,z) coordinates in the camera frame.

Chapter 4: Results

Section 1: Parameter Configuration

The following two section will present results for perception based on laser range images and stereo vision. In both cases, the following configuration of important system parameters applies. While adaptive perception resamples a range image for optimum coverage of the terrain, the specific attributes of the range sensor and cameras used for the following results are given in the table below:

Table 12: ERIM Range Sensor Parameters

Attribute	Value
Image Rows	64
Image Cols	256
HFOV	80°
VFOV	30°
HIFOV (angular resolution)	0.3125°
VIFOV (angular resolution)	0.4688°
Frame Rate	2 Hz

Table 13: Camera Parameters (Vertical Baseline Case)

Attribute	Value
Image Rows	640
Image Cols	486
HFOV	20°
VFOV	20°
HIFOV (angular resolution)	0.0412°
VIFOV (angular resolution)	0.0312°
Frame Rate	30 Hz

Adaptive perception and adaptive regard are affected primarily by the following parameters:.

Table 14: Configuration Parameters

Attribute	Value
Vehicle Speed	3 m/s
Imaging Density	1.2
Column Oversampling Factor	2
Row Oversampling Factor	2
Map Resolution - x	0.75 m
Map Resolution - y	0.75 m.
# Turn Commands	15

Section 2: Range Images

In a typical image, the pixels that are actually processed by the adaptive perception algorithm form a jagged-edged band across the horizontal. The width of the band decreases quickly as the vehicle speed increases and adaptive lookahead moves the window up the image. However, the validity of the **small incidence angle assumption**¹⁰ guarantees that adaptive perception will generate a perfect wedge of geometry which is exactly the requirement for the current planning cycle regardless of the vehicle attitude or terrain shape.

The following figure gives a sequence of range images for a run of the system simulator on very rough terrain using a simulated rangefinder where the pixels that were actually processed fall between the thin black lines. On average, even in this worst case, only 75 range pixels out of the available 10,000 (or 2%) were processed per image¹¹. Thus, the 2% geometric efficiency of the sensor is effectively increased to 100% and throughput is increased by a factor of 100 times, or two orders of magnitude.

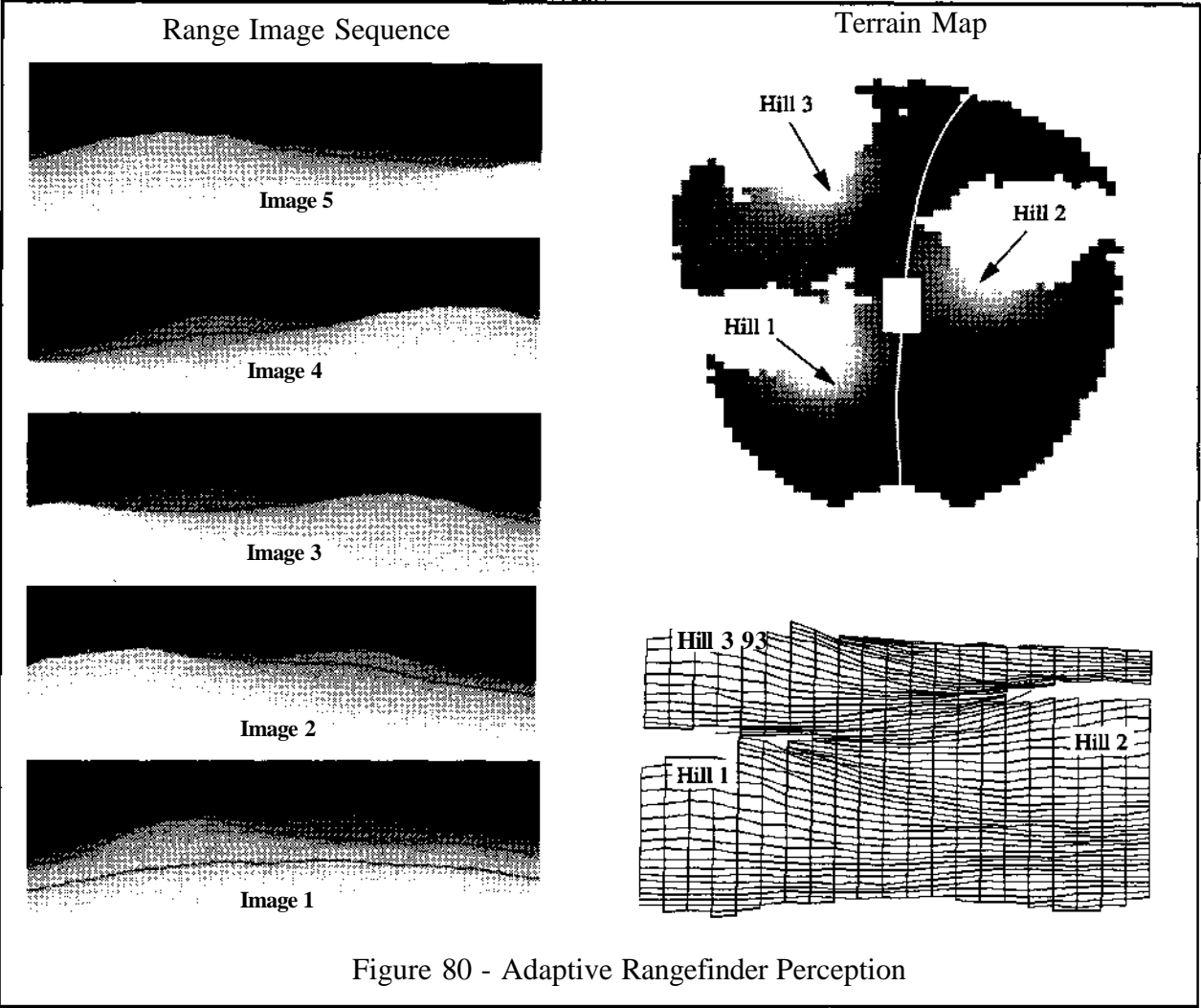
There are five range images arranged vertically on the left. These are rendered as intensity images where darker greys indicate increasing distance from the sensor. The terrain map constructed by the perception system is rendered on the right. The top figure shows the map as an image where lighter greys indicate higher elevations. In the center of the map is the vehicle at the position where the 5th image was captured. The lower right figure is the same terrain map rendered as a wireframe surface from the vantage point of the initial position.

There are three hills in the scene whose range shadows are clearly visible in the terrain map. In the first image, the vehicle is accelerating but still travelling relatively slowly. The range window is relatively wide and positioned near the bottom of the image. The first hill is in the range window. In the second image, the second hill is in the range window and the first hill has already been processed. Indeed, none of the left side of the image is processed because the data in the range window is occluded. In the third image, the third hill is now in the range window. In the fourth image, the vehicle is driving past the first hill and is rolled to the right because of it. This rolls the image to the left and the algorithm compensates appropriately. In the fifth image, the range window has moved past the third hill to the flats beyond and a fourth hill is barely visible in the distance.

The system performs identically on real images, but simulated ones were used here to illustrate several points within a limited space.

10. Of course, it is the relative accuracy of the small angle assumption across two images which really matters for throughput reasons, so the continuity assumption makes the small angle assumption even more valid.

11. Note that the 100 times improvement is consistent with Figure 59 at a speed of 5 meters/second. This is an experimental and theoretical result. At 20 mph, the improvement is theoretically 4 orders of magnitude.



Actual perception performance is given in the tables below for a series of images of flat terrain:

Table 15: Adaptive Perception Performance

Attribute	Value
# Pixels Processed	75
Run Time	0.022 sees

Table 16: Nonadaptive Perception Performance

Attribute	Value
# Pixels Processed	16384
Run Time	0.352 secs

Performance on rough terrain can be expected to be worse than this because vertical surfaces project into larger regions in the image plane than do horizontal ones¹². The results do not scale linearly because the adaptive result includes a constant setup time. Nonetheless, the adaptive result is 16 times faster than the nonadaptive result and if the ERIM sensor had better angular resolution, the improvement would be proportionally better.

In more concrete terms, the example can be understood in terms of the number of scanlines per image that would be processed if the terrain were flat. If the terrain were flat, there would be no occlusion and the number of pixels processed per image would be on the order of 100, not the 75 given in the table. There are 256 range pixels in azimuth. These are subsampled at roughly a factor of 8. The angle spanned between pixels is thus 2.5°. The maximum range of the range window was 15 meters. Thus, the horizontal spacing of pixels is 0.65 meters which is within the map resolution.

In the downrange direction, there are about 3 scanlines processed per image because there are 100 total and 32 in a line. Thus, the angular width of the region of interest is 1.4°. Using the acuity relationships in the downrange direction, the distance projected onto the groundplane is 1.83 meters for three pixels or 0.61 meters per pixel which is also within the map resolution.

Clearly, the system achieves barely adequate spatial resolution and hence minimum throughput. The system achieves the throughput necessary for 20 mph rough terrain autonomy on a typical engineering workstation.

12. Thanks to Martial Hebert for pointing this out.

Section 3: Stereo Vision

3.1 Horizontal Baseline

The following figure illustrates the operation of adaptive stereo on two horizontal baseline input images. The initial input images appear at the left. The normalized, texture-enhanced images appear to the right of the input images. To the right are the nonadaptive and adaptive disparity and range images. The disparity images are shown to demonstrate the spurious matches which are caused by incorrectly chosen extrema in the correlation curve. In the case of the adaptive version, further spurious matches are a by-product of the disparity window approach.

In the adaptive case, some of these spurious matches correspond to local maxima in the correlation curve, but there is no information available to detect this. Finally, the cleaned-up range images are presented below the disparity images. The cleanup algorithm incorporates an efficient filter, based on classical region-finding techniques, which removes the local maxima and provides a clean range image for processing by the rest of the navigation system:

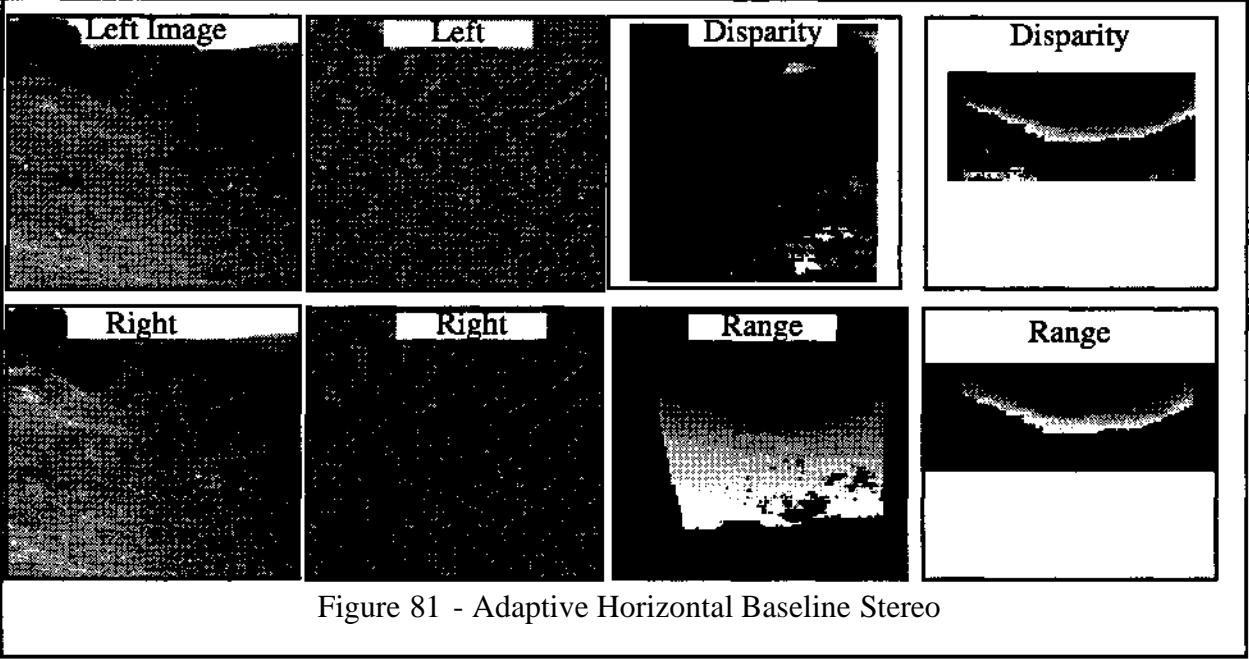


Figure 81 - Adaptive Horizontal Baseline Stereo

A breakdown of this horizontal baseline run is shown in the table below:

Table 17: Horizontal Baseline Stereo - SPARC 20

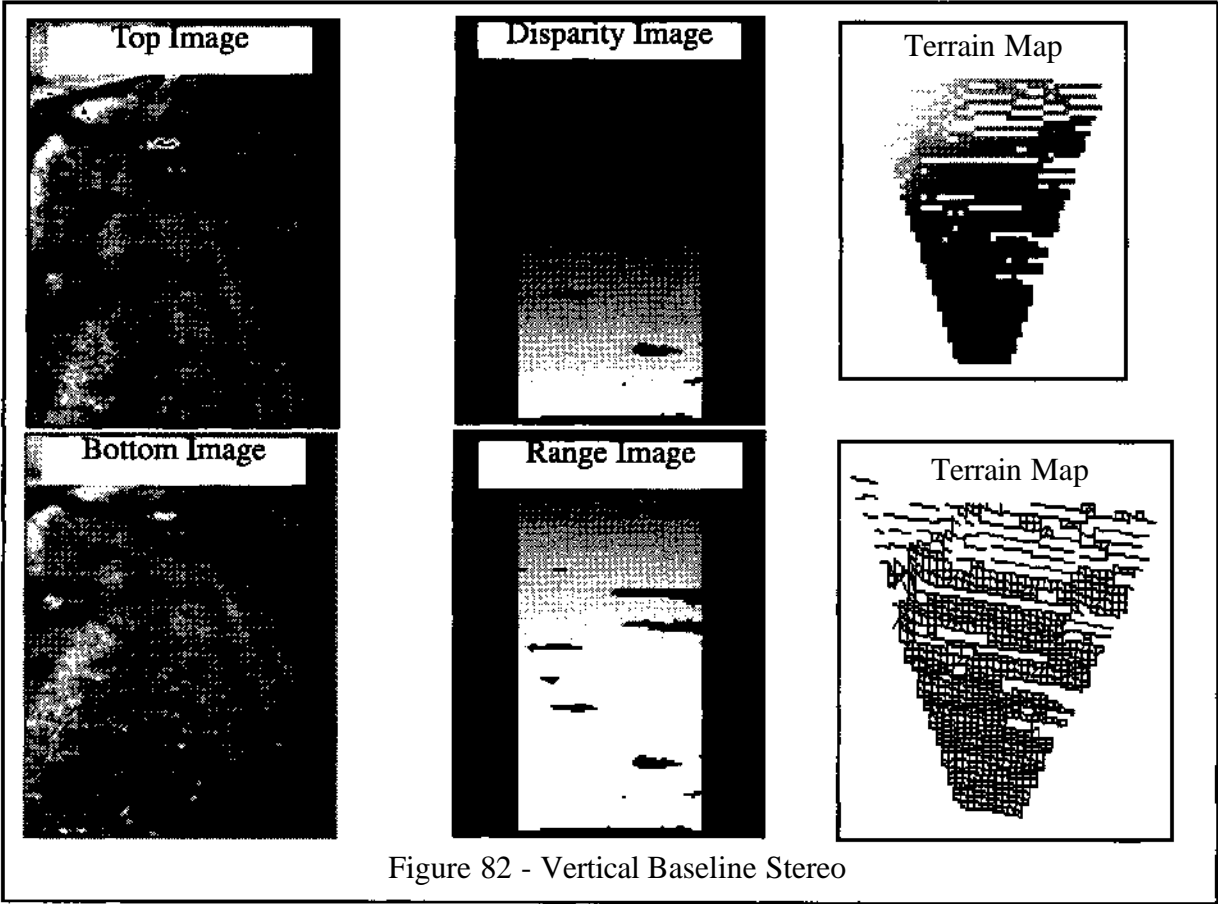
Attribute	Value (Nonadaptive)	Value (Adaptive)
Input / Output / Adaptive Rows	480/120	480/120/48
Input / Output / Adaptive Cols	512/128	512/128/128
Disparities	60	10
Setup	9 msec.	1 msec.

Table 17: Horizontal Baseline Stereo - SPARC 20

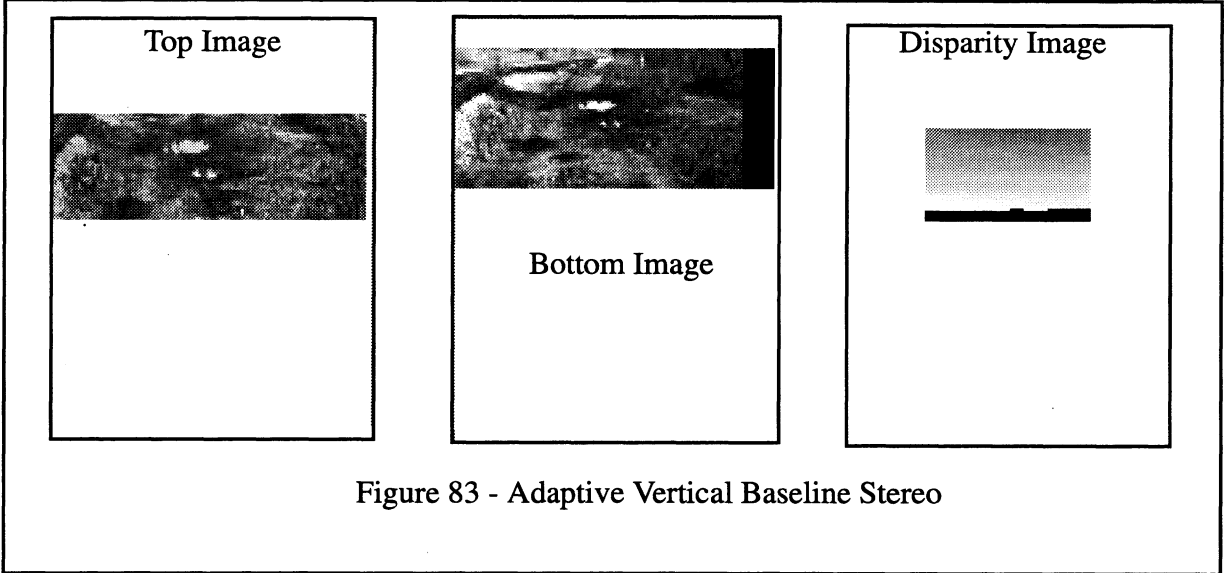
Attribute	Value (Nonadaptive)	Value (Adaptive)
Scale & Rectify	23 msec.	10 msec.
Normalize	70 msec.	30 msec.
Correlation	683 msec.	69 msec.
Disparity	713 msec.	55 msec.
Cleanup	25 msec.	10 msec.
Triangulate	16 msec.	9 msec.
Total Runtime	1539 msec.	203 msec.

3.2 Vertical Baseline

The feasibility of vertical baseline stereo is demonstrated in the following vertical stereo pair.



A run of the same stereo pair from the previous example is shown below. This run is of sufficient density to completely populate a 1/2 meter resolution terrain map between the ranges of 25 meters and 30 meters.



A breakdown of both vertical baseline runs is shown in the table below:

Table 18: Vertical Baseline Stereo - SPARC 20

Attribute	Value (Nonadaptive)	Value (Adaptive)
Input / Output / Adaptive Rows	640 / 160	640 / 160 / 45
Input / Output / Adaptive Cols	486 / 123	486 / 123 / 123
Disparities	60	10
Setup	1 msec.	4 msec.
Scale & Rectify	32 msec.	11 msec.
Normalize	84 msec.	26 msec.
Correlation	532 msec.	50 msec.
Disparity	14 msec.	4 msec.
Cleanup	22 msec.	6 msec.
Triangulate	8 msec.	5 msec.
Total Runtime	700 msec.	96 msec.

These results imply 10 Hz update rates on a serial workstation - ignoring the time required to i/o (which can be made negligible).

Chapter 5: Summary and Conclusions

This section summarizes all of the most important content of Part IV: Perception.

In Part II: Analysis, Chapter 6: Interactions, it was shown that an autonomous vehicle requires computational throughput on the order of $O(TV^2J)$ where T is the vehicle reaction time and V is the velocity. On the other hand, the traditional approach of nonadaptive range image processing requires throughput on the order of $O(T^4V^4J)$. The product TV is on the order of 10 for a conventional automobile so the difference between these two expressions is four orders of magnitude at 20 mph. Nonadaptive range image processing requires about 200 Megaflops in order to achieve 20 mph speeds whereas the algorithm presented here requires 3/100 of 1 Megaflop under identical assumptions(see page 135).

An adaptive approach to perception based on the techniques of **adaptive sweep** and **adaptive scan** has several advantages. A more complete list of these advantages is as follows:

- It has the potential to solve the **throughput problem** which has tended to increase vehicle reaction time beyond the regimes of usefulness in real-time vehicle control.
- It amounts to a computational solution to the stabilization problem within the limits of the sensor field of view. It essentially converts any imaging sensor into an ideal adaptive line scanner.
- For limited field of view sensors, it provides an obvious basis for the generation of sensor pointing commands which keep the region of interest centered in the image.
- It solves the sampling problem for practical purposes because variation in the range ratio is very low over the small elevation width of the range window.
- It theoretically guarantees throughput, response, and acuity within the limits of the vehicle and the sensor.
- It adapts to vehicle speed, attitude and terrain shape implicitly.

This part concentrated on the adaptive perception algorithm which forms the basis of RANGER'S Map Manager and Perception objects. The techniques used should be applicable to any application that models the environment with a terrain map.

Section 1: Concept

An **adaptive perception** algorithm is the organizing principle of the design of the navigator's perception system. It confines the processing of range geometry to a **focus of attention** or **region of interest** which extends to the end of the detection zone - but whose size is governed by the distance moved since the last perception cycle. Adaptive perception has three conceptual parts:

- **adaptive lookahead** - set the maximum range to the end of the detection zone
- **adaptive sweep** - set the region of interest width to the distance travelled since last time
- **adaptive scan** - manage image resolution to achieve uniform groundplane resolution

The motivation for this approach to perception can be expressed in familiar terms. Analysis suggests that much of the computational resources used to image and interpret the environment is a waste of resources in mobility scenarios. This waste occurs for two principle reasons:

- The sensor vertical field of view is aligned with the direction of travel so that image sequences normally contain redundant information. Obstacles appear in the field of view long before they can be resolved, and long after they cannot be avoided.
- The projection of image pixels on the groundplane is elongated in the wrong direction for robust obstacle detection and minimum throughput.

1.1 Adaptive Lookahead

The design of **adaptive lookahead** is a straightforward matter. There is some maximum range beyond which it is unnecessary to look because there will be time to look there later. In detail implementation, the responsibility for computing this maximum range is left to adaptive regard and the results of this computation are used to bound the far end of the focus of perceptual attention.

1.2 Adaptive Sweep

The design of **adaptive sweep** is also relatively straightforward, once a key assumption is identified. Any straightforward attempt to selectively process data in an area of interest apparently falters because *the problem of selection is as difficult as the problem of perception*. Luckily, for high-speed autonomy, the **small incidence angle assumption** solves the problem of selection inexpensively by decoupling the problems of selection from perception.

The focus of attention is specified as a **range window** - some maximum and minimum range within which the completely processed data must lie. In the case of stereo vision, the basic principle of the range window can be converted to a **disparity window**. This simple mechanism points the sensor vertical field of view computationally - providing implicit instantaneous adaptation of the focus of perceptual attention based on both the vehicle attitude and the shape of the external terrain.

1.3 Adaptive Scan

The design of **adaptive scan** is a straightforward matter. Over the width of the range window, the aspect ratio of a range pixel typically varies little - again because of the small incidence angle assumption. Therefore, for the data of interest, a constant mapping from world model resolution to image resolution can be assumed and images can be subsampled by appropriate factors to achieve that resolution.

Section 2: Design

2.1 Adaptive Perception From Range Images

In the case of laser rangefinders, a range image is generated in hardware. For this class of sensors, adaptive perception is limited to extraction of the range window from the complete image. The **adaptive perception** algorithm constitutes a simultaneous implementation of **adaptive lookahead**, **adaptive sweep**, and **adaptive scan** in one place. It proceeds in two phases: the computation of the range window based on the current speed and cycle time; and the mapping of this window into image space and the extraction of the data.

Adaptive lookahead is implemented by computing the distance required to execute an impulse turn¹³ at the current speed. This gives the maximum range of the range window. Adaptive sweep is implemented by computing the distance travelled based on the current and last cycle distance travelled and subtracting this distance from the maximum range. The problem of finding this data in an image taken previously involves several aspects of time delays and geometric offsets:

- The sensor is not mounted at the vehicle control point.
- The vehicle is not itself a point
- There may be significant delay associated with the acquisition of an image.
- The most recent vehicle state estimate is itself somewhat old.
- It will take some time before the command computed this cycle reaches the output communications routines.

By the **small incidence angle assumption**, the projection of the sensor range onto the groundplane is essentially the groundplane y coordinate. However, terrain roughness and nonzero vehicle roll mean that the position of the range window in the image is *different for each column*. Thus the range window is processed on a per column basis. In order to robustly find the range window, each column is processed in the bottom-to-top direction.

2.2 Adaptive Stereo Perception

The range of disparities which corresponds to a typical range window is very small. This implies that any process which robustly identifies global maxima of the disparity curve can generate the range window in an image. Such a process implements **adaptive sweep** inside the stereo algorithm. The essential problem with this simple idea is that local extrema in the correlation curve masquerade as a global extremum.

Several solutions to this **repetitive texture problem** help the situation somewhat but the simple technique of computing connected components and removing small regions [59] works effectively and is basically free because a disparity image cleanup pass is required even when a wide disparity range is searched.

2.3 Refinements to Correlation-Based Stereo

This section introduced some refinements to correlation-based stereo vision that adapt the traditional algorithm to the specifics of the application.

2.3.1 Optimum Shape of Correlation Window

It was shown that the optimum aspect ratio of the correlation window is the inverse of the ratio of the corresponding components of the disparity gradient. Of course, the disparity gradient is unknown until the range image is available, but it is a good assumption in outdoor work that it is mostly vertical in most places in most images.

13. An impulse turn was chosen as the maneuver upon which to base adaptive lookahead because a vehicle which stops when it sees an obstacle (panic stop maneuver) is not useful when obstacles are dense. The impulse turn is a turn from zero curvature to the maximum. Other distinguished maneuvers are the panic stop, the turning stop, and the reverse turn.

2.3.2 Complexity of Stereo Vision

It is a straightforward matter to compute the complexity of stereo vision when implemented on a serial workstation by computing the complexities of the component processes. Complexities were computed for traditional nonadaptive stereo, partially subsampled stereo, and completely subsampled stereo. It was shown that **completely subsampled** stereo is theoretically an order of magnitude faster than traditional stereo in the target application and that this theoretical speed improvement is slightly less in the case of **partially subsampled** stereo.

2.3.3 Adaptive Scan in Stereo Vision

It was shown that because **geometric decorrelation** is identical in horizontally and vertically triangulated systems, distortion is no basis for preferring one to the other. Then it was shown that:

- Horizontal triangulation couples crossrange resolution to downrange resolution.
- Vertical triangulation couples vertical resolution to downrange resolution.

Adaptive scan can be implemented in either latter case for an overall improvement in throughput roughly equal to the inverse perception ratio. **Vertical baseline stereo** has other advantages as well. Some of these are:

- Complete subsampling is more straightforward to implement than partial subsampling, because this allows the requirement on the horizontal resolution to be relaxed at the input to the stereo algorithm. One simply runs the traditional stereo algorithm on rotated and scaled input images.
- It does not reduce the horizontal field of view (which is often already too small) but instead reduces the vertical field of view (which is often larger than necessary).
- It promises to permit extremely wide field of view stereo from a single camera pair because the distortions introduced by a wide field of horizontal view do not matter much when triangulating vertically.
- It permits a straightforward operation to remove the distortions due to the average expected disparity gradient. The vertical dimension of the bottom image is simply scaled spatially by the inverse of the disparity gradient.

2.3.4 Adaptive Sweep

An implementation of **adaptive sweep** for stereo vision was introduced based on the kinematics of projection and a rough flat world assumption.

2.4 Terrain Mapping

This section described the terrain map data structure which is used to model the environment in rough terrain. Terrain mapping is the process by which surface descriptions, obtained from different vantage points, are accumulated into a consistent environmental model [31]. In order to provide the path planner with a single, coherent, uniform density data structure that may span several images, the approach taken is to transform images into a regularly-spaced cartesian grid called a Cartesian Elevation Map (CEM) or simply a **map**. The use of a terrain map normally means that the **2-1/2 D world assumption** is being adopted.

2.4.1 Wrappable Map

The overhead involved in representing vehicle motion through the environmental model can be overwhelming unless something is done to make the process efficient. The navigator introduces a wrappable map data structure which represents this motion at negligible computational cost. The impact of this data structure is that it is never necessary to perform a copy of the map data structure, or completely traverse it for any reason.

Although all data remains in the map until it is overwritten, each entry is tagged with the distance that the vehicle had travelled since the start of the mission at the time the pixel was measured. If the "age" of the last update is too old, the cell is considered to be empty¹⁴. This technique eliminates all of the overhead of map management except for the coordinate transformation necessary to access it.

2.4.2 Motion Distortion Removal

By the time an image is received by the perception system, the vehicle may have moved a considerable distance since the image was acquired. So, the processing of the geometry in the image must account for the exact position of the vehicle when the image was taken. Other distortions result from the non-instantaneous scanning of laser rangefinder images.

This distortion of range images is removed by maintaining a history of vehicle poses sampled at regular intervals for the last few minutes of execution and searching this list for the precise vehicle position at which each range pixel was measured.

2.4.3 Delayed Interpolation

The terrain map is not interpolated at all because interpolation requires a complete traversal which is too expensive to perform. Instead, the responsibility for interpolation is left with the users of the map. The path planner *interpolates in time instead of in space* because this is a more efficient solution to the sampling problem.

2.4.4 Image Registration

A simple image registration algorithm is used in situations where the imaging density is reduced below the amount necessary to ensure that the geometry under the rear wheels comes from the same image as the front wheels in the feedforward simulation.

2.4.5 Elevation Uncertainty

In order to represent the variation in geometry in a single map cell and to improve signal to noise ratios, a **scatter matrix** is computed [35] incrementally as each new range pixel is merged into the map.

14. Age is measured by distance and not time because otherwise the geometry under the wheels will eventually disappear when the vehicle stops.

Section 3: Implementation

This section overviewed the implementation of the perception system. This subsystem incorporates

- An image processing element which converts a raw image into an ideal image.
- An adaptive perception element which finds the region of interest in the image.
- The coordinate transformations necessary to create the terrain map.

3.1 Coordinate Transformation

Old vehicle positions are used in the coordinate transformation in order to remove the distortion associated with the delay of the perception and communication electronics. The elements of this subsystem are Ambiguity Removal, Median Filter, Range Window Filter, Range Window Computation, Image to Sensor, Sensor to Vehicle, Vehicle to World, and Pose Selector.

3.2 Map Manager

The map manager deals with the transformation of data into and out of the discrete representation of the terrain map. The elements of this subsystem are World to Map, Map to World, Registration, Fusion/Accumulation, Write Map, Read Map, and Age Filter.

3.3 Stereo Vision

The implementation of correlation-based stereo vision is a straightforward series of transformations of the input image based on the techniques used in [71]. The elements of this subsystem are, Scale and Normalize, Rectify, Correlation, Disparity, Cleanup, and Triangulation.

Section 4: Results

4.1 Range Images

A sequence of range images for a run of the system simulator on very rough terrain was provided in order to demonstrate the feasibility and performance of adaptive perception from range images. On average, only 75 range pixels out of the available 10,000 (or 2%) were processed per image. The system performs identically on real images, but simulated ones were used here to illustrate several points within a limited space.

In terms of runtime performance, the adaptive algorithm is 16 times faster than the nonadaptive result and if the simulated sensor had better angular resolution, the improvement would be proportionally better. The system achieves the throughput necessary for 20 mph rough terrain autonomy on a typical engineering workstation.

4.2 Stereo Vision

The operation of adaptive stereo on two horizontal baseline input images was illustrated. The feasibility of vertical baseline stereo was also demonstrated in a vertical stereo pair. The results imply 10 Hz update rates on a serial workstation - ignoring the time required to i/o (which can be made negligible).

PART V: Planning/Control

This part describes the state space model which forms the core of the planning system. This part concentrates on the trajectory tracking, obstacle avoidance and sensor stabilization aspects of the navigator. These algorithms form the basis of RANGER'S Controller and Vehicle objects.

An approach to the high-speed mobility problem is presented which is based fundamentally on the state space representation of a multi-input / multi-output dynamical system and the problem analysis presented in Part II: Analysis.

Chapter 1: Concept

The principle of adaptive regard is the organizing principle of the design of the navigator's planning system. **Adaptive regard** is a notion applied to the planning problem. Essentially, this mechanism confines the search for hazards through the environmental model to the **detection zone** - that region of the near environment which the vehicle can reach and is not already committed to travelling. Thus, this mechanism applies to the terrain map and the planning problem and is based on a response argument. Adaptive regard is based on four principles:

- don't look where the vehicle cannot go (free zones)
- don't look where the vehicle is committed to going (dead zone)
- don't look where the vehicle will get another look later (beyond detection zone)
- look where there is an immediate decision of traversability required (detection zone)

The motivation for this approach to planning can be expressed in familiar terms. Analysis suggests that much of the computational resources used to plan paths through the environment are a waste of resources in high-speed mobility scenarios. This waste occurs because:

- Dynamics and other constraints of many kinds including braking and steering maneuverability, underactuation of the vehicle, and processing and communication delays amount to an overwhelming constraint on the feasibility of arbitrary trajectories expressed in configuration space.
- Search through any space of candidate trajectories which do not respect the dynamics constraint amounts to a lot of time wasted thinking about the impossible which tends to increase reaction time and therefore to limit system performance.

In order to implement adaptive regard, several component problems must be solved. First, the hierarchical approach to planning implicit in the standard model is used to augment the local navigator by the trajectory recommendations of the strategic layer. Second, an optimal control formulation is used to merge the actuator commands that result from obstacle avoidance and goal-seeking components of the planner in order to solve the actuator contention problem. Third, the state space modelling technique of multivariate control theory is used as the mechanism by which

vehicle response is predicted and therefore by which the detection zone is computed. Fourth, an actuation space search strategy is used in cooperation with forward modelling to implement the optimal control approach.

Section 1: Optimal Control

In this domain of autonomous mobility, the problem is that of achieving some useful goal while simultaneously avoiding damage to the vehicle. The goal of the vehicle may be to move from its initial position to some distant other position, to map an entire area, or to search an area for objects of interest. It may also be useful to optimize use of fuel, distance travelled, or some other criterion over the mission.

In realistic terrain, the vehicle is challenged by regions that would cause tipover, trapped wheels, or loss of traction. Some regions are not traversable at all and others may cause disastrous system failures like falling into an abyss. On rough terrain, the concept of an obstacle can be generalized to that of any unsafe condition as was initially done for the Autonomous Land Vehicle [13].

This section outlines an approach to high-speed cross-country navigation based on an actuation space search expression of path planning. The approach discussed here applies to high-speed, rough-terrain, cross-country navigation. The navigator attempts to travel continuously when the terrain permits, and to traverse, when necessary, any terrain that the vehicle can negotiate.

1.1 Hierarchical Planning

The **local minimum problem** is perhaps the most serious threat to robustness of any local obstacle avoidance strategy for autonomous navigation. In order to achieve robust navigation, a sound strategic plan that avoids these minima is a solution that works well in practice. In order to enhance robustness, the navigator supports the input and continuous monitoring of strategic goals. With goal-seeking and obstacle avoidance goals operative simultaneously, a means for arbitrating between them in order to avoid actuator contention is necessary. Thus, the highest level design strategy for the planning/control system is *hierarchical planning with arbitration* as shown below:

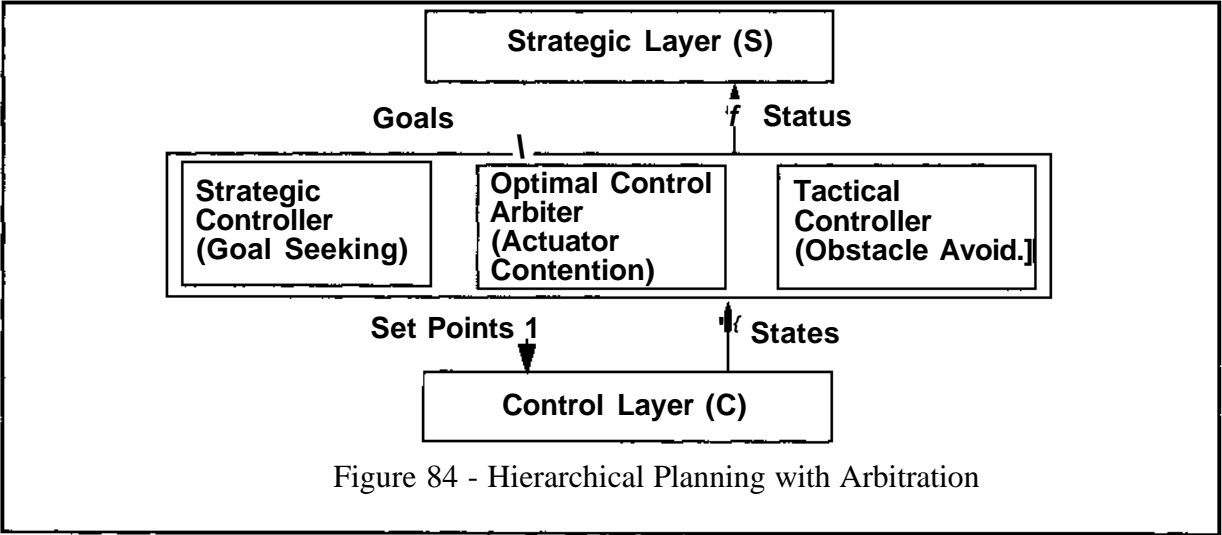


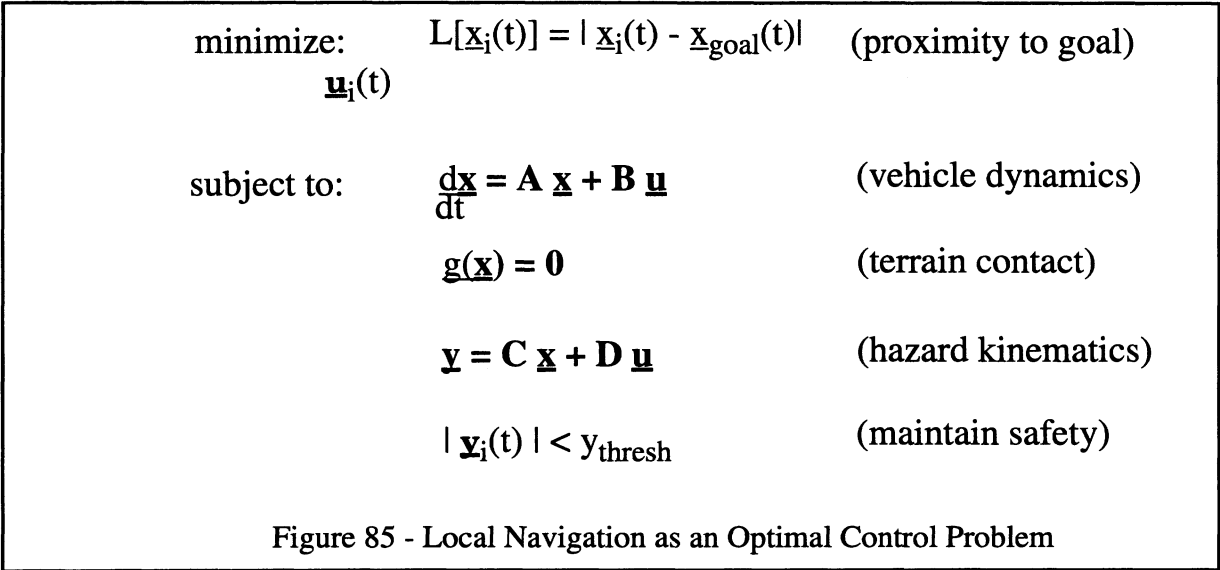
Figure 84 - Hierarchical Planning with Arbitration

The navigator defines a **strategic goal** as some path or position to be followed or achieved and a **tactical goal** which is to simultaneously avoid all hazardous conditions. The essential problem of this hierarchical approach is that these goals conflict at times and the conflicts between them must be resolved in some way. This is the **actuator contention problem**.

While there are many potential conflict resolution strategies, the one used here is to notice that the problem can be expressed in the familiar terms of optimal control theory and to then apply the associated techniques and abstractions of this theory to the problem as described in the next section.

1.2 Optimal Control

The task of safe navigation can be expressed in classical optimization terms. The navigator must achieve some useful goal while simultaneously satisfying the constraint of avoiding damage to the vehicle or its payload and the constraints of limited vehicle maneuverability. Navigation of such a system can be cast as a suboptimal solution to the following optimal control problem:



for some suitable norm (| |) of $\underline{y}(t)$.

The concept of **state space** is well-known from control theory. It is spanned by the **state vector** \underline{x} . The **actuation space** is the space spanned by the **command vector** \underline{u} . The **hazard vector** \underline{y} can be considered to span an abstract **hazard space**. Thus, corresponding trajectories can be formed in actuation space ($\underline{u}_i(t)$), state space ($\underline{x}_i(t)$), and hazard space ($\underline{y}_i(t)$) which represent a command under consideration and the state space and hazard space trajectories to which it corresponds.

The **objective function** $L[\underline{x}_i(t)]$ expresses the need to cause the navigator to perform some useful function. Optimal performance against this requirement is achieved by minimizing the deviation of the state space trajectory from the goal trajectory. The goal trajectory may be a path to follow, or a point in space, or generally any arbitrary trajectory $\underline{x}_{goal}(t)$. Higher-level abstractions of the objective function to include fuel consumption, complete coverage of an area etc. are also possible.

1.3 State Space Model

The navigator is a state space controller because it explicitly forms an expression of the vehicle dynamic state vector in order to predict the hazard signals upon which decisions are based. Instead of forming an **inverse model** which attempts to compute curvature and speed from a C-space curve, a **forward model** is used which implements the equations of dead reckoning. This process which converts an actuation space trajectory to a state space trajectory is a *constrained multidimensional differential equation* called the **state space model**.

This model encodes actuator constraints, kinematics, and dynamics, body dynamics, and terrain contact. In the model, A models the steering and propulsion dynamics, B models the communications delays, and the abstract kinematic equation $g(\underline{x}) = 0$ models the terrain contact constraint.

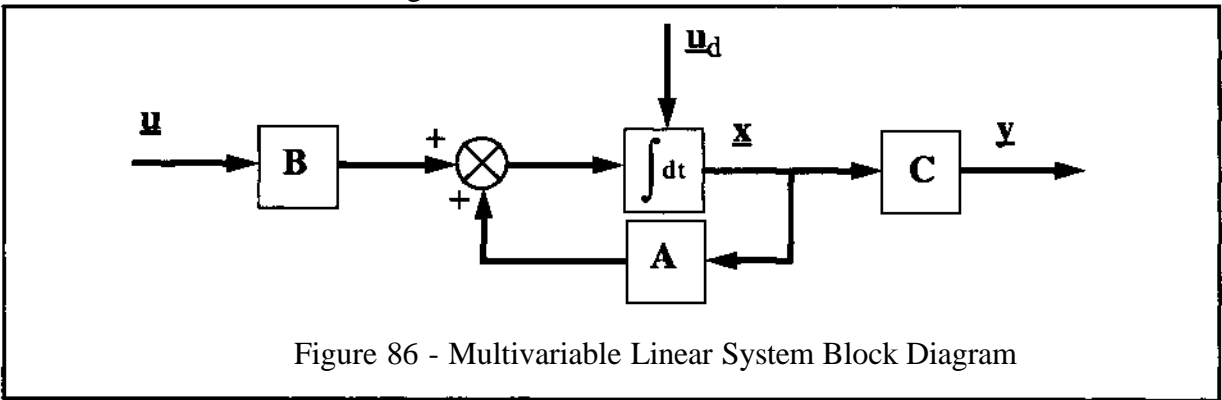
This kind of model is known classically as a **multivariable state space system** (Figure 86). For a *linear system*¹, the conventional state space model of a system is the venerated two matrix equations:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

Note in particular that the first equation is a differential one. The **command vector** \mathbf{u} includes vehicle steering and speed as well as demand signals to any sensor heads. The terrain disturbances \mathbf{u}_d model the terrain contact constraint². Terrain geometry is represented in a terrain map data structure that is generated by the perception system.

The **state vector** \mathbf{x} includes the vehicle steering, speed, and the state of motion of the vehicle body and the sensor heads. It includes the 3D position and 3-axis orientation of the vehicle body as well as its linear and angular velocity.

The **output vector** \mathbf{y} is also the **hazard vector** in this case. It can be any function of both state and inputs, and in this case it is also a function of the terrain on which the vehicle rests. These equations are often indicated in a block diagram as shown below:



1. Our system is not linear. No real system is perfectly linear. Consider this a model for now, and the domain of validity of the model to be an open question.
2. Of course, at sufficiently high speeds, the vehicle need not remain in contact with the terrain.

The system dynamics matrix A propagates the state of the vehicle forward in time. The system model is based on the assumption that velocity can be considered constant for a small period of time. While the input commands to the model respect only the maximum curvature constraint, the output state estimate is consistent with the response of all actuators and the body kinematics of motion over rough terrain. Alternative trajectories of the system are expressed in terms of steering alternatives, but more generally, can be any time-varying command vector $\underline{u}(t)$.

Hazards h are complex functions of the vehicle state, the terrain on which it rests, and the input commands. Each element of the hazard vector corresponds to a different hazardous condition and the hazard vector moves over time in hazard space as the vehicle state moves over time in state space.

Once the vehicle trajectory is known, a set of kinematic operators is convolved with the terrain along the known trajectory in order to assess vehicle safety. Safety is considered to be a time-varying, multidimensional field. Later stages in the planner integrate these safety assessments with a higher-level strategic goal and ultimately decide on a command to be sent to the vehicle for the current planning cycle.

1.4 Actuation Space Search

Perhaps the most well-known abstraction used in robot path planning is **configuration space** or **C-space**[55]. For a rigid-bodied vehicle moving in three dimensional space, the C-space can be considered to be a subset of state space - that is, the coordinates of the vehicle control point expressed as (x, y, z, roll, pitch, yaw). C-space has several advantages as an abstract space in which to conduct search. However, C-space techniques do not capture dynamics well.

The system discussed here plans instead in actuation space by considering a number of actuation space alternatives which span the **feasible set** of commands for the vehicle at some gross resolution. For each command, a number hazardous conditions are considered, and these hazards are evaluated at every point on each trajectory. An actuation space approach to path planning has been adopted because:

- *A conventional automobile is underactuated (non-holonomic), so the mapping from C-space to actuation space is under-determined. It is not possible, in general, to compute the speed and steering commands which will cause a vehicle to follow an arbitrary C-space curve.* The use of actuation space and forward modelling avoids entirely the problem of path generation for non-holonomic vehicles.
- *Many constraints on vehicle motion are differential in nature and most naturally expressed in actuation space.* In the case of steering in particular, a kinematic model can be considered to be one for which curvature can be changed instantaneously. Yet, a moderate-speed vehicle cannot reverse curvature within the maximum useful range measurement that a stereo or lidar sensor can produce. Thus, high-speed motion demands that actuator response characteristics be considered, and the natural space in which to model them is actuation space. The use of actuation space permits expression of these dynamic constraints in terms of dynamic models of actuators.
- *Actuation space planning is more computationally efficient. Continuous high-speed motion implies that a vehicle has very little time to react to what it sees. A planner **must** decide what to do under stringent timing constraints or the entire navigator will fail.* Actuation space permits a fast, coarse-grained evaluation of all feasible alternatives. The

problem of C-space combinatoric explosion is solved by planning in actuation space.

The actuation space for a conventional automobile is spanned by the variables of speed and path curvature. These variables map more or less directly to the controls of throttle and steering.

1.5 Feedforward Optimal Control

One of the defining characteristics of intelligent systems is the ability to act now on the basis of the future consequences of candidate actions. This property of **deliberation** is clearest in systems that normally exist in the strategic layer of the standard model and techniques of AI planning [24] have been very successful in solving such problems. However, because the high-speed autonomous navigation problem is highly dynamic and requires time-continuous differential equation models, deliberation here takes the form of **feedforward**.

The navigator attempts to guarantee safety by **predicting** into the future the relationship between future safety and current command options. Then it acts now based on future consequences. It does this by using a **state space simulator**. The details of the previous figure can be suppressed into a single box that generates hazard estimates based on the terrain map, the input commands, and some hidden vehicle state as shown below:

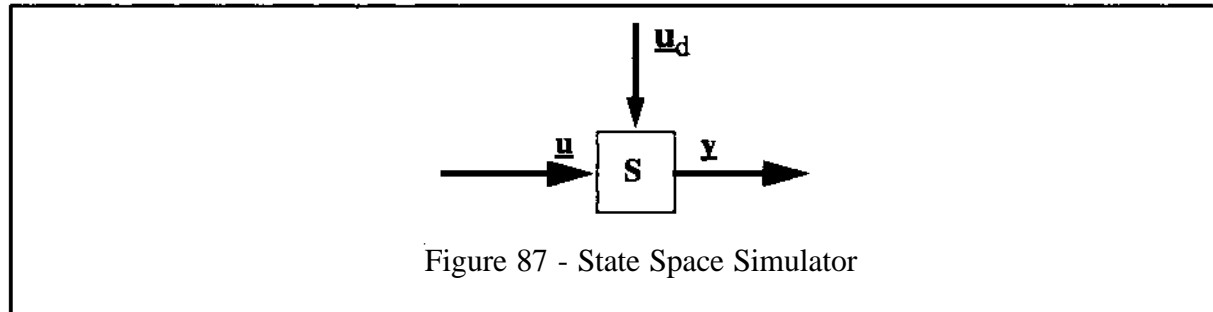


Figure 87 - State Space Simulator

Feedforward is employed for several reasons:

- It imparts stability to both goal-seeking and obstacle avoidance³.
- It computes the detection zone required to implement adaptive regard.
- It allows the planner to conduct its search over the space of feasible commands $\underline{U}(t) \in \underline{U}$ without explicitly considering feasibility because trajectories expressed in actuation space are inherently feasible. Feedforward solves the **clothoid generation problem** trivially.

The system model amounts to a complicated differential equation constraint. The satisfaction of this constraint is generally very difficult to achieve unless alternatives are expressed in actuation space where the constraints are trivially satisfied. The set of trajectories $\underline{X}_j(t)$ which:

- satisfies the system model equations ($d\underline{x}/dt = \underline{A}\underline{x} + \underline{B}\underline{u}$ and $y = \underline{C}\underline{x} + \underline{D}\underline{u}$)
- maintains the vehicle in contact with rigid terrain ($\underline{g}(\underline{x}) = 0$)
- and avoids dangerous hazards ($|\underline{r}_j(t)| < y \text{ thresh}$)

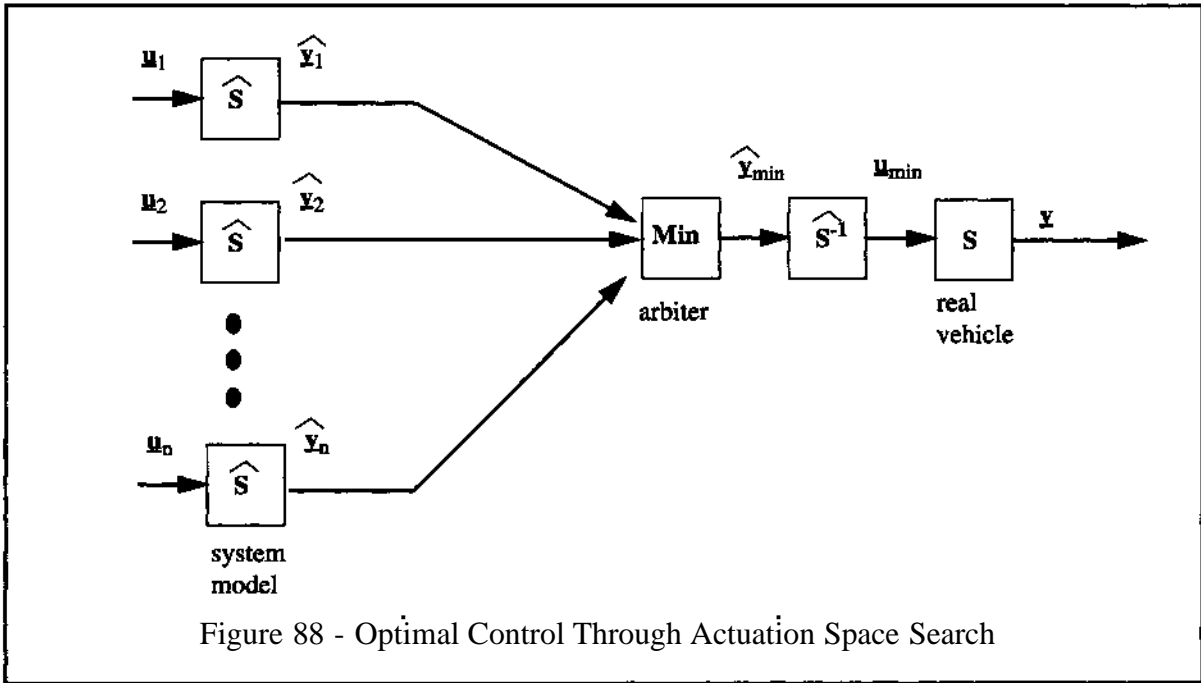
is called the **feasible set**.

3. It has not been mentioned yet, but feedforward prevents "waffling" in the arbiter because once a slight turn is commanded, it affects all subsequent response trajectories by biasing them to reflect the distance the steering wheel will turn for that cycle. Feedforward leads to a more confident controller because it recognizes when a partial commitment has been made to turn in a certain direction.

The navigator satisfies the first constraint *by construction* through feedforward. The second constraint is satisfied *by construction* by altering the vehicle attitude at each step in the simulation. The third constraint is satisfied by pruning alternatives that do not satisfy it in later stages of the planner.

The optimization problem is solved by sampling the feasible set of trajectories at some practical resolution that ensures adequate coverage of the set, and then choosing the trajectory with the best value of the functional $L[\underline{X}_j(t)]$.

The feedforward element can be represented by duplicating the state space model and indicating that it is an estimate with a super hat diacritical mark. The fact that an optimal trajectory is chosen can be represented by a complex minimum transformation. This entire process is illustrated in the following figure:



1.6 Forward Modelling

The differences between classical C space planning and actuation space (A space) planning are indicated in the following figure.

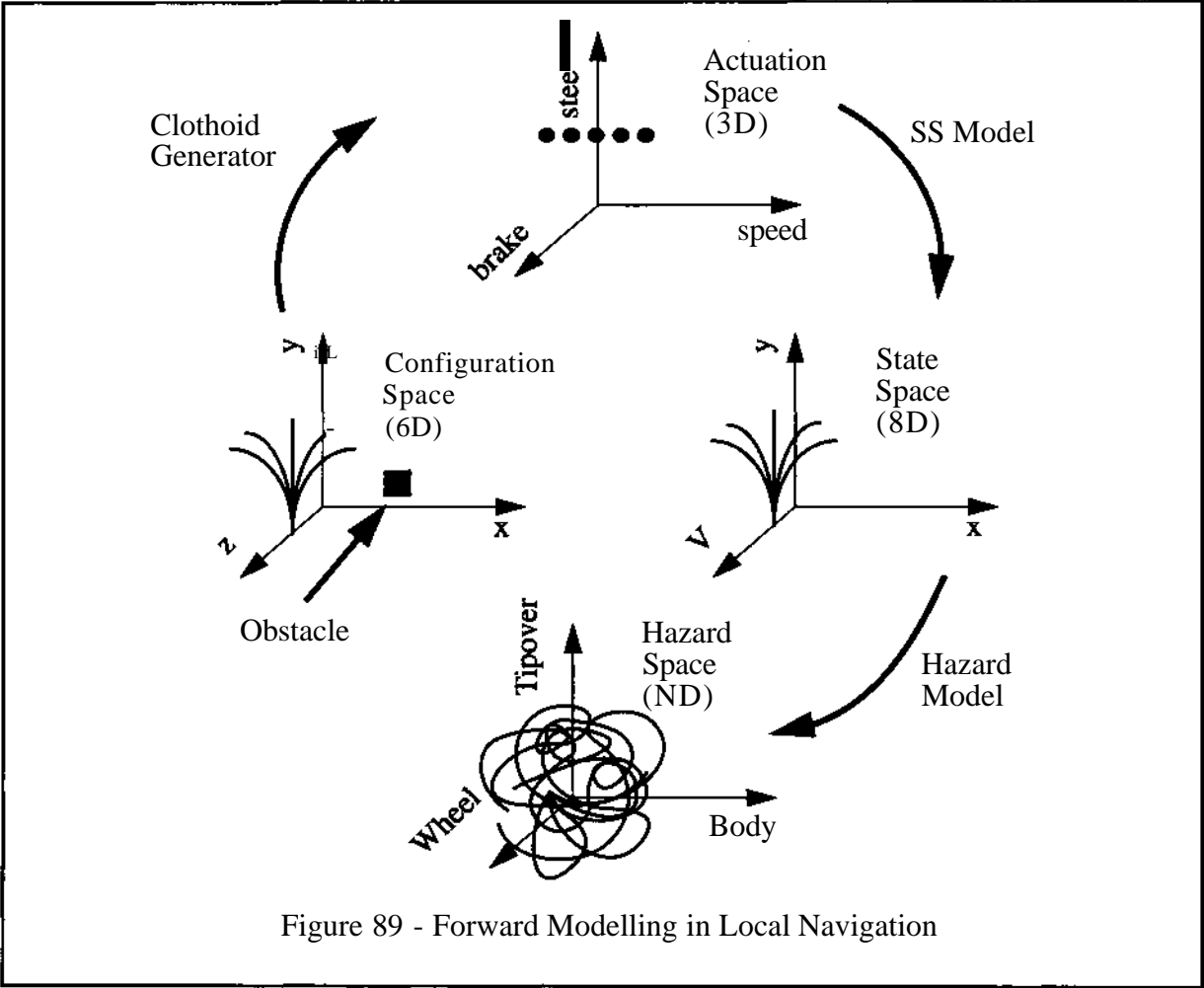
On the left of the figure, the search of planning alternatives is expressed in **configuration space**. A commonly invoked assumption is the expression of obstacles as discrete points in this space. When a clear region or set of points has been found in front of the vehicle, a **clothoid generation** algorithm is invoked to map C space onto the vehicle actuation space and these commands are sent to the hardware for execution.

On the right of the figure is the **actuation space** approach. Note first that the direction of the arrows are reversed. The inverse system model of the above diagram is never evaluated explicitly - the system simply remembers the correspondence of commands to response trajectories and inverts this list of ordered pairs. In this manner, alternatives are considered in actuator terms, and the *navigator plans in actuation space*.

Note also that **state space** serves as an intermediate between actuation space and **hazard space** whereas the classical mapping is more direct. It is clear from the figure that state space is, in fact, a superset of configuration space - including all C space variables plus any derivatives that appear in the system dynamic model. Like C space, hazard space abstracts the vehicle to a point, but that point executes a trajectory as the vehicle moves. This is because hazard is associated with the moving vehicle instead of the stationary environment.

The essential computational efficiency argument of the approach is:

- Motion planning is a problem involving search.
- Efficiency of search can be improved with constraint ordering heuristics.
- Forward modelling, which considers dynamic modelling before obstacle avoidance, is a valuable ordering heuristic which improves both computational efficiency and system robustness.



Section 2: Obstacle Avoidance

Several hazardous conditions can be evaluated at each point along a trajectory. Some typical hazards are:

- Tipover: The movement of the weight vector outside of the support polygon formed from the wheel contact points.
- Body Collision: Collision of the underbody with the terrain.
- Discrete Obstacles: Regions of locally high terrain gradient.
- Unknown Terrain: Regions that are occluded, outside the field of view of the sensors, unknown from poor measurement accuracy or resolution, or devoid of matter (such as the region beyond a cliff edge).

As an example of a hazard signal, a tipover hazard signal can be split into two signals - for pitch and roll. The pitch signal is then given by:

$$y_{\text{pitch}}(t) = \frac{|z_{\text{front}}(t) - z_{\text{rear}}(t)| / \text{wheelbase}}{\max}$$

Each of these hazards can be computed over time and a normalized signal can be prepared whose amplitude is limited to values between 0 and 1 where 0 represents complete safety and 1 is complete system failure. This normalization permits the navigator to compare different hazards in a consistent system of units. On normalized hazard vectors, the distance between points can have meaning if the normalization is performed carefully.

In order to make a decision based on these signals, it is necessary first to integrate out the time dimension and second to merge all of the hazard predictions together. The second step generates a holistic estimate of the degree of safety expected if the command is executed. In general, this can be performed by some measure of distance in hazard space such as:

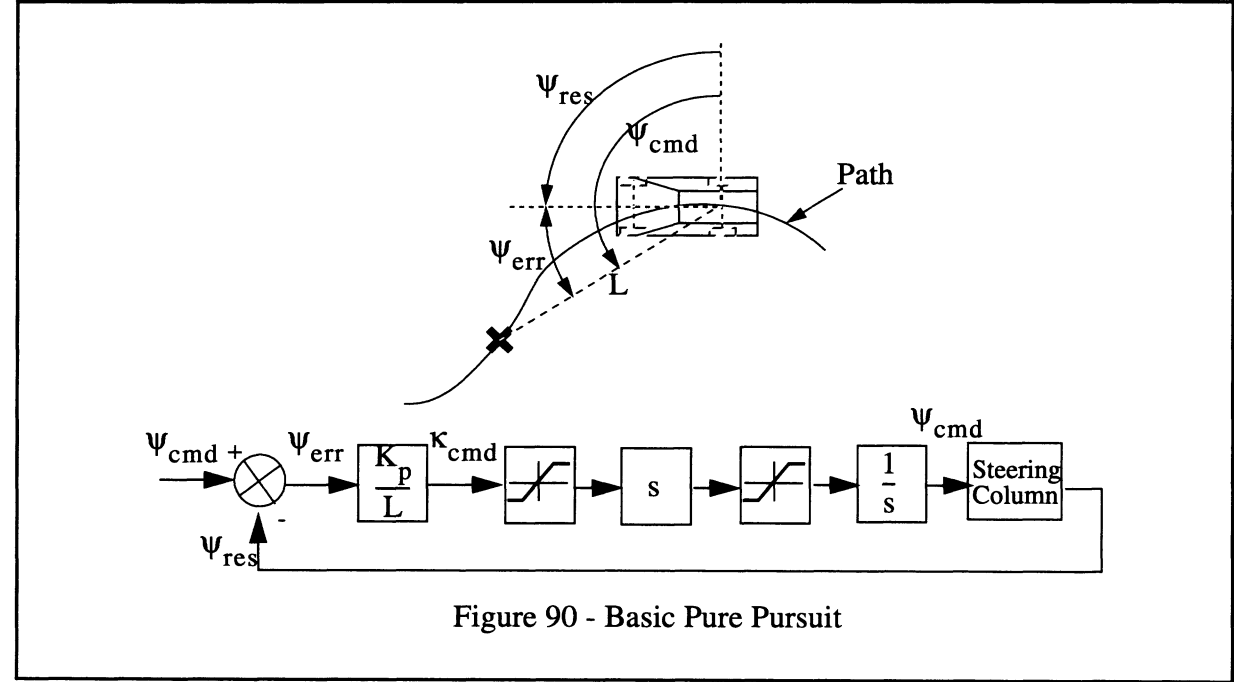
$$L(\underline{y}) = \left[\sum_j (y_j)^\alpha \right]^{1/\alpha}$$

In practice, a simple maximum over the dimensions of hazard space performed acceptably in system tests. The output of this process is the hazard vector which is supplied to the optimal control arbiter. Detailed hazard kinematics will be given in a later section.

Section 3: Goal-Seeking

Several forms of strategic goals can be supported within the optimal control context of the planner design. The only nontrivial aspect of tracking a strategic goal is high-fidelity tracking of a convoluted path over rough terrain. Other types of goals such as goal headings, goal points, goal curvatures, etc. can be extracted as trivial subcases of the more general path tracking problem discussed below. Although the stated algorithms are given for a conventional automobile, they can be extended to other kinds of vehicles with little changes because it is normally possible to express a trajectory in terms of speed and curvature independent of the underlying steering and propulsion kinematics.

Goal-seeking incorporates an adaptive path tracker which is based on the **pure pursuit** algorithm. The pure pursuit algorithm has been around for some time [74]. It is basically a proportional controller formed on the heading error computed from the current heading and the heading derived from the current vehicle position and a goal point on the path. The goal point is computed by finding the point on the path which is a predetermined distance from the current vehicle position. There are many variations possible on the basic algorithm.



Heading is measured at the center of the rear axle. The proportional gain is normalized by the lookahead distance L . This can be viewed as an adaptive element or, more simply, as a unit conversion from heading error to curvature. Indeed, the ratio ψ_{err}/L is the average curvature required to reacquire the path at the goal point.

A limiter is used which ensures that the curvature is limited at higher speeds to prevent rollover. Another limiter maintains the angular acceleration below a threshold. These measures ensure vehicle safety and directly prevent instability at the cost of an inability to track high curvature paths at high speed. However, vehicle dynamics already prevent tracking high curvature paths at high speed so generality is not lost.

Chapter 2: Design

This section provides detailed algorithm descriptions for the planning and control subsystem.

Section 1: Optimal Control

The system attempts to satisfy two goals simultaneously. The **tactical goal** is to avoid obstacles while the **strategic goal** is to track the goal trajectory. At times, goal-seeking may cause collision with obstacles because, for example, the goal may be behind an obstacle. The system incorporates an arbiter which permits obstacle avoidance and goal-seeking to coexist and to simultaneously influence the behavior of the host vehicle. The arbiter can also integrate the commands of a human driver with the autonomous system. The arbitration mechanism is the implementation equivalent of the optimal controller specified in the concept section.

1.1 Goal Arbitration

The mechanism used for merging the conflicting goals of obstacle avoidance and goal-seeking is to form a vote vector for each goal and to choose the command signal which is closest to the strategic vote maximum while still remaining acceptable to obstacle avoidance. The strategic goal is used to bias obstacle avoidance when there are a number of alternatives, and obstacle avoidance wrests absolute control from the strategic goal seeker when it is necessary to do so.

Several times a second, the planner considers approximately ten steering angles to use during the next control cycle. The forward model simulates the effect of using these steering angles over a short period of time and evaluates each of the resultant paths. Any steering angles that result in paths that go near or through hazardous vehicle configurations (determined from an elevation map) are discarded. The steering angle that results in a path that both satisfies obstacle avoidance constraints and optimizes the cost function is chosen.

The output of the steering arbiter may be very discontinuous if a hazard suddenly appears or disappears and because high steering rates constitute a hazard by themselves, the steering arbiter output is smoothed to remove steering discontinuities except when drastic action is called for. That is, the strategic steering output is smoothed whereas the tactical one is not.

This approach to arbitration has the side effect that following of an extended feature such as a cliff edge or roadside hill will emerge naturally, and the system will immediately take the opportunity to reacquire the strategic goal if an opportunity presents itself. While this describes the emergent behavior, the algorithm itself is to:

- sort the votes of the tactical planner using the corresponding strategic votes as the key
- choose the first path in the sorted list which satisfies obstacle avoidance

If obstacle avoidance does not care which direction to choose, the strategic planner is given control. If it does care, it will continually veto the strategic votes in order of strategic preference until a safe one is presented to it. If no safe path exists, a stop command is issued.

1.2 State Space Model

A high fidelity feedforward actuator dynamics and terrain following model is introduced here. The forward vehicle model is mathematically involved because it involves solution of a coupled set of eight nonlinear differential equations which form the vehicle state space model. The system mechanizes these equations in real time. The equations are coupled because:

- position at step i depends on attitude and speed at step $i-1$
- attitude at step i depends on steering response, propulsion response, and attitude step $i-1$
- steering response at step i depends on steering response and steering command at step $i-1$
- propulsion response at step i depends on attitude, propulsion response and propulsion command at step $i-1$
- attitude at step i depends on suspension state at step $i-1$

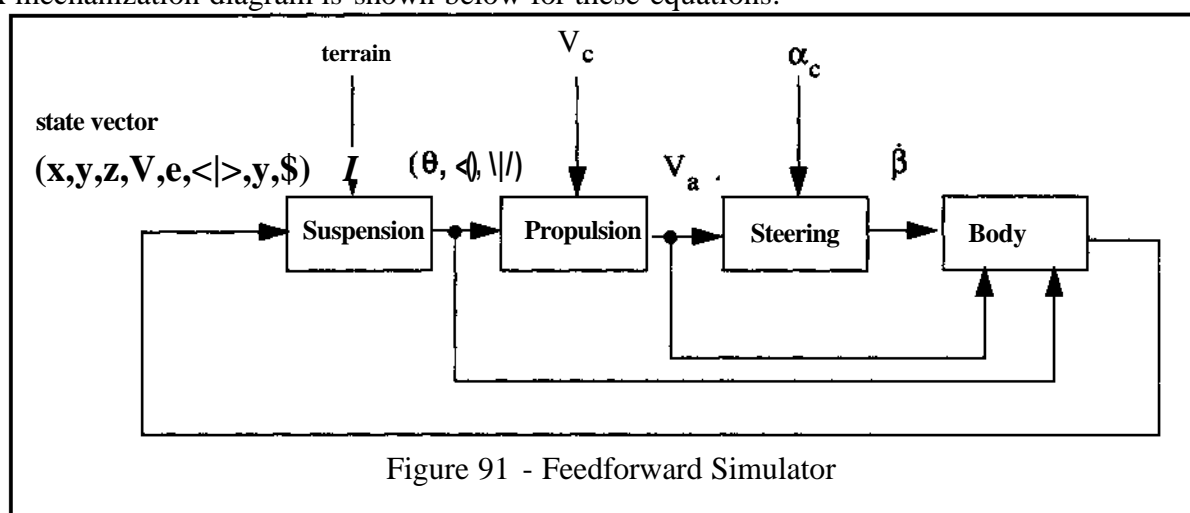
1.2.1 State Space Terrain Following Model

By solving the equations in their true coupled form, the system can correctly simulate a vehicle driving along the side of a hill, for example. It correctly simulates the dependence of actual trajectory on the terrain itself, the speed, the actuator response characteristics, and the initial conditions.

The basic simulation loop can be written as follows. At each time step:

- simulate suspension - determine attitude from terrain geometry and position
- simulate propulsion - determine new speed from command, state, and attitude
- simulate steering - determine angular velocity from steering and speed
- simulate body - dead reckon from linear and angular velocity and time step

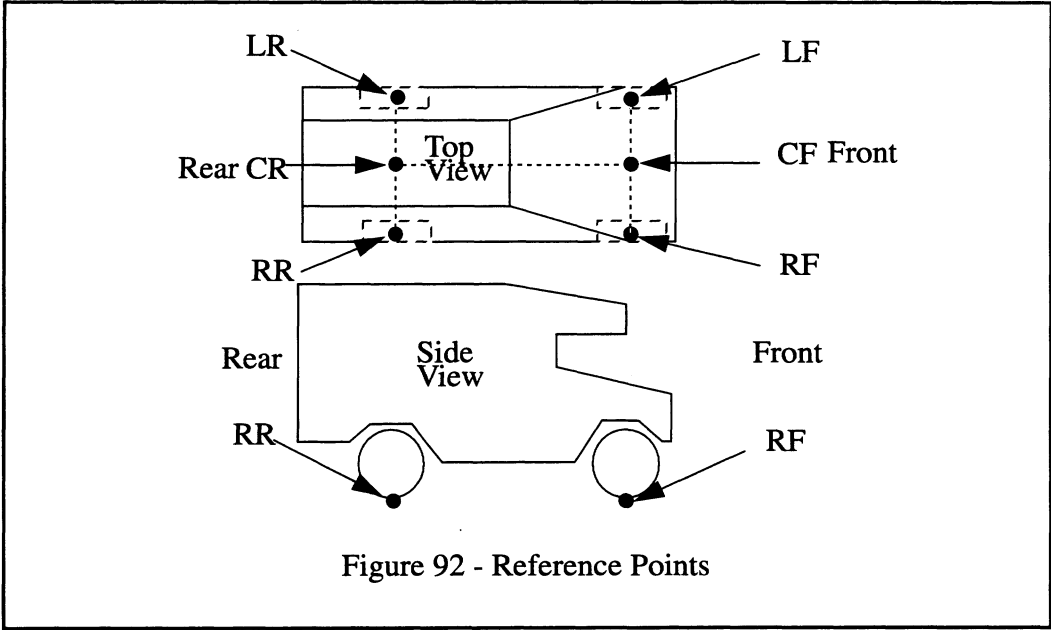
A mechanization diagram is shown below for these equations:



In the figure, (x, y, z) represents the position of the vehicle in the navigation frame; (θ, ϕ, ψ) represents its attitude in terms of pitch, roll, and yaw angles respectively, V is the vehicle speed along the body y axis; α is the steer angle of the front wheels; $\dot{\mathbf{p}}$ is the angular velocity of the body projected onto the body z axis; and the subscripts c and a represent commanded and actual quantities respectively.

1.2.2 Reference Points

The positions of distinguished points on the body, called **reference points**, are maintained in navigation coordinates throughout the simulation. The kinematics transforms involved in doing this are documented in the appendices. These points are at the same elevation as the bottom of the wheels and are positioned at the wheel contact points or the center of the axles. The processing of the reference points consists of converting coordinates from the body to nav frame based on the vehicle position and attitude for each cycle of the simulation.



All aspects of both system state and hazardous conditions can be computed from the reference points and the terrain elevation under them. In this way, coordinate system transformation operations are reduced to an absolute minimum.

1.2.3 Suspension Model

Using the reference points only, a proper model of a flexible vehicle suspension would:

- compute the elevation “under” each wheel
- compute the deflections of each wheel from the minimum potential energy principle [76]
- compute the positions of three points on the body
- compute the body attitude from these three points

This is a lot of somewhat costly work to do. These computations are considered too costly so the suspension model that is actually used is based on:

- a **rigid terrain assumption**
- a **rigid suspension assumption**
- a **locally-planar terrain assumption**

The first assumption results from using the terrain map elevations measured by a sensor when the vehicle was not loading the terrain. The second occurs because the positions of the wheels relative to the body are taken as fixed. The third occurs because the elevations under the four wheels are used in forming two vectors even though they do not necessarily lie on the same plane.

The last problem in the process can be expressed as that of recovering an unknown rotation matrix from a few points which are transformed using it. The inverse RPY transform is computed in the appendices for this purpose. The displacement transform from body coordinates to world coordinates for a z-x-y Euler angle sequence can be expressed as:

$$\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} (C\psi/C\theta) - S\psi/S\theta S(\phi) & -s\psi/c\theta & (C\psi/S(\theta + S\psi)/S\theta C(\phi)) \\ (S\psi/C(\theta) + C\psi/S\theta S(\phi)) & c\psi/c\theta & (S\psi/S(\theta - C\psi)/S\theta C(\phi)) \\ -c\theta s(\phi) & s\theta & c\theta c(\phi) \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}$$

Yaw and pitch can be determined from any vector known to be aligned with the body y axis:

$$\psi = atan2(r_{22}, -r_{12})$$

$$\theta = atan2(r_{32}, -r_{12}s\psi + r_{22}c\psi)$$

Roll can be derived from the world coordinates of a vector known to be aligned with the body x axis.

$$\phi = atan2(s\theta[-r_{11}s\psi + r_{21}c\psi] - r_{31}c\theta, (r_{11}c\psi + r_{21}s\psi))$$

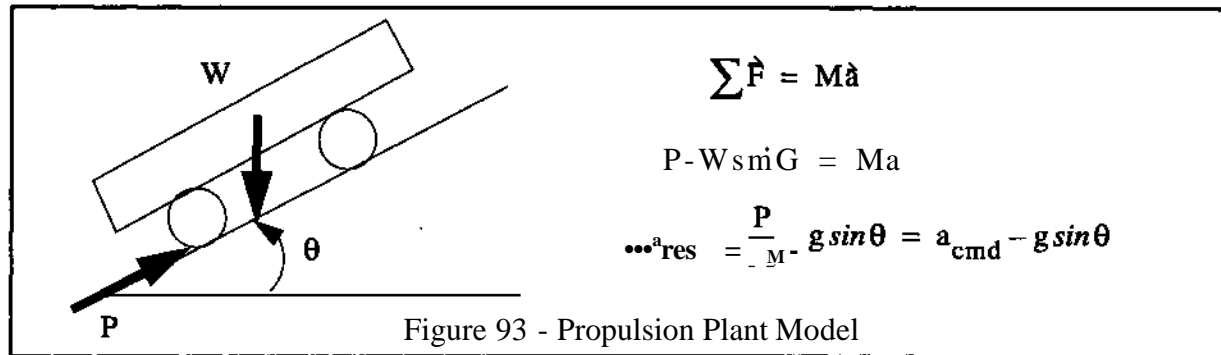
The above equations are an exact inverse kinematic solution. However, the system makes a valid **small pitch assumption** currently in order to reduce the trigonometric computations. The vectors are formed from the reference points. The algorithm is simply:

```

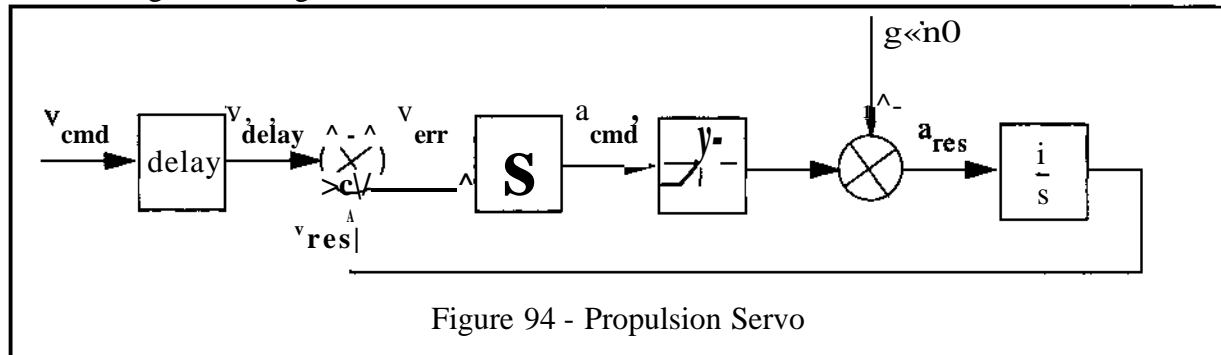
/*
** Simulate Suspension
*/
zleft == (df[z] -f lr[z]) / 2.0;
zrght == (rf[z] -f rr[z]) / 2.0;
zfrnt == (rf[z] -f lf[z]) / 2.0;
zrear == (rr[z] -f lr[z]) / 2.0;
z = zrear + tire_radius;
pitch == atan2(zfrnt - zrear)/wheelbase;
roll = atan2(zleft - zrght)/width;
```


1.2.4 Propulsion Model

Propulsion is modelled as a proportional controller with gravity compensation⁴. Under an assumption that the torque at the wheels can be directly actuated and that the terrain can generate any necessary shear reaction (no slip) Newton's law can be written along the body y axis as follows:



which gives the plant model. A proportional controller using velocity feedback is configured as in the following block diagram⁵:



The inverse of the proportional gain K_p is the time constant. A time constant of 5 seconds corresponds to the generation of 0.1 g command for a 5 m/s speed error. A limiter is added to reflect the fact that the output torque has some limit. One way to compute this limit is to choose a pitch angle corresponding to some grade that is the highest grade that the vehicle can climb. Under this model, the computer algorithm is the finite difference version of the system differential equation, which is simply⁶:

```

/*
** Simulate Propulsion
*/
now = read_clock();
vdelay = queue_lookup(prop_queue, now - prop_delay);
vterr = vdelay - vres;
acmd = vterr / prop_time_constant;
if (fabs(acmd) > amax) acmd = amax * acmd / fabs(acmd);
ares = acmd - g * sin(pitch);
vres += ares * dt;

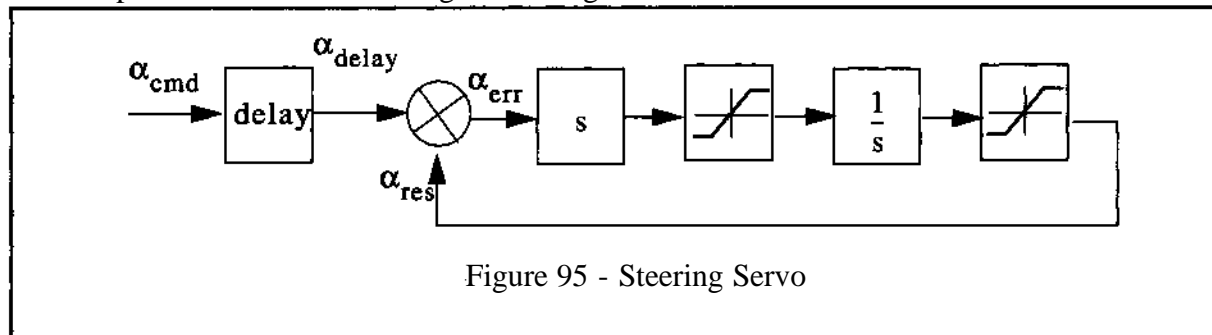
```

4. This model is similar to the JPL speed controller except that throttle is actuated, not torque.
5. If, like most of us, the reader has forgotten Laplace Transforms, $1/s$ is an integral and s is a derivative.

This model has several advantages over an ideal one. It will correctly cause the simulated vehicle to take several seconds to respond to a step input command and it will even cause the simulated vehicle to back up on a steep grade and generally slow down or speed up depending on the grade.

1.2.5 Steering Model

The steering model is involved because it must account for the nonlinear relationship between curvature and steer angle. This model is also a primary area of coupling between speed and attitude rate, so it is one of the most important elements of the model. The bicycle model of the steering kinematics is given in the appendices. Typically, the steering wheel itself is actuated, so the loop must be closed back at the steer angle. This assumes only that the steering wheel to steer angle transform is linear. Either position or speed could be the controlled variable, but a proportional position loop is assumed⁷. Both the position and the speed are limited⁸. The complete dynamics can be expressed with the following block diagram:



Again the coding is straightforward:

```

/*
** Simulate Steering Dynamics
*/
now = read_clock();
alpdelay = queue_lookup(steer_queue, now - steer_delay);
alperr = alpdelay - alpres;
if (fabs(alperr/dt) > alpdotmax)
    alperr = alpdotmax * dt * alperr / fabs(alperr);
ares += alperr;

```

These few lines of code are responsible for the high-speed stability of the navigator.

6. If we distinguish control algorithms from complete controllers, many actuator control algorithms can be implemented in just a few lines of code. The above situation is the rule, not the exception. Code is provided to illustrate that however sophisticated dynamic models may sound, implementing simple ones is easy.

7. At this moment, a position loop is used because a speed loop requires speed feedback which is not available on some vehicles. It could be generated internally by differentiating the position feedback, but the speed limit is considered to be the overriding element of the delay anyway.

8. There may also be a trapezoidal command velocity profile. This amounts to an acceleration limit on the command.

1.2.6 Body Dynamics

The 3D dead reckoning equations also form the system model in the Kalman filter which is documented in Part VI: Position Estimation. The attitude rate about the body y axis is available from the instantaneous velocity and the instantaneous steer angle (or curvature) as follows:

$$\frac{d\beta(t)}{dt} = \frac{1}{L} \tan[\alpha(t)] \frac{ds}{dt} = \kappa(t) \frac{ds}{dt}$$

1.3 Discrete Time Nonlinear Model

The last section gave a linear systems model of a MIMO system. The system model actually used is a nonlinear model, but it can still be expressed as a set of matrix equations if the matrices themselves are allowed to vary with the state vector.

1.3.1 State Vector

The **state vector** includes the position and attitude of the body and the linear velocity along body y and the angular velocity around body z:

$$\bar{x} = [x \ y \ z \ V \ \theta \ \phi \ \psi \ \dot{\beta}]^T$$

1.3.2 Command Vector

The **command vector** includes the commanded speed and commanded "yawrate":

$$\bar{u} = [V_c \ \dot{\beta}_c]^T$$

1.3.3 System Model

The system model is based on a **low dynamics assumption**. This is the assumption that velocity can be considered constant for a small period of time. The model is given below:

$$\begin{bmatrix} x \\ y \\ z \\ V \\ \theta \\ \phi \\ \psi \\ \dot{\beta} \end{bmatrix}_{K+1} = \begin{bmatrix} x \\ y \\ z \\ V \\ \theta \\ \phi \\ \psi \\ \dot{\beta} \end{bmatrix}_K + \begin{bmatrix} -V_{sync8} \\ V_{cv}/c9 \\ V_{s9} \\ 0 \\ \dot{p}^{s(t)}_{t9} \\ -P_{t9c<|} \\ \dot{p}_{c(t)/c9} \\ 0 \end{bmatrix}_K dt$$

The terms in the bottom of the vector account for the nonlinear dependence of the attitude rate on the attitude of the body. Angular velocity kinematics are given in the appendices. This model will correctly account for motion out of the plane - correctly turning the vehicle around the body z axis instead of the gravity vector. This aspect of the model can be expressed as transition matrix, if its elements are allowed to vary with the state vector.

1.3.4 Forcing Function and Ackerman Kinematics

The relationship between the yawrate and the steer angle is:

$$\dot{\beta} = \kappa \mathbf{V} = \frac{\tan \alpha}{L} \mathbf{V}$$

If we ignore the nonlinear limiters for the purpose of illustration, and model the actuator loops as first order systems, the relationship between the command vector and the state derivative vector can be written as:

$$\begin{bmatrix} \dot{\mathbf{V}} \\ \dot{\beta} \end{bmatrix}_{K+1} = \begin{bmatrix} \mathbf{V} \\ \mathbf{A} \end{bmatrix}_K + \left[\begin{array}{c} \kappa_{\text{prop}} (\mathbf{v}_{\text{cmd}} - \mathbf{V}) \\ \frac{\tan f_{\text{steer}}^K (\mathbf{a}_{\text{cmd}} - \mathbf{a})}{L} \end{array} \right] dt$$

1.3.5 Terrain Contact Constraint

Under a **small pitch assumption**, the terrain contact constraint alters the attitude each cycle. It can be written as:

$$\begin{bmatrix} \theta \\ \mathbf{W} \end{bmatrix}_{K+1} = \frac{1}{2} \begin{bmatrix} \frac{1}{L} & \frac{1}{L} & -\frac{1}{L} & -\frac{1}{L} \\ \frac{1}{W} & -\frac{1}{W} & \frac{1}{W} & -\frac{1}{W} \end{bmatrix} \begin{bmatrix} z_{lf} \\ z_{rf} \\ z_{lr} \\ z_{rr} \end{bmatrix}$$

Section 2: Obstacle Avoidance

The system adopts the basic assumption that all hazardous conditions can be computed from the reference points. Hazards are represented as sampled time signals which are parameterized by the commands to which they correspond. At any point in forward projected time, the system is designed to be robust to the absence of information because the **sampling** and the **occlusion problems** are intrinsic and inevitable.

2.1 Temporal State Interpolation

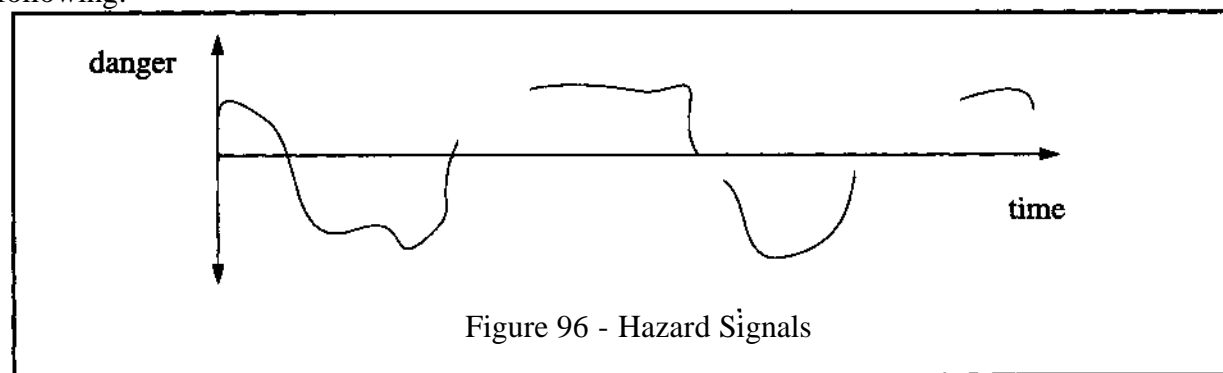
The interpolation algorithm used in the system is an interpolation of both the state vector and the hazard signals in time, rather than an interpolation of the terrain map in space.

Field testing of the system on typical rough terrain indicates that, *if occlusions are distinguished from poorly sampled regions* of the environment, the need for sophisticated terrain map interpolation is questionable. Specifically, the degree of undersampling depends on the local incidence angle of pixels to the terrain and the need for interpolation directly implies that the terrain is mostly flat anyway. The reverse argument is more compelling. If a small vertical step in the terrain exists, then it must span several pixels due to its near normal incidence. Therefore, it cannot be undersampled.

These observations imply that it is unnecessary to interpolate an entire terrain map before it is used for planning computations. It can be done *on the fly*, and more importantly, it can be done *on the vehicle state vector*. The state vector alone supplies the information necessary to propagate the system state forward in time, and the terrain contact constraint can be modelled as a "disturbance" which modifies the vehicle attitude whenever the terrain under the wheels is known (which, in practice, is almost always).

2.2 Hazard Representation

Each hazard is represented on a normalized scale where the maximum value indicates certainty of danger and the minimum value indicates certainty of safety. A typical hazard signal looks like the following:



2.3 Hazard Arbitration

The evaluation of hazards amounts to the generation of a 3D field of the form $merit(hazard,command,time)$ because safety is different for each command, different for each hazard, and a time signal. The obstacle avoidance hazard estimator computes a functional on this 3D field which forms the basis of the decision of whether the entire trajectory is to be considered safe. The first step of the process is to collapse the time dimension. This is done by computing the signal norm as follows:

$$merit = \left(\sum_i (merit(i))^\alpha \right)^{1/\alpha}$$

for some power α . In practice, however, a straightforward maximum was found to produce acceptable results. This step reduces the field to $merit(hazard,command)$.

The next step is to collapse the hazard dimension. Hazards are considered independent because, for example, body collision is completely irrelevant to static stability. Therefore, a straightforward maximum applied across this dimension gives the worst unsafe condition for all dimensions of safety considered provided all hazard elements are represented in some normalized unit system. This step reduces the field to $merit(command)$.

The final step is to reduce the command dimension, and this has been discussed in the context of joint arbitration of tactical and strategic votes. However, before the tactical vote vector is passed to the arbiter concerned, it is sometimes smoothed by **Gaussian filtering**⁹ in order to increase reliability. The principle of this approach is that since the environment exhibits a degree of smoothness, it is appropriate at times for poorly-rated trajectories to degrade the ratings of their spatial neighbors. In this manner, the impact of errors in both feedback and actuation will be less relevant because this smoothing mechanism will cause the vehicle to give obstacles a wide berth. At times, this mechanism is disabled depending on the spacing of the candidate trajectories investigated.

2.4 Hazard Estimation

Three principal forms of hazards are incorporated:

- unknown terrain
- collision of the terrain with the body
- static instability

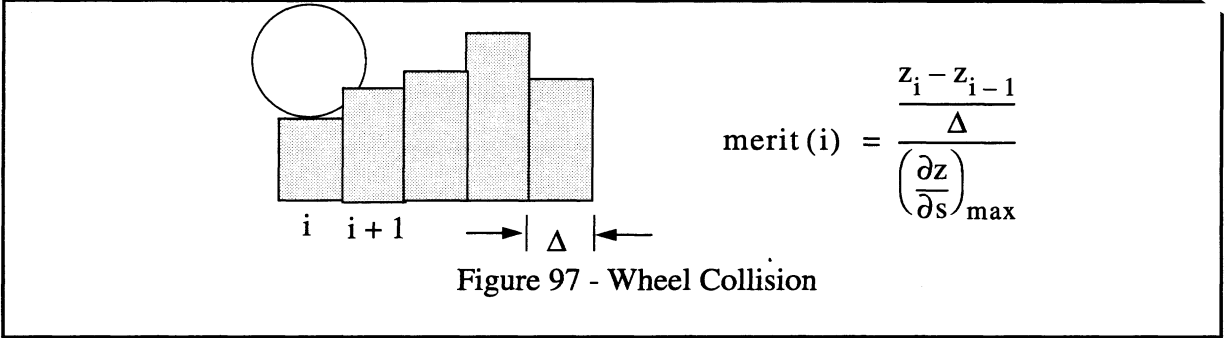
2.4.1 Unknown Terrain

Unknown map cells may arise from undersampling or terrain self-occlusion. An overall information density merit is computed for each candidate command which is based on the number of wheel elevations which are known at each time step.

9. Earlier, it was stated that only the strategic output is smoothed, but this was a discussion about temporal smoothing. The above smoothing is spatial.

2.4.2 Terrain-Wheel Collision

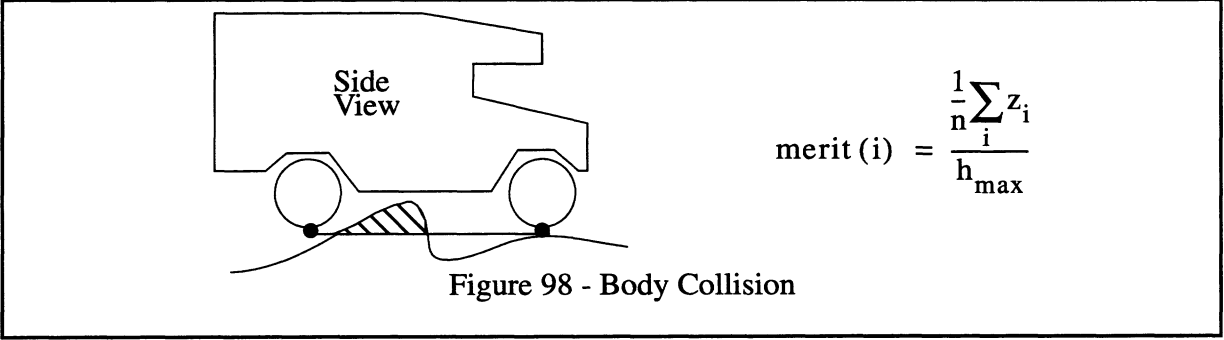
Wheel collision is computed as the slope of the elevations under each wheel as $t_{w,c}$ propagated forward.



In practice, this signal can be noisy if the sensor cannot supply the angular resolution and relative range accuracy necessary. Sometimes, it is not used at all.

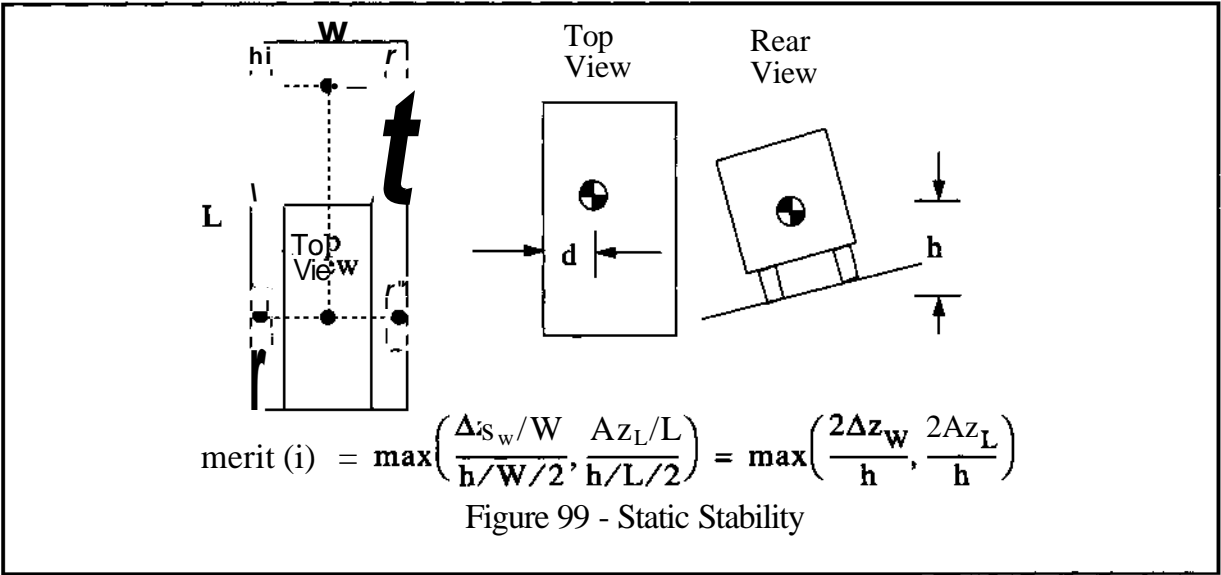
2.4.3 Terrain-Body Collision

Body collision is computed as the lost volume under the body normalized by the area under the body, so it is an expression of the average height under the body. The result is normalized by the number of map cells which were actually known.



2.4.4 Static Stability

Static stability measures the proximity of the vehicle to tipover. The **static stability margin** can be expressed as the remaining angle between the gravity vector and the support polygon formed by the wheel contact points (which is a rectangle in this case). Graphically, the margin is the minimum distance to any side of the support rectangle normalized by the height of the center of gravity above the terrain.



The margin can be expressed as the maximum of the tangent of simple approximations to roll and pitch normalized by some maximum. The distinction between measurement as an angle or its tangent is irrelevant for small angles¹⁰.

2.5 Matrix Representation

The hazard estimates can be expressed in matrix notation in order to relate them to the abstract system model described previously.

2.5.1 Hazard Vector

The hazard vector can be written as a 4 vector which includes the wheel collision, body collision, and stability margin:

$$\bar{y} = [\bar{y}_{wc} \quad \bar{y}_{bc} \quad \bar{y}_{sm}]^T$$

10. One of the implications of this model is that the vehicle will turn into a steep grade so as to increase the stability margin unless the strategic goal biases it away from the grade. The system actually computes pitch and roll margins individually, and performs the maximum outside the hazard evaluator. This is a measure used to permit easy debugging. The system can navigate successfully over smooth terrain based on this hazard alone.

The wheel collision output relationships are:

$$\overline{y_{wc}}|_K = \frac{1}{\Delta\left(\frac{\partial z}{\partial s}\right)_{\max}}\left(\begin{bmatrix} z_{lf} \\ z_{rf} \\ z_{lr} \\ z_{rr}|_K \end{bmatrix} - \begin{bmatrix} z_{lf} \\ z_{rf} \\ z_{lr} \\ z_{rr}|_{K-1} \end{bmatrix}\right)$$

The static stability output relationships are:

$$\overline{y_w}|_K = \frac{2}{h}\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}\begin{bmatrix} z_{lf} \\ z_{rf} \\ z_{lr} \\ z_{rr}|_K \end{bmatrix}$$

2.6 Hazard Uncertainty

Most hazards outlined above are computed as gradients of the terrain over some support. In such cases, the elevation uncertainty stored in the terrain map is used to compute an uncertainty on the hazard estimates as outlined in Part II: Analysis.

Section 3: Goal-Seeking

The basic pure pursuit algorithm specified in the concept section suffers from a few problems. Large tracking errors or, equivalently, too short a lookahead distance or too high a gain all result in servo instability. This is a well-known problem with pure pursuit which can be addressed with feedforward as discussed later.

3.1 Rough Terrain Pure Pursuit

A few modifications are introduced to adapt pure pursuit for rough terrain. First, extremely large tracking errors must be acceptable to the servo without causing instability. This is managed by two devices indicated in the following figure.

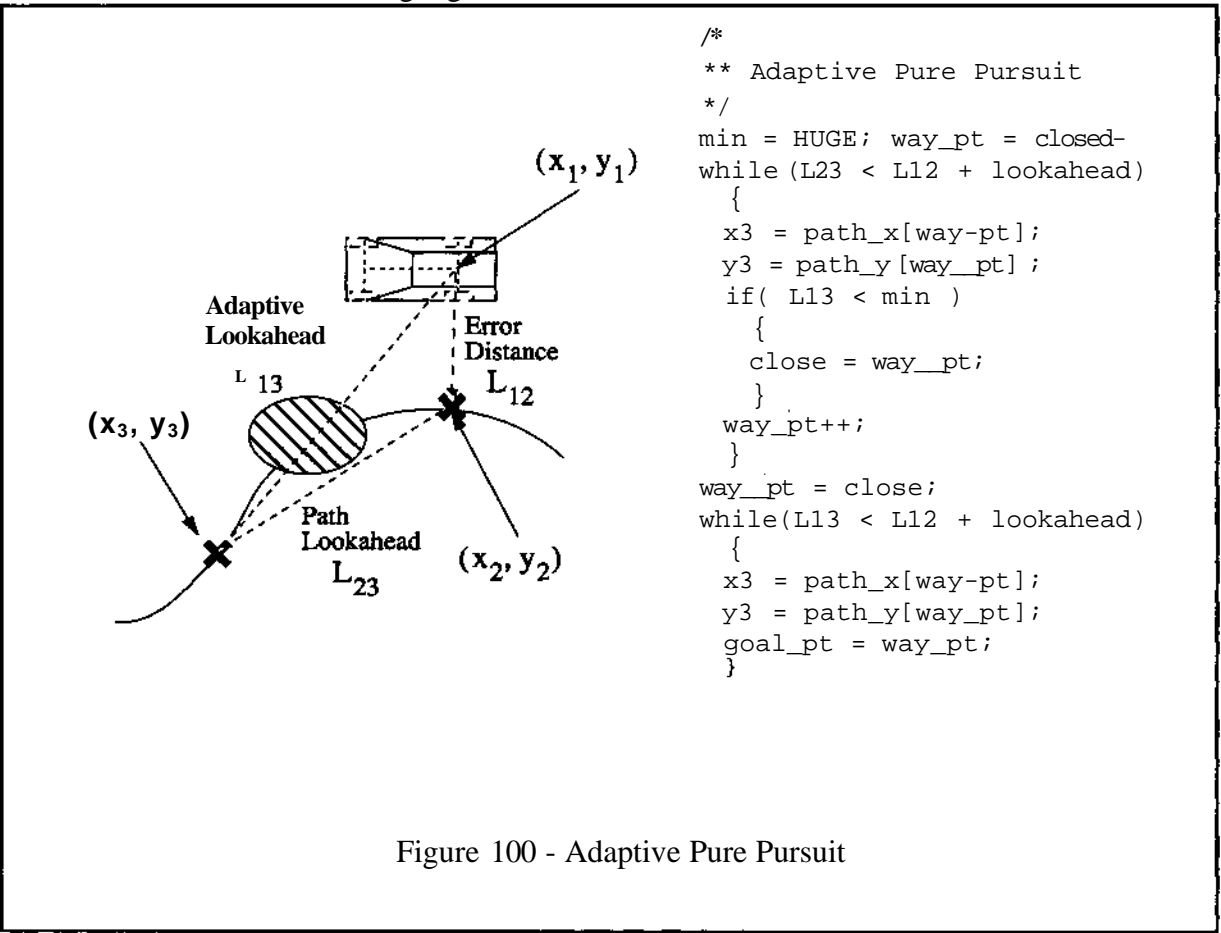


Figure 100 - Adaptive Pure Pursuit

Instead of using the current vehicle position as the origin of the lookahead vector, the system maintains a running estimate of the point on the path which is closest to the current vehicle position. This is done because it is very expensive to search the entire path each iteration. In doing so, the system is assuming that the vehicle will never have a heading error which exceeds 90° for an extended period of time. This is a **monotone arc length assumption**. This assumption completely eliminates the overwhelming computational cost of simpler implementations of the algorithm.

The lookahead distance is adaptive to the current tracking error - increasing as the error increases (indicated in the accompanying code fragment). The first while loop is responsible for maintaining a running record of the close point, point 2. It searches through an arc length window which adapts to the path tracking error. As the error gets larger, this loop will cause the close point to jump over high curvature kinks in the path as they become less relevant at the resolution of the tracking error.

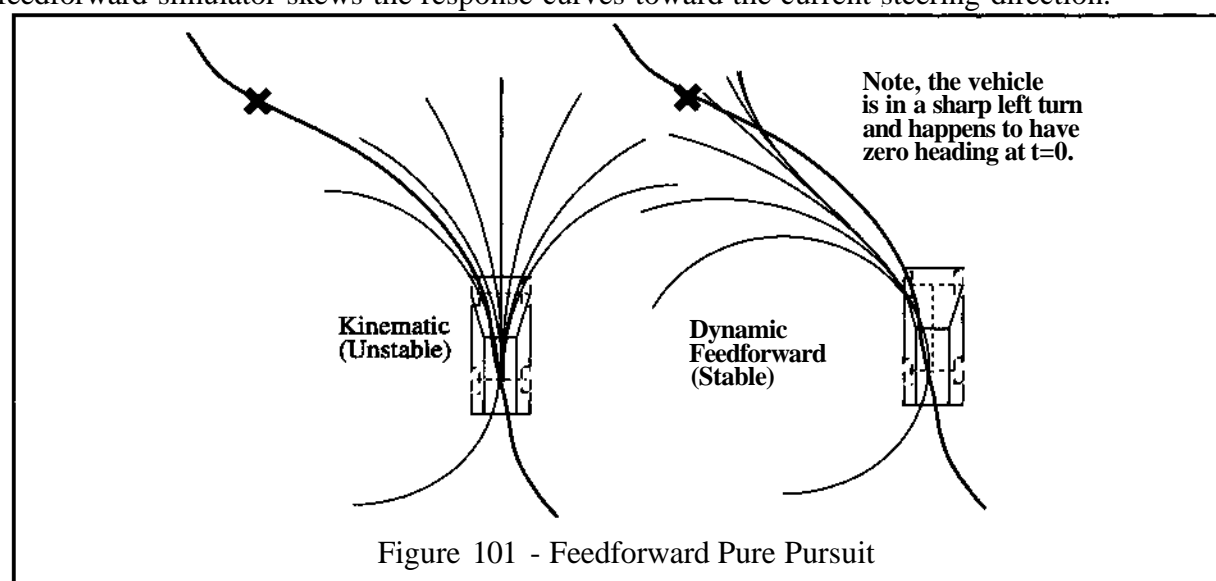
The second while loop computes the goal point in an identical manner. It basically moves point 3 forward until it falls outside a circle centered at the vehicle whose radius is the sum of the error distance and the nonadaptive lookahead. In this way, when an obstacle avoidance maneuver causes significant path error, the algorithm will search to reacquire the path on the other side of the obstacle instead of causing a return to the front of the obstacle.

Notice that under normal circumstances when the vehicle is tracking the path, the close point is at the vehicle position, the error distance is zero, and the adaptive lookahead is the nonadaptive lookahead. In plain terms, the algorithm gracefully degenerates to classical pure pursuit when obstacle avoidance is not necessary.

3.2 Feedforward Pure Pursuit

A feedforward option in the tracking algorithm incorporates the output of the system model simulator into the tracker. The cost of a feedforward simulator must be borne, so an additional feedforward element in the tracker is available for free. The basic idea is as follows. First, at each point in the simulation, evaluate the distance from the vehicle to the goal point. Second, the candidate command which comes closest to the goal point becomes the vote of the strategic controller. Such an algorithm provides excellent path following in three dimensions over rough terrain.

The concept is indicated in the following figure. During a high curvature turn at speed, the feedforward simulator skews the response curves toward the current steering direction.



A kinematic tracker would issue a hard right command in the situation depicted above whereas a dynamic one would recognize that such a command would actually *increase* the tracking error. A dynamic tracker would issue a zero curvature command and would correctly acquire the goal point as the steering wheel slowly turned toward zero.

Section 4: Adaptive Regard

In order to evaluate safety, the tactical control module computes the response to candidate commands simulated forward in time until the **detection zone** is reached. The spatial extent of the detection zone can be computed from the state space system model as the convex hull of the response trajectories for all considered command alternatives. However, there is no real need to compute this region explicitly - the only real requirement is to determine where it starts and where it ends in terms of distance from the vehicle position.

4.1 Planning Window

The **planning window** of tactical planning is analogous to the **range window** in perception. It confines the search for hazards to the detection zone. One of the highest level system requirements is to attempt to maintain continuous motion, so adaptive regard must be based on turning maneuvers (which consume more space) instead of braking maneuvers.

In the worst case, the vehicle will need to execute a very aggressive turning maneuver in order to avoid a large obstacle so the detection zone for this maneuver is used in the design. Specifically, the planning window is computed by predicting the *distance* required to execute an *impulse turn* at the current speed with the best available estimates of the output latencies that will apply. An impulse turn is a turn from zero curvature to the maximum allowed curvature.

Precision in computation of the planning window requires careful treatment of time as will be discussed in the next section. The planning window is measured from the vehicle position but because the vehicle is moving while all i/o and processing takes place, there are many candidates for this position, including:

- where the last image was taken
- where the last pose was measured
- where the vehicle is at the moment the planning window is computed
- where the vehicle will be when the steering command is issued
- where the vehicle will be when the steering actuator starts moving

Depending on which of these positions is chosen, a slightly different algorithm applies to the computation of the detection zone. The alternative used is the fourth one - the position of the vehicle when the steering command is issued by the system. In this case, the time required to turn the wheel to full deflection depends on the communications delay to the actuator and the dynamics of the actuator.

The algorithm for computing this window based on an impulse turn maneuver is given below. An idealized model of response is used in the sense that all transients are lumped into the turning reaction time. When it expires, the wheel is considered to switch curvature instantaneously. After the wheel moves, the vehicle still requires a distance equal to the turn radius to actually execute the turn.

Finally, in the worst case, an obstacle will exist just outside the planning window computed so far. If this is the case, the next cycle of the controller will be unable to avoid it because the vehicle will have moved a distance of $\text{speed} * \text{cycle_time}$ closer to it by the time the steering command is

recomputed. Therefore, this distance must be added to the current cycle's lookahead in order to be safe. This "extra cycle" is necessary regardless of the obstacle avoidance maneuver used because it arises from the finite cycle time of the system:

```
/*  
** Adaptive Regard  
*/  
treact = time to turn steering wheel to full deflection;  
rhomin = mininum safe turn radius;  
Pmax = speed * treact + rhomin;  
cycle_time = controller cycle time;  
speed = current speed;  
Pmax += speed*cycle_time;
```

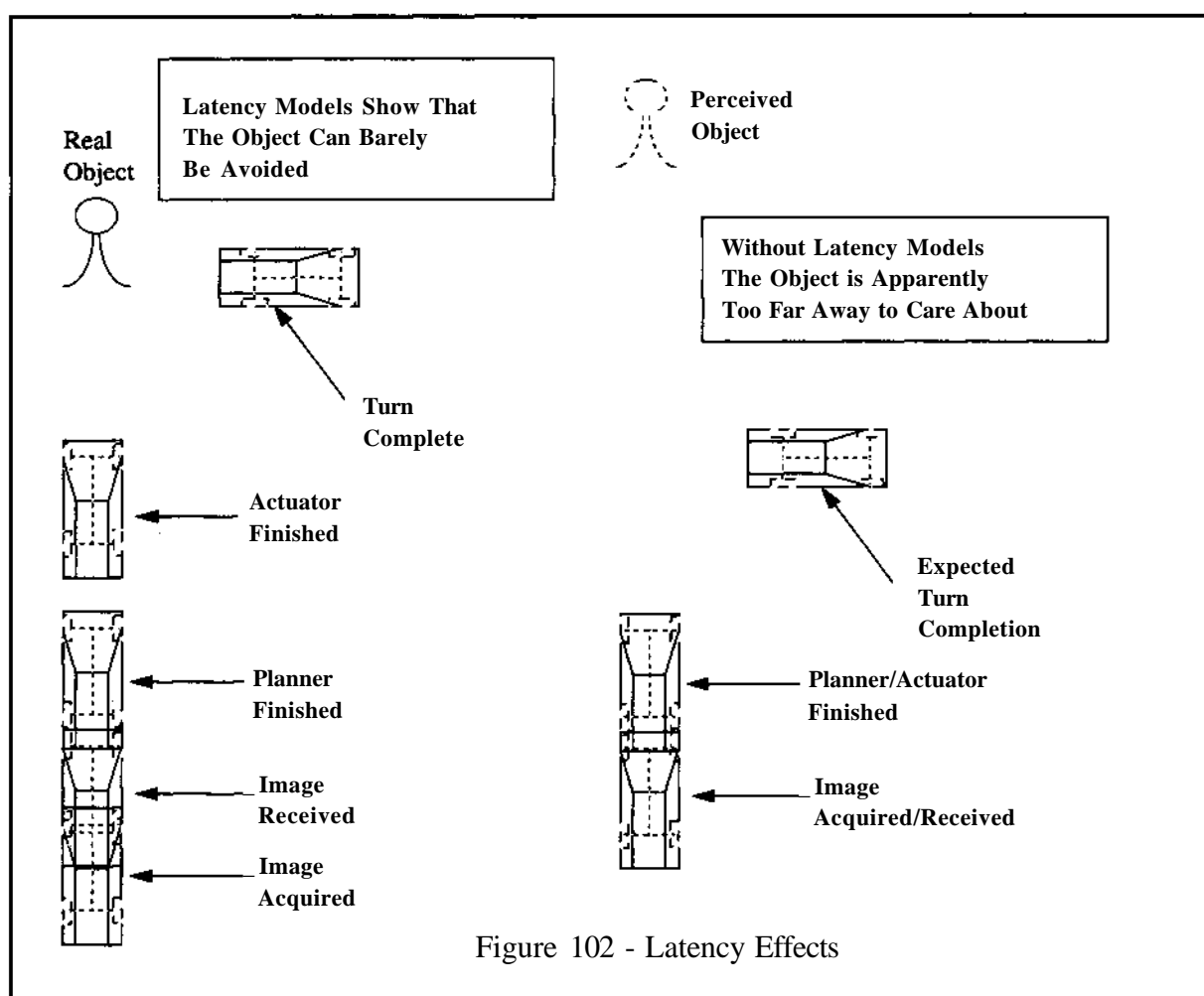
4.2 Real-Time Latency Modelling

It was shown in the perception part that failing to account for image input latency causes obstacles to be incorrectly positioned laterally by as much as 30° when turning and the distance to them to be overestimated by as much as 20 meters when moving at 20 m/sec. Further, failing to account for the scanning time of rangefinders can cause discrete objects to be smeared across 30° of the horizontal field of view.

The planning problem has similar latency concerns. While perception overestimates the distance to obstacles significantly, obstacle avoidance significantly underestimates the distance required to react for two reasons.

- Position estimate input latency implies that the vehicle is actually much closer than the last available position measurement suggests.
- Command output latency and actuator dynamics imply that it actually takes much more distance to turn than would be expected from instantaneous response models.

These issues are illustrated below for typical latencies and a moderate speed of 3 m/s:



A reliable measurement of the precise times at which events occur is *essential* for the system to function correctly. In particular, the system needs to match images with vehicle states which may arrive intermittently and it needs to remember the commands that were issued to all actuators in previous iterations of the main control loop because the system cycles faster than the output latencies. This is accomplished with the following mechanisms:

- all input and output is time-stamped
- all input and output is stored in internal FIFO queues
- all sensor latencies are modelled
- all actuator latencies are modelled

It is important to recognize that these FIFO queues do not introduce artificial delay. They are used to *model* the delays which already exist in the hardware and to register contemporary events in time.

The availability of a state space model makes it possible to model sensor and actuator latencies very precisely. For example, the instantaneous position of the vehicle at any point during the main processing loop can be computed by feeding the last known position forward over the terrain map based on the FIFO-queued commands issued in previous iterations. Some of the procedures involved are described below.

4.2.1 Localization of Image Pixels

The smearing of the environment model which arises from incorrect models of the motion of the perception sensor during the image digitization process is called the **motion distortion problem**. All incoming vehicle state information is stored in a FIFO queue until it is needed to process images which normally arrive much later. The state queue and the time tags are used to register the ray through every pixel with a unique vehicle pose before its value is converted to world coordinates.

4.2.2 Localization of the Vehicle

When the system uses the latest vehicle state as its estimate of position, the error involved in doing so is the velocity times the latency of vehicle state information. Clearly a delay of only one second causes 5 meters of localization error in the vehicle position at 5 m/s and pretty well guarantees collisions with obstacles solely because the system does not know how close it is to the obstacle. For this reason, an estimate of position sensor latency is used and fed forward just like any other parameter¹¹.

4.2.3 Localization of the Feedforward Vehicle

All output commands are stored in a FIFO queue and are used for the next several iterations in implementing command feedforward. When the system is generating the current command, there may be quite a few previous commands *still en route to the hardware*, so a proper model must feed these pending commands into the model before the current one is used and make decisions based on the last command to go out, because the system still has control over only that current command.

11. This is a practical measure because extant vehicle positioning systems and computing hardware have no consistent time standard. Ideally, the latency could be simply measured from the current time and the time tag of the state.

Section 5: Adaptive Regulators

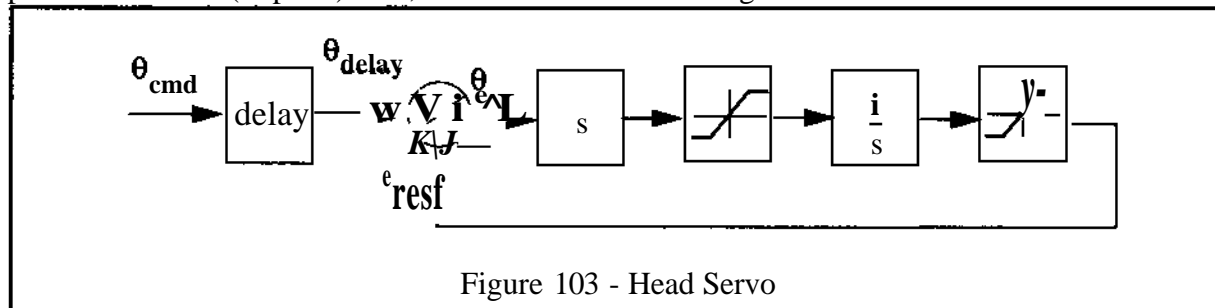
In cases where the vehicle infrastructure permits, adaptive regulation mechanisms are available to couple vehicle speed to its own estimates of reliability and to intelligently point the sensor field of view.

5.1 Sensor Stabilization and Pointing

Sensor head control algorithms are provided that incorporate both somatic and environmental feedback. These algorithms can also be used to control the field of view of real-time programmable sensors.

5.1.1 Head Model

A controllable pan/tilt sensor head is modelled by two differential equations which are similar to those used for the other actuators. It is likely that the system would not have access to speed feedback, so again, a position loop is used to model the dynamics. A loop which is identical to the steering model can be used to reflect the limited angles of excursion and the limited maximum speed. For the tilt (or pitch) axis, the servo would be configured as follows:



5.1.2 Azimuth Controllers

The pan axis *must move with the vehicle* and because it is mounted to the vehicle, any response on this axis is useful.

The first azimuth controller is the **steering regulator**. This controller generates pan commands which match the commands to the steering wheel. It would have been possible to servo to the steering feedback, but that option is considered to incorporate a delay and provide no benefit in return.

The second azimuth controller is the **detection zone tracker**. The steering feedforward simulator provides an intelligent basis for pointing the pan axis because the centroid of the region which is reachable by the vehicle can easily be computed. This controller computes the average angle subtended at the initial position by the response curve endpoint. In a sentence, this servo looks where the vehicle is going.

5.1.3 Elevation Controllers

The tilt axis *must move against the vehicle* and therefore must be able to respond significantly faster than the vehicle itself in order to be of any use.

The first elevation controller is the **pitch regulator**. This servo drives the elevation degree of freedom to the opposite of the vehicle pitch. Note that if the vehicle pitch is indicated by an inertial sensor, *the servo gives effective inertial stabilization of the head without the cost of extra gyroscopes*.

The second elevation controller is the **range window tracker**. In rough terrain, the field of view can only be pointed by a loop closed around the perception system because the body attitude and terrain roughness affect the projection of the field of view onto the groundplane and only the high level control loop knows this information.

The adaptive sweep algorithm provides an excellent measure of the deviation of the sensor tilt from the ideal. This is computed as the average deviation of the range window from the vertical center of the image. This error signal is used to drive the head, and in the process, a rough-terrain adaptive sensor head controller results. This controller provides impressive stabilization on rough terrain and it can also drive a roll axis with little modification.

5.2 Speed Planning

The speed which should be commanded of the vehicle depends on the local environment. It makes sense, for instance, for the vehicle to slow down when it encounters challenging terrain and likewise increase speed when benign terrain is encountered.

The motivation for speed planning is there is reason to expect that:

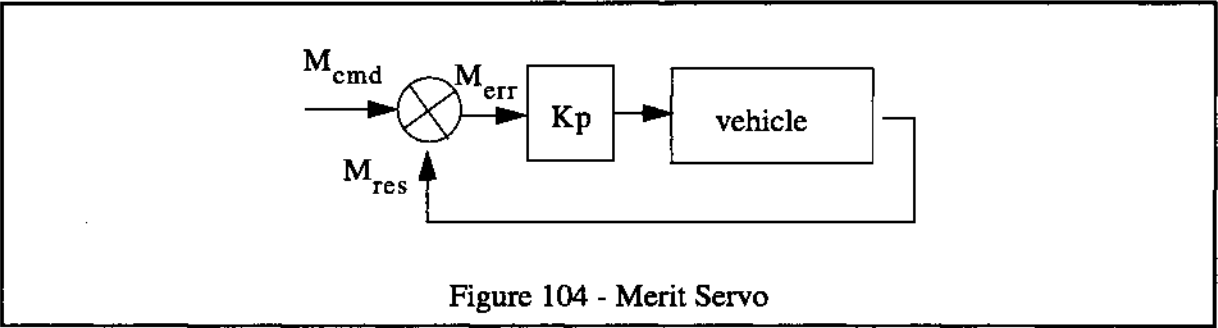
- there is a way to measure system robustness
- there is a way to relate robustness to speed in rough terms

The principle used for a speed planner is to close a proportional control loop around the average predicted merit of all candidate commands because this average merit is low in dense hazard environments and high in regions free of hazards. This principle is complemented by the fact that higher speeds tend to lead to reduced maneuverability and hence lower reliability for the same local terrain. That is, robustness can be altered in a small region by adjusting speed alone.

Many aspects of performance are implicitly figured in this algorithm. For example, occlusion, resolution, map quantization noise, etc. are all implicitly expressed in the hazard signals. The system will eventually reach equilibrium with respect to the most constraining sensor characteristic and an appropriate speed will emerge naturally.

When a planner gets confused, it tends to stay confused for a time and when it is relatively stable, it also tends to stay stable. This implies that there is a low frequency component to system robustness which is long enough in time constant to exceed the vehicle response limits. Hence, a vehicle may be able to respond fast enough to change things before it's too late. It should also be noted that the tactical controller is designed so that *a panic stop (slamming on the brakes) is always guaranteed to avoid vehicle damage*. The speed planner uses more graceful measures to improve performance.

A simple loop is configured as follows:



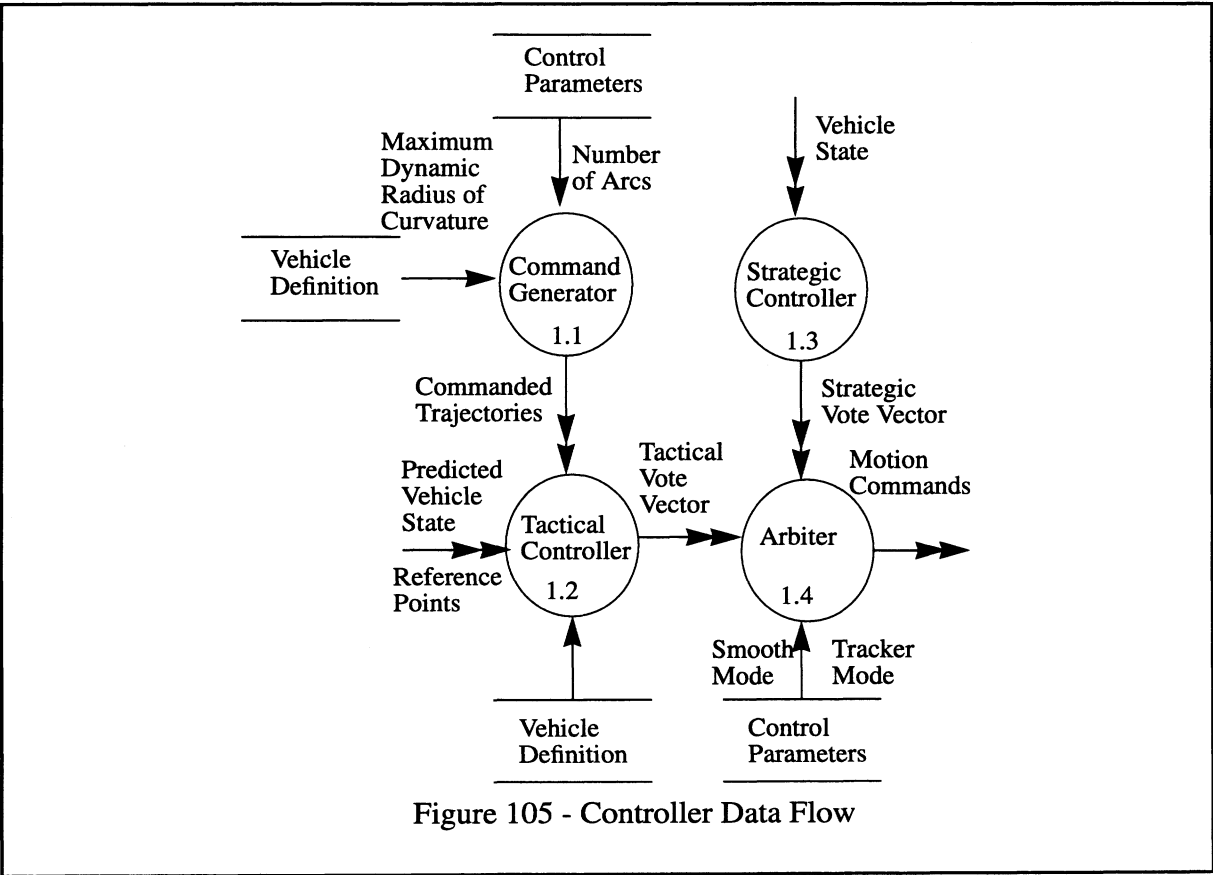
This loop is implemented with a time constant that is relatively large so that the speed loop will respond only to the low frequencies in the overall merit.

Chapter 3: Implementation

This section overviews the implementation of the planning and control system. This system is implemented as two main objects - the vehicle and the controller. The vehicle encapsulates vehicle state information and maintains the time histories of both measured states and issued commands. It also implements the state space model used for implementing feedforward. The controller consists of a tactical controller responsible for obstacle avoidance, a strategic controller responsible for goal-seeking, and an arbiter which implements the optimal control algorithm through a prioritized sort of the votes of the two controllers.

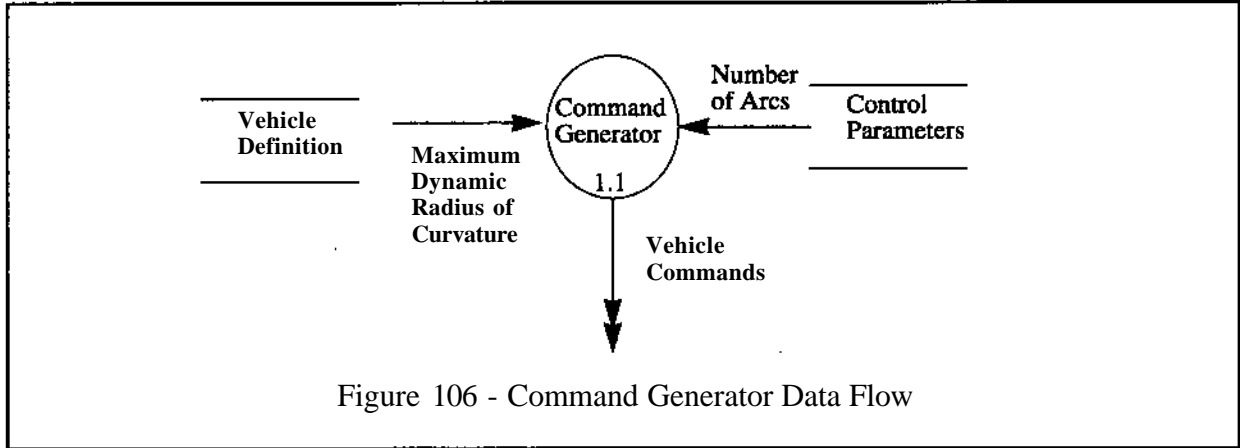
Section 1: Controller

Most aspects of the **control laws** (with the exception of feedforward itself) are implemented as controller modules as illustrated below:



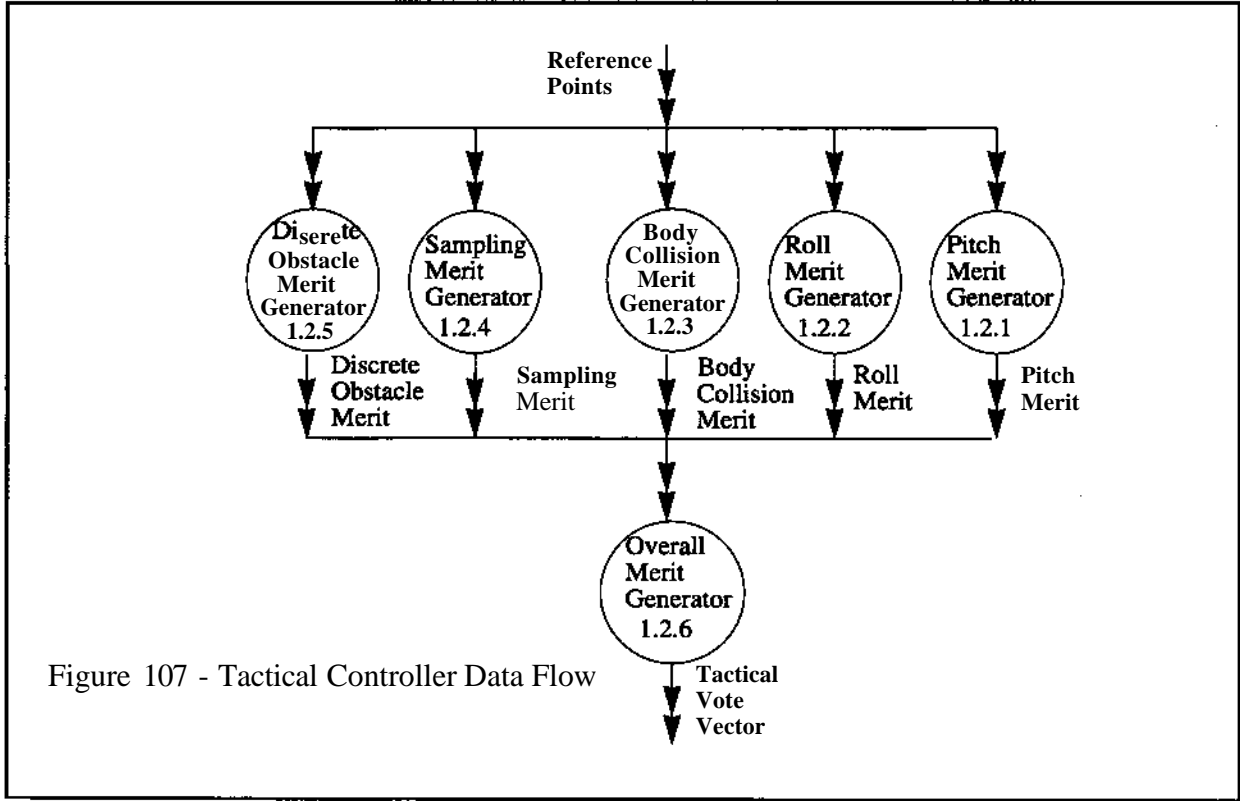
1.1 Command Generator

The **Command Generator** module uses knowledge of the vehicle kinematic steering constraints and the dynamics of rollover to compute a set of equally spaced command trajectories which are presented to the vehicle simulator for feedforward. The data flow for this module is given below:



1.2 Tactical Controller

The **Tactical Controller** is a module which attempts to maintain nonzero forward velocity while avoiding tipover, collision, or other hazardous situations. Based on the response state signals, the tactical controller forms a holistic noise-immune estimate of the overall merit of turning in each direction. Then it chooses the best alternative, and sends a command vote vector to the arbiter for integration with the strategic vote. The data flow diagram is given below.



The hazard signals, for some suitable hazard vector, encode all information necessary to assess safety in a minimalist manner. The hazard signals themselves are computed in this module from the reference point coordinates and the terrain under them.

The hazards computed are static stability, body collision, discrete obstacles, and the information density of the signals, called sampling. This module also stores, in signal form for graphical display purposes, a time history of the feedforward vehicle states used.

This module computes the optimal controller objective functional. There are a number of hazards considered, and each is a time signal, and each hazard time signal is computed for each candidate trajectory. The first two dimensions are integrated out in this module to produce the tactical vote vector which is passed to the arbiter for the final decision.

1.2.1 Pitch Merit Generator

This module computes the signal norm of the pitch turnover hazard signal for each curvature.

1.2.2 Roll Merit Generator

This module computes the signal norm of the roll turnover hazard signal for each curvature.

1.2.3 Body Collision Merit Generator

This module computes the signal norms for the body collision hazard signal for all reference points on the body for each curvature.

1.2.4 Sampling Merit Generator

It is considered very undesirable for the vehicle to drive onto unknown terrain. This module computes a norm of the sampling signal which is basically the degree to which terrain is unknown for sampling or occlusion reasons.

1.2.5 Discrete Obstacle Merit Generator

In order to maintain acceptable response characteristics, it is necessary to use a relatively low resolution terrain map. This implies that obstacles smaller than a map cell cannot be resolved from terrain map gradients. This module computes a norm of the discrete obstacle signal which is generated by applying an obstacle detector to the scatter matrix in each cell as discussed in [53].

1.2.6 Overall Merit Generator

This module computes the minimum of the merit factors of the above four hazards. In principle, the existence of any one of the above hazards is undesirable, so it is appropriate to rate the corresponding command by the worst of these.

1.3 Strategic Controller

The **Strategic Controller** module accepts a strategic goal which biases the system behavior in the long-term. The strategic goal may be to follow a heading, a fixed curvature, a single point, or a convoluted path. The path tracker component is the only nontrivial one. It accepts a specification of a path to be followed in global coordinates and generates steering preferences which would cause the vehicle to track the path. The tracking algorithm is a modification of **pure pursuit**. The output of this module is overridden by the arbiter if tracking the path would lead to a hazardous situation. The data flow diagram for the tracker module is given below.

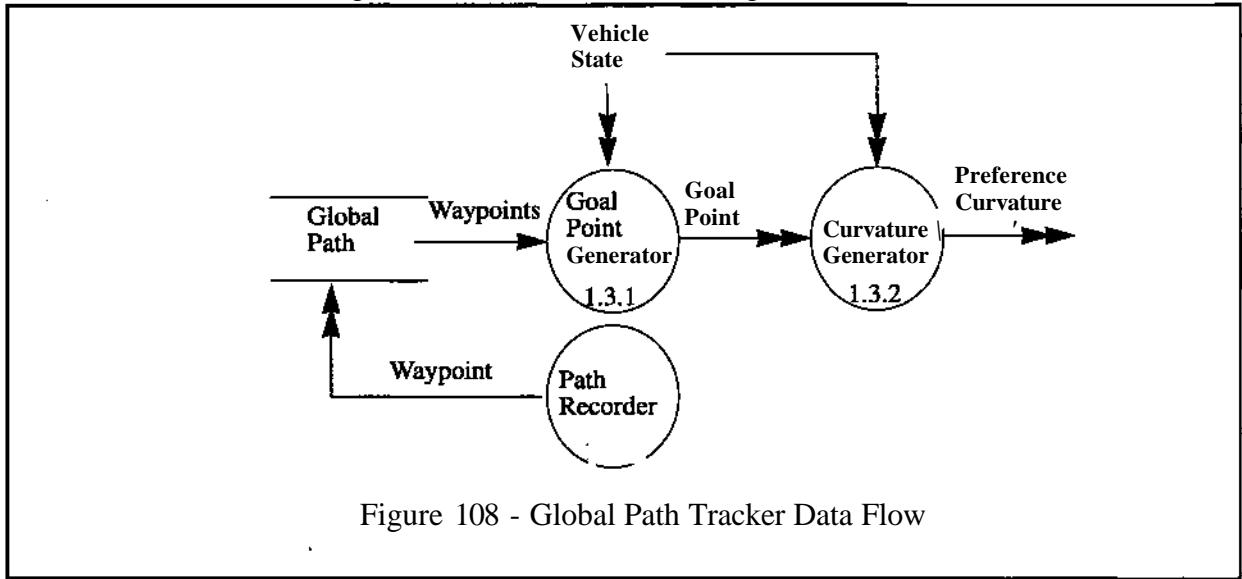


Figure 108 - Global Path Tracker Data Flow

1.3.1 Goal Point Generator

This module uses the current vehicle position and heading to determine the closest point on the path which is outside the pure pursuit **lookahead distance**. The lookahead distance is adaptive to the current tracking error - increasing as the error increases. In this way, when an obstacle avoidance maneuver causes significant path error, the algorithm will search to reacquire the path on the other side of the obstacle instead of causing a return to the front of the obstacle.

1.3.2 Curvature Generator

This module computes a preference curvature for the vehicle based on where it is now, and the position of the goal point on the path. This algorithm is pure pursuit. The feedforward version chooses the curvature corresponding to the arc command whose associated response trajectory is closest to the goal point.

1.3.3 Path Recorder

The tracker incorporates a recording facility which permits the recording of paths which are traversed when a human is driving. This permits convenient generation of test trajectories. Raw positions are recorded each cycle to a file which is further processed later when it is read during playback.

1.4 Arbiter

The **Arbiter** integrates out the curvature dimension of the vote vectors of both the tactical and strategic controllers to generate a single command to the hardware. When the system operates as a voting element in a larger system, no physical i/o takes place and the vote vectors are available externally. The data flow diagram for the arbiter module is given below.

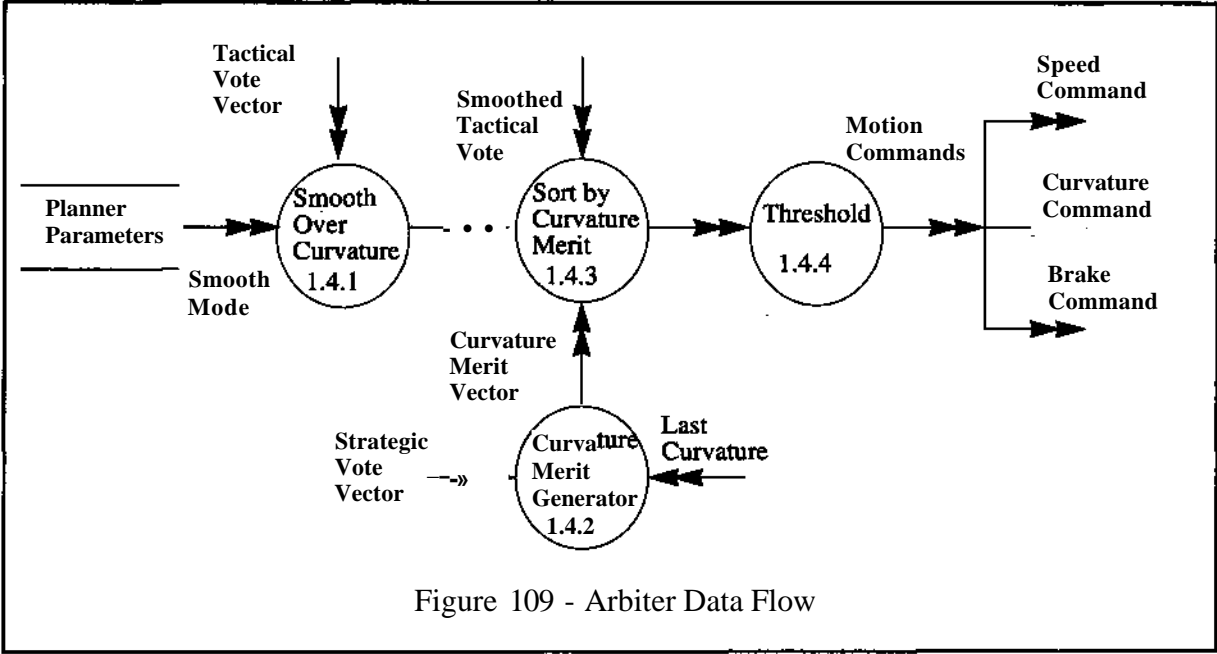


Figure 109 - Arbiter Data Flow

1.4.1 Smooth Over Curvature

The environment normally exhibits a degree of smoothness, so it is appropriate at times for poorly-rated trajectories to degrade the ratings of their spatial neighbors. This mechanism will reduce the impact of errors in both feedback and actuation by causing the vehicle to give obstacles a wide berth. At times, this module is disabled depending on the spacing of the candidate trajectories investigated. The algorithm itself is simple **Gaussian filtering** in one dimension.

1.4.2 Curvature Merit Generator

This module computes a merit for the strategic vote vector elements based on the distance of each command from the maximum of the strategic vote vector measured in curvature space. This measures the degree of disagreement between the two controllers for each element of the tactical vote.

1.4.3 Sort by Curvature Merit

This module sorts the tactical vote vector using the curvature merit of the corresponding strategic votes as the key. Candidate commands are then presented to the final decision process in order of preference.

1.4.4 Threshold

This module implements the final decision. The first trajectory in the sorted list which exceeds a merit threshold is chosen and sent to the vehicle controller. If no curvature exceeds the threshold, a panic stop command is issued.

Section 2: Vehicle

The **Vehicle** object implements very precise models of actuator dynamics, body dynamics, and terrain following. It also encapsulates the vehicle command and vehicle state FIFO queues and provides various daemons for accessing the queued information conveniently. The data flow diagram for the vehicle simulator is given below.

2.1 System Model

The simulation component of the vehicle object serves double duty as the feedforward element in the tactical controller and as a simulation tool for simulated runs in a simulated world. The commanded trajectory is specified in terms of curvature and speed. The predicted vehicle states are the computed vehicle response to that command expressed in terms of the six-axis position and orientation of the vehicle, for each command, as a function of time.

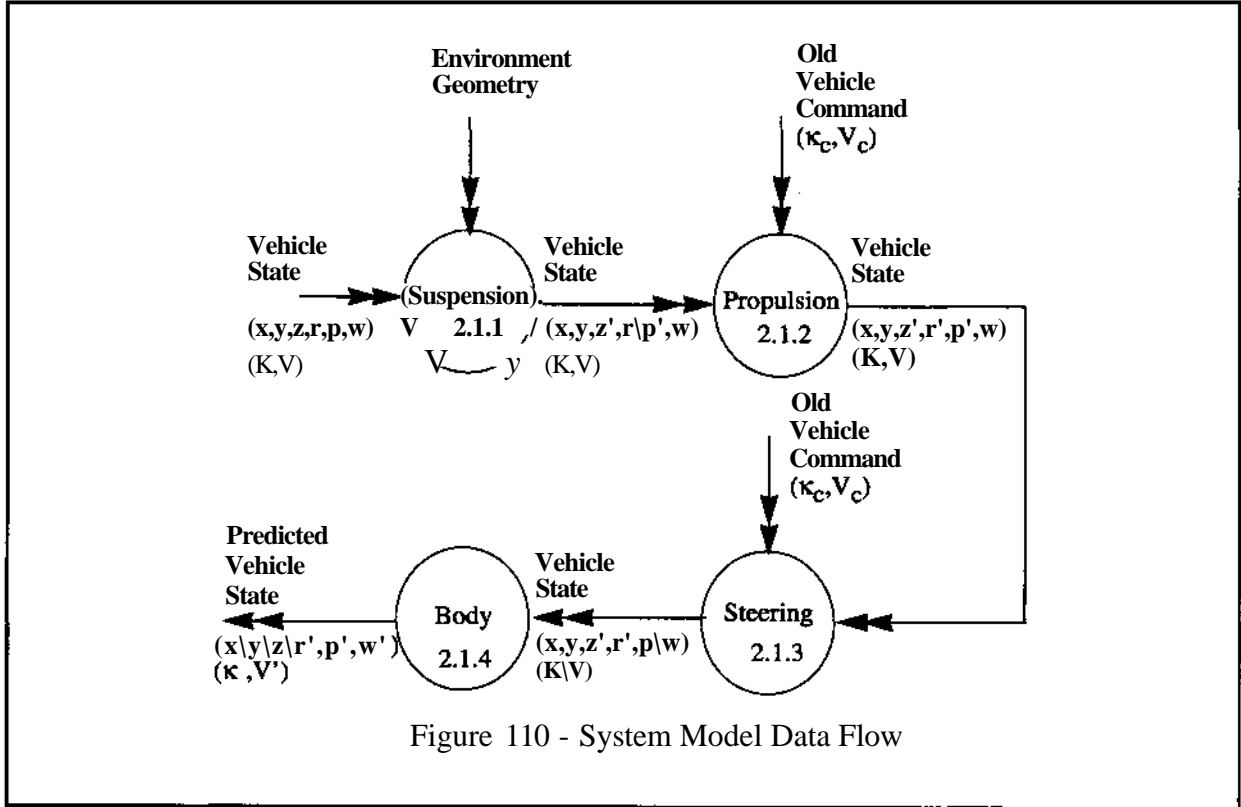


Figure 110 - System Model Data Flow

2.1.1 Suspension

This module uses the current estimate of position and heading and drops the vehicle slowly onto the terrain map in order to determine the attitude it achieves when it settles. Strictly speaking, the attitude determination problem is nonlinear and nonlinear techniques are necessary to solve it. However, the simple technique used here works well in practice.

This module is also responsible for **on demand interpolation** of the vehicle state. It computes the elevation of the terrain under each wheel and if the map is unknown, it attempts to propagate elevation forward from the last known wheel elevation along the response trajectory. Interpolation is limited to a small distance which corresponds to the assumed smoothness of the terrain.

2.1.2 Propulsion

This module computes the speed for the current cycle from the speed of the last cycle and the speed command.

2.1.3 Steering

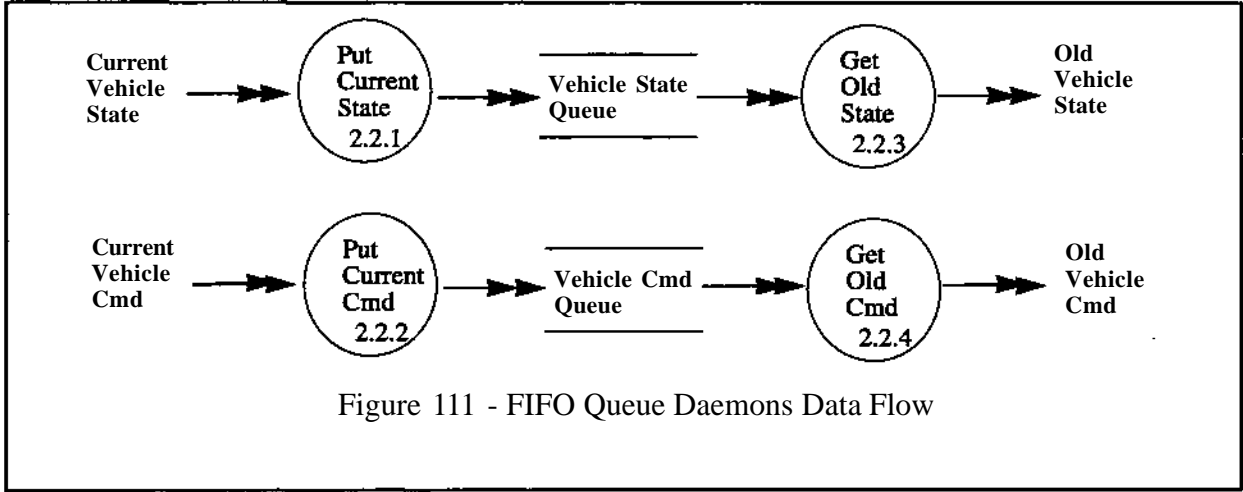
This module implements the differential equation which models the transfer function of the steering column as a delayed, limited, first-order system in velocity. Basically, the current steering position and velocity are the initial conditions for the differential equation and a straightforward rectangular rule integration scheme gives the response of the steering to the input command. A **bicycle model** approximation to the Ackerman steering model is used.

2.1.4 Body

This module computes the Fresnel integrals which convert the speed and curvature signals into position and heading. Correct 3D linear and angular dead reckoning equations are used.

2.2 FIFO Queue Daemons

The FIFO queue daemons perform simple transformation functions on data on an as-required basis.



2.2.1 Put Current State

This module stores incoming time-tagged vehicle states in the state FIFO. Normally this data is generated from an external sensor and the position estimator.

2.2.2 Put Current Command

This module stores incoming time-tagged vehicle commands in the command FIFO. Normally this data is generated by the controller. After a command is issued to the real vehicle, a copy is stored in the queue.

2.2.3 Get Old State

This module looks up old state data. Normally this data is needed by the map manager when processing images.

2.2.4 Get Old Command

This module looks up old command data. This data is needed by the vehicle itself in implementing feedforward of delays.

2.3 Vehicle Daemons

The vehicle daemons compute important properties of the vehicle for the purpose of implementing adaptive regard.

2.3.1 Get Planning Window

This module uses the current vehicle velocity and vehicle and processing latencies to determine the lookahead required in the planning algorithm.

2.3.2 Get Adaptation Speed

The problem of computing the planning window is circular because the trajectory traversed is not known until the speed is, and the speed is not known until the trajectory is. For the purpose of computing the planning window, this module uses either the current speed (when a human controls the throttle) or the mean predicted speed (when the vehicle has a propulsion controller).

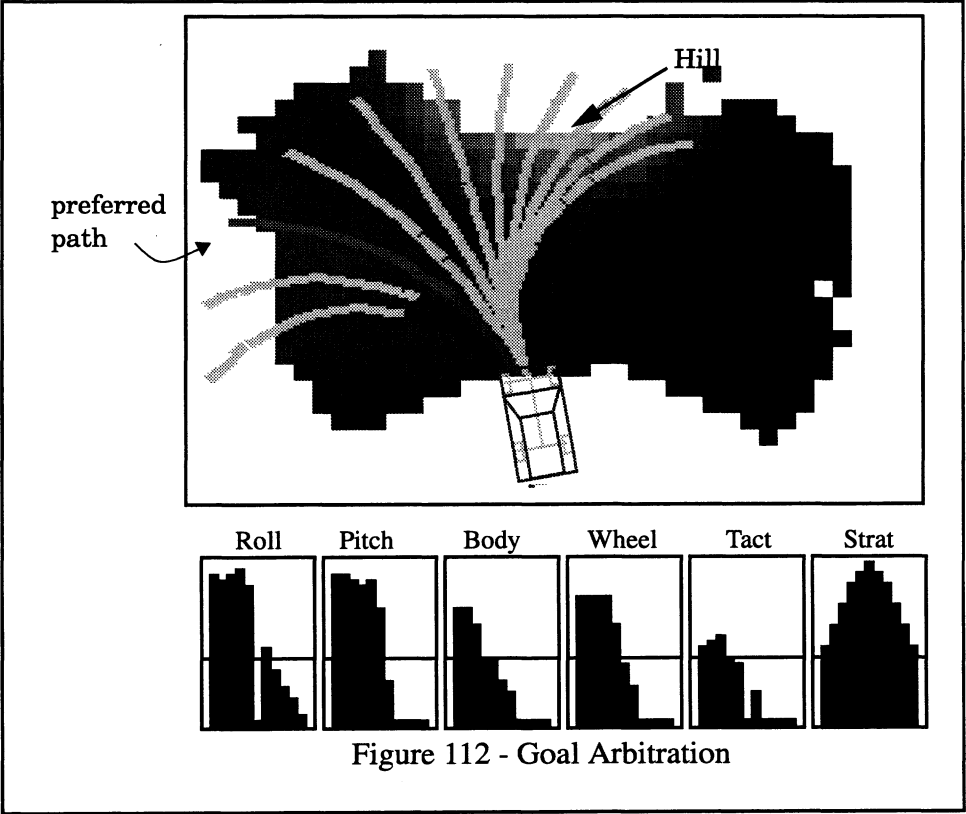
2.3.3 Get Turn React Time

This module estimates the turning reaction time from an estimate of the dynamics of the steering column for the purposes of calculating the planning window.

Chapter 4: Results

Section 1: Optimal Control

Figure 112 is a screen capture of the terrain map data structure including the vehicle and the feedforward response trajectories as the vehicle approaches a large hill that is not traversable. The system issues a left turn command to avoid a hill to its right. The histograms represent the votes for each candidate trajectory (higher values indicate preferred trajectories). The hazards are excessive roll, excessive pitch, collision with the body, and collision with the wheels. The tactical vote is the overall vote of hazard avoidance. The strategic vote is the goal-seeking vote. The arbiter chooses the third trajectory from the left because this is the trajectory which is closest to the strategic vote maximum (straight) but which also satisfies obstacle avoidance.

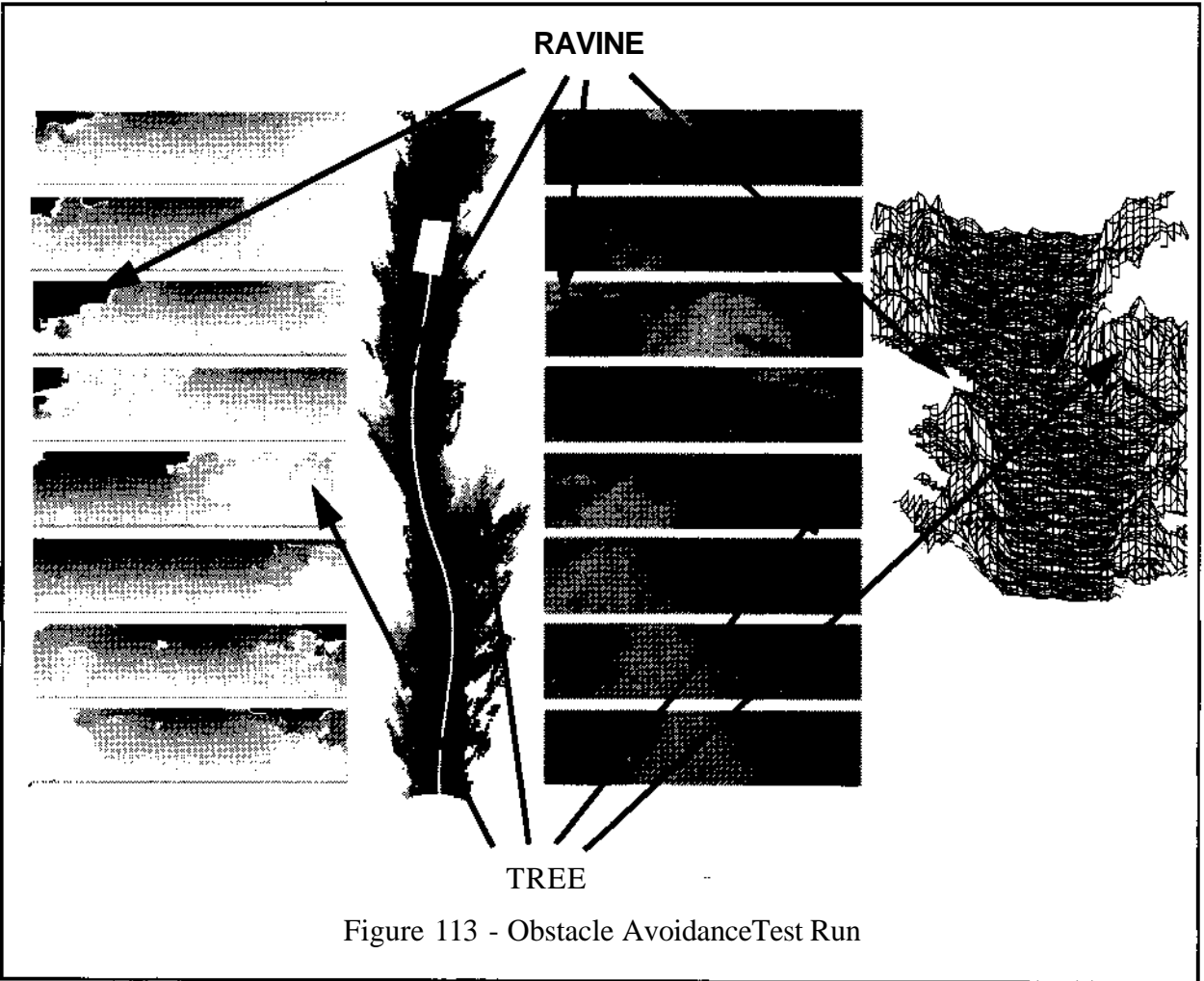


Section 2: Obstacle Avoidance

The system has been integrated with a stereo vision system [57] at the Jet Propulsion Laboratory on a HMMWV. The following figure presents a short autonomous excursion along a dirt road near JPL bounded by trees and bushes on the right and a ravine on the left. In this run of the system, the strategic controller is turned off, and the vehicle wanders the road based on obstacle cues to the sides.

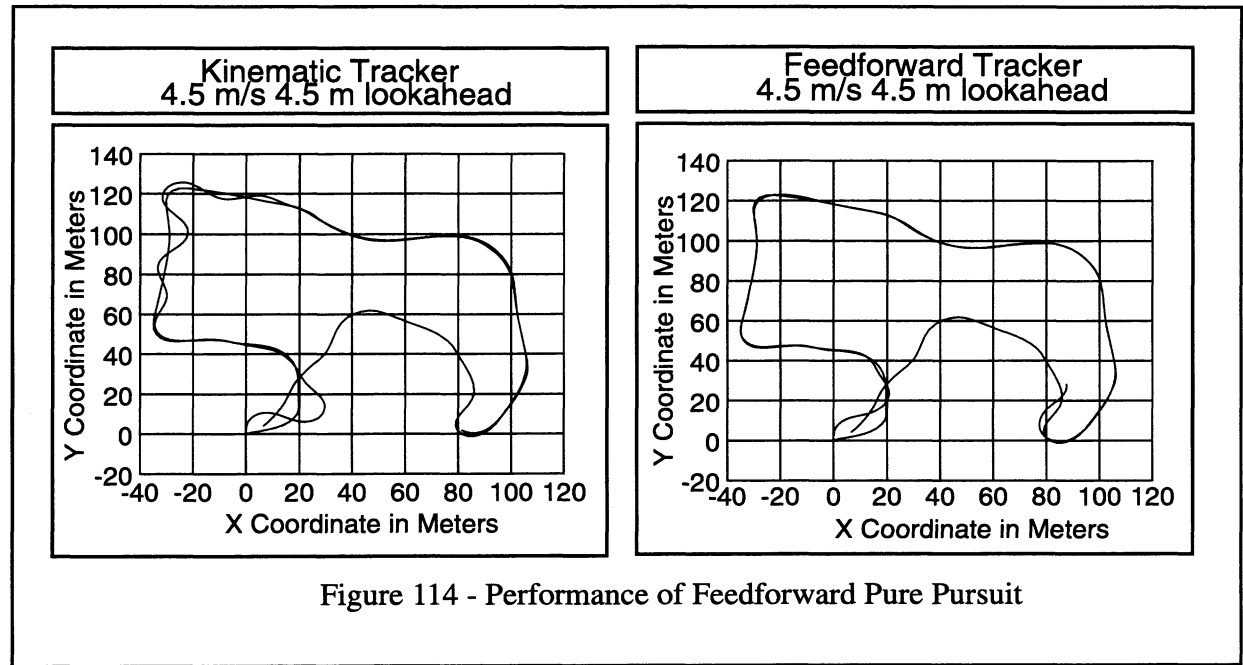
The sequence of images to the left are the stereo range images. To the right are intensity images of the scene corresponding to the range images. The images are positioned more or less in correspondence with their associated position in the terrain map. The terrain map, drawn in the center, is rendered with intensity proportional to elevation. The path followed is drawn leading to the position of the vehicle near the end of the run. A wireframe rendering of the terrain map is provided at the extreme right of the figure.

The run terminates at the end of the road. Two distinct obstacle avoidance maneuvers occur. The first is a left turn to avoid a large tree and the second is a recovery right turn to prevent falling into the ravine. The system drives this road routinely at this point in its development.



Section 3: Goal-Seeking

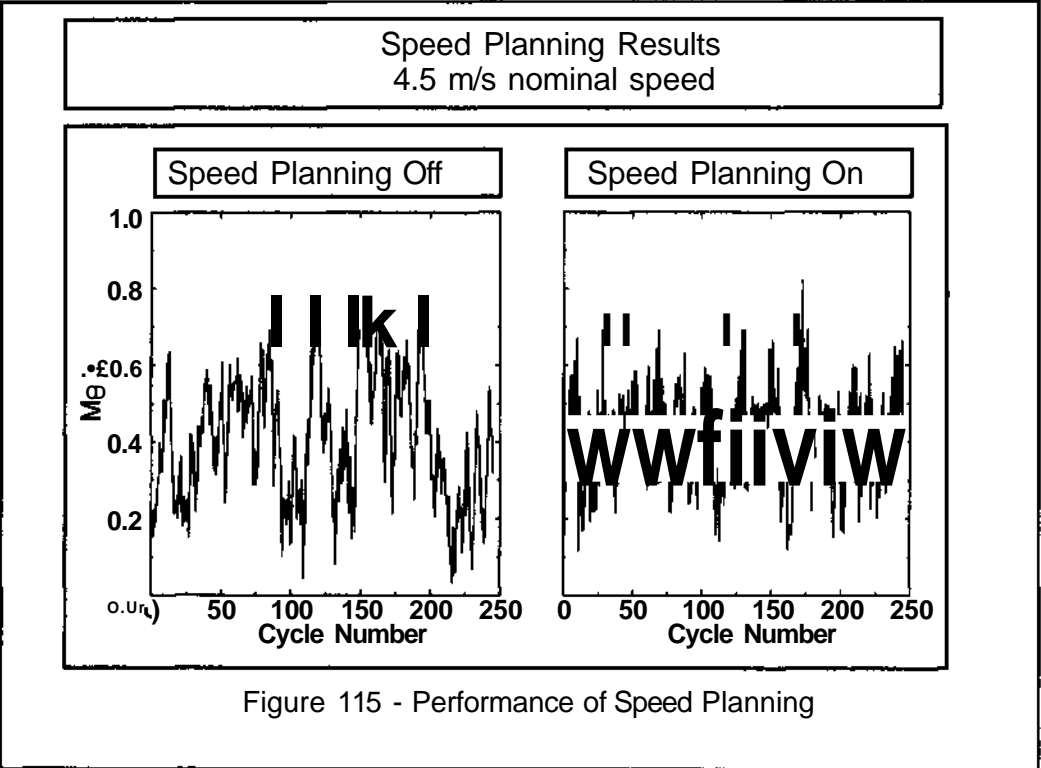
The following figure illustrates the performance of the feedforward algorithm relative to the kinematic one for a simulated vehicle:



Each graph shows the target commanded trajectory and the actual trajectory followed under the algorithm used. Both paths are indistinguishable most of the time. Note, however, that the kinematic tracker goes unstable in the top left figure whereas the feedforward one does not.

Section 4: Adaptive Regulation

The results for a run of the speed planning system in a simulated dense obstacle field at high speed are indicated below:



The vehicle was driven in simulation for 500 cycles at a nominal speed of 4.5 m/s at 3 Hz cycle time. When speed planning was on, the excursion was 703 meters versus 687 meters when it was off. The planner was considered to have failed when the merit fell below a threshold of 0.4. With speed planning on, "failure" occurred 4% of the time, whereas it occurred 6% of the time with speed planning off. More important, the seriousness of the failures with planning off were much worse than those with planning on. The system attempts to maintain a constant merit of 0.4 when the loop is closed. The figure shows that the deviation from 0.4 is much higher in the left figure. The peaks and valleys are wider in the left figure indicating sustained periods of confusion. In plain terms the algorithm makes it possible to drive farther more robustly.

An extra benefit of the speed planning loop is that it will automatically back the vehicle up and drive off in another direction when a panic stop is issued. The path planner supports this as well because it plans correctly for negative velocities, so it backs up "smartly" avoiding obstacles until a new alternative presents itself.

Chapter 5: Summary and Conclusions

This section summarizes all of the most important content of Part V: Planning/Control.

An approach to the high-speed mobility problem is presented which is based fundamentally on the state space representation of a multi-input / multi-output dynamical system and the problem analysis presented in Part II: Analysis. The high-speed local navigation problem has been formulated as an optimal control problem where the search of alternatives is conducted in actuation space. Obstacle avoidance, path planning, path generation, and path following algorithms are introduced which are based on the state space model. These algorithms are stable and reliable at 10 mph on rough terrain and promise to maintain stability at 20 mph and above.

This part describes the state space model which forms the core of the planning system. This part concentrates on the trajectory tracking, obstacle avoidance and sensor stabilization aspects of the navigator. These algorithms form the basis of RANGER'S Controller and Vehicle objects.

Section 1: Concept

The principle of adaptive regard is the organizing principle of the design of the navigator's planning system. **Adaptive regard** is a notion applied to the planning problem. Essentially, this mechanism confines the search for hazards through the environmental model to the **detection zone** - that region of the near environment which the vehicle can reach and is not already committed to travelling. Thus, this mechanism applies to the terrain map and the planning problem and is based on a response argument. Adaptive regard is based on four principles:

- don't look where the vehicle cannot go (free zones)
- don't look where the vehicle is committed to going (dead zone)
- don't look where the vehicle will get another look later (beyond detection zone)
- look where there is an immediate decision of traversability required (detection zone)

The motivation for this approach to planning can be expressed in familiar terms. Analysis suggests that much of the computational resources used to plan paths through the environment are a waste of resources in high-speed mobility scenarios. This waste occurs because:

- Dynamics and other constraints of many kinds including braking and steering maneuverability, underactuation of the vehicle, and processing and communication delays amount to an overwhelming constraint on the feasibility of arbitrary trajectories expressed in configuration space.
- Search through any space of candidate trajectories which do not respect the dynamics constraint amounts to a lot of time wasted thinking about the impossible which tends to increase reaction time and therefore to limit system performance.

1.1 Optimal Control

The problem of intelligent mobility is that of achieving some useful goal while simultaneously avoiding damage to the vehicle.

1.2 Hierarchical Planning

The **local minimum problem** is perhaps the most serious threat to robustness of any local obstacle avoidance strategy for autonomous navigation. In order to enhance robustness, the navigator supports the input and continuous monitoring of strategic goals. With goal-seeking and obstacle avoidance goals operative simultaneously, a means for arbitrating between them in order to avoid actuator contention is necessary. The highest level design strategy for the planning/control system is *hierarchical planning with arbitration*.

1.2.1 Optimal Control

The task of safe navigation can be expressed in classical optimization terms. The navigator must achieve some useful goal while simultaneously satisfying the constraint of avoiding damage to the vehicle or its payload and the constraints of limited vehicle maneuverability.

1.2.2 State Space Model

The navigator is a state space controller because it explicitly forms an expression of the vehicle dynamic state vector in order to predict the hazard signals upon which decisions are based. Instead of forming an **inverse model** which attempts to compute curvature and speed from a C-space curve, a **forward model** is used which implements the equations of dead reckoning. This process which converts an actuation space trajectory to a state space trajectory is a *constrained multidimensional differential equation* called the **state space model**.

1.2.3 Actuation Space Search

Perhaps the most well-known abstraction used in robot path planning is **configuration space** or **C-space**[55]. C-space techniques do not capture dynamics well. The system discussed here plans instead in **actuation space** by considering a number of actuation space alternatives which span the **feasible set** of commands for the vehicle at some gross resolution. An actuation space approach to path planning has been adopted because:

- *A conventional automobile is underactuated (non-holonomic), so the mapping from C-space to actuation space is under-determined. It is not possible, in general, to compute the speed and steering commands which will cause a vehicle to follow an arbitrary C-space curve. The use of actuation space and forward modelling avoids entirely the problem of path generation for non-holonomic vehicles.*
- *Many constraints on vehicle motion are differential in nature and most naturally expressed in actuation space. In the case of steering in particular, a kinematic model can be considered to be one for which curvature can be changed instantaneously. Yet, a moderate-speed vehicle cannot reverse curvature within the maximum useful range measurement that a stereo or lidar sensor can produce. Thus, high-speed motion demands that actuator response characteristics be considered, and the natural space in which to model them is actuation space. The use of actuation space permits expression of these dynamic constraints in terms of dynamic models of actuators.*
- *Actuation space planning is more computationally efficient. Continuous high-speed motion implies that a vehicle has very little time to react to what it sees. A planner *must* decide what to do under stringent timing constraints or the entire navigator will fail. Actuation space permits a fast, coarse-grained evaluation of all feasible alternatives. The problem of C-space combinatoric explosion is solved by planning in actuation space.*

1.2.4 Feedforward Optimal Control

One of the defining characteristics of intelligent systems is deliberation. The high-speed autonomous navigation problem is highly dynamic and requires time-continuous differential equation models, so deliberation here takes the form of **feedforward**.

Feedforward is employed for several reasons:

- It imparts stability to both goal-seeking and obstacle avoidance¹².
- It computes the detection zone required to implement adaptive regard.
- It allows the planner to conduct its search over the space of feasible commands $U(t)$ G U without explicitly considering feasibility because trajectories expressed in actuation space are inherently feasible. Feedforward solves the **clothoid generation problem** trivially.

1.2.5 Forward Modelling

The differences between classical C space planning and actuation space (A space) planning were summarized. In the classical C space scheme, the search of planning alternatives is expressed in **configuration space**. In the **actuation space** approach used here, the inverse system model is never evaluated explicitly - the system simply remembers the correspondence of commands to response trajectories and inverts this list of ordered pairs. In this manner, alternatives are considered in actuator terms, and the navigator *plans in actuation space*.

The essential computational efficiency argument of the approach is:

- Motion planning is a problem involving search.
- Efficiency of search can be improved with constraint ordering heuristics.
- Forward modelling, which considers dynamic modelling before obstacle avoidance, is a valuable ordering heuristic which improves both computational efficiency and system robustness.

Section 2: Design

This section provided detailed algorithm descriptions for the planning and control subsystem.

2.4 Optimal Control

The system attempts to satisfy two goals simultaneously. The **tactical goal** is to avoid obstacles while the **strategic goal** is to track the goal trajectory. At times, goal-seeking may cause collision with obstacles because, for example, the goal may be behind an obstacle. The system incorporates an arbiter which permits obstacle avoidance and goal-seeking to coexist and to simultaneously influence the behavior of the host vehicle. The arbiter can also integrate the commands of a human driver with the autonomous system.

12. It has not been mentioned yet, but feedforward prevents "waffling" in the arbiter because once a slight turn is commanded, it affects all subsequent response trajectories by biasing them to reflect the distance the steering wheel will turn for that cycle. Feedforward leads to a more confident controller because it recognizes when a partial commitment has been made to turn in a certain direction.

2.1.1 Goal Arbitration

Several times a second, the planner considers approximately ten steering angles to use during the next control cycle. The steering angle that results in a path that both satisfies obstacle avoidance constraints and optimizes the cost function is chosen.

2.1.2 State Space Model

A high fidelity feedforward actuator dynamics and terrain following model was introduced here. The forward vehicle model is mathematically involved because it involves solution of a coupled set of eight nonlinear differential equations which form the vehicle state space model.

The basic simulation loop can be written as follows. At each time step:

- simulate suspension - determine attitude from terrain geometry and position
- simulate propulsion - determine new speed from command, state, and attitude
- simulate steering - determine angular velocity from steering and speed
- simulate body - dead reckon from linear and angular velocity and time step

The positions of distinguished points on the body, called **reference points**, are maintained in navigation coordinates throughout the simulation. The suspension model that is used is based on assumptions of rigid terrain and suspension and it computes the attitude of the vehicle which is consistent with terrain contact. Propulsion is modelled as a proportional controller with gravity compensation. The steering model is based on an angular velocity limit on the steering wheel and the bicycle model of steering kinematics. Body dynamics are simulated using the 3D dead reckoning equations which also form the system model in the Kalman filter which is documented in Part VI: Position Estimation.

2.2 Obstacle Avoidance

Hazards are represented as sampled time signals which are parameterized by the commands to which they correspond.

2.2.1 Temporal State Interpolation

The interpolation algorithm used in the system is an interpolation of both the state vector and the hazard signals in time, rather than an interpolation of the terrain map in space.

2.2.2 Hazard Representation

Each hazard is represented on a normalized scale where the maximum value indicates certainty of danger and the minimum value indicates certainty of safety.

2.2.3 Hazard Arbitration

The evaluation of hazards amounts to the generation of a 3D field of the form *merit(hazard,command,time)* because safety is different for each command, different for each hazard, and a time signal. The obstacle avoidance hazard estimator computes a functional on this 3D field which forms the basis of the decision of whether the entire trajectory is to be considered safe.

2.2.4 Hazard Estimation

Three principal forms of hazards are incorporated:

- unknown terrain
- collision of the terrain with the body
- static instability

2.2.5 Hazard Uncertainty

Most hazards outlined above are computed as gradients of the terrain over some support. In such cases, the elevation uncertainty stored in the terrain map is used to compute an uncertainty on the hazard estimates as outlined in Part II: Analysis.

2.3 Goal-Seeking

The basic pure pursuit algorithm specified in the concept section is modified to adapt it to both rough terrain and high speeds. First, the lookahead distance is adaptive to the current tracking error. Second, a feedforward option in the tracking algorithm incorporates the output of the system model simulator into the tracker.

2.4 Adaptive Regard

In order to evaluate safety, the tactical control module computes the response to candidate commands simulated forward in time until the **detection zone** is reached. There is no real need to compute this region explicitly - the only real requirement is to determine where it starts and where it ends in terms of distance from the vehicle position.

2.4.1 Planning Window

The **planning window** of tactical planning is analogous to the **range window** in perception. It confines the search for hazards to the detection zone. One of the highest level system requirements is to attempt to maintain continuous motion, so adaptive regard must be based on turning maneuvers (which consume more space) instead of braking maneuvers.

The planning window is computed by predicting the *distance* required to execute an *impulse turn* at the current speed with the best available estimates of the output latencies that will apply. An impulse turn is a turn from zero curvature to the maximum allowed curvature.

Precision in computation of the planning window requires careful treatment of time. The planning window is measured from the position where the vehicle will be when the steering actuator starts moving.

2.4.2 Real-Time Latency Modelling

The planning problem has latency concerns similar to those of perception. Without latency models, obstacle avoidance significantly underestimates the distance required to react for two reasons.

- Position estimate input latency implies that the vehicle is actually much closer than the last available position measurement suggests.
- Command output latency and actuator dynamics imply that it actually takes much more distance to turn than would be expected from instantaneous response models.

A model of these latencies is accomplished with the following mechanisms:

- all input and output is time-stamped
- all input and output is stored in internal FIFO queues
- all sensor latencies are modelled

- all actuator latencies are modelled

It is important to recognize that these FIFO queues do not introduce artificial delay. They are used to *model* the delays which already exist in the hardware.

2.5 Adaptive Regulators

In cases where the vehicle infrastructure permits, adaptive regulation mechanisms are available to couple vehicle speed to its own estimates of reliability and to intelligently point the sensor field of view.

Section 3: Implementation

This section overviewed the implementation of the planning and control system. This system is implemented as two main objects - the Vehicle and the Controller. The Vehicle encapsulates vehicle state information and maintains the time histories of both measured states and issued commands. It also implements the state space model used for implementing feedforward. The Controller consists of a Tactical Controller responsible for obstacle avoidance, a Strategic Controller responsible for goal-seeking, and an arbiter which implements the optimal control algorithm through a prioritized sort of the votes of the two controllers.

3.1 Controller

Most aspects of the **control laws** (with the exception of feedforward itself) are implemented in the controller module. The elements of this subsystem are Command Generator, Tactical Controller, Strategic Controller, and Arbiter.

3.2 Vehicle

The **Vehicle** object implements very precise models of actuator dynamics, body dynamics, and terrain following. It also encapsulates the vehicle command and vehicle state FIFO queues and provides various daemons for accessing the queued information conveniently. The elements of this subsystem are System Model, and FIFO Queue Daemons.

Section 4: Results

4A Optimal Control

An illustration of the operation of the optimal control arbiter was provided where the goal-seeking behavior wants to go straight and hazard avoidance wants to turn left. The output of the arbiter is the softest left turn which avoids the obstacle directly in front of the vehicle.

4.2 Obstacle Avoidance

The system has been integrated with a stereo vision system [57] at the Jet Propulsion Laboratory on a HMMWV. A short autonomous excursion along a dirt road near JPL bounded by trees and bushes on the right and a ravine on the left. The run terminates at the end of the road. Two distinct

obstacle avoidance maneuvers occur. The first is a left turn to avoid a large tree and the second is a recovery right turn to prevent falling into the ravine. The system drives this road routinely at this point in its development.

43 **Goal-Seeking**

The performance of the feedforward algorithm relative to the kinematic one was illustrated for a simulated vehicle. The kinematic tracker goes unstable whereas the feedforward one does not.

4.4 **Adaptive Regulation**

The results for a run of the speed planning system in a simulated dense obstacle field at high speed were presented. The algorithm makes it possible to drive farther more robustly.

PART VI: Position Estimation

This part describes the state space Kalman Filter which forms the core of the position estimation system. These algorithms form the basis of RANGER’s Position Estimator object.

The Kalman filter has many applications in mobile robotics ranging from perception, to position estimation, to control. This part formulates a navigation Kalman filter. That is, one which estimates the position of autonomous vehicles. The filter is developed according to the state space formulation of Kalman’s original papers which is particularly appropriate for the problem of autonomous vehicle position estimation.

R. E. Kalman solved the optimal estimation problem for a certain class of linear systems about 35 years ago. There is really only *one* Kalman filter with a few variations and most research on the Kalman filter amounts to filling in the elements in the matrices which appear in the original formulation. This is not to say that this **modelling problem** is trivial.

The most important decisions in the filter design problem are the **modelling** decisions which outline the states, the measurements, the noise models, coordinate systems, linear approximations, etc. Thus, the purpose of this part is to outline a set of plausible modelling decisions for the navigation problem for autonomous vehicles which have led to a working Kalman filter.

Chapter 1: Concept

This section outlines an approach to autonomous vehicle position estimation based on the state space Kalman filter. This particular filter formulation is quite general. This generality is possible because the problem has been addressed

- in 3D
- in state space, with an augmented state vector
- asynchronously
- with tensor calculus measurement models

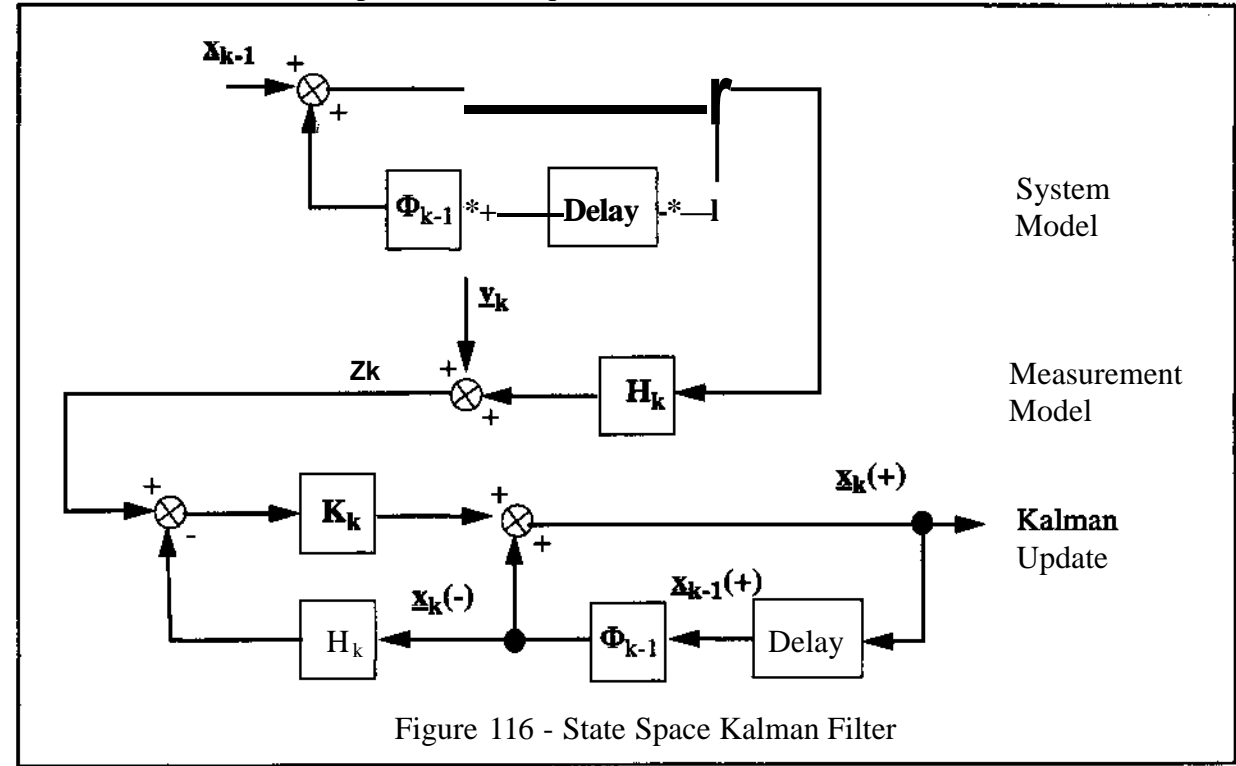
The formulation has wide-ranging uses. Some of the applications include:

- as the basis of a vehicle position estimation system, whether any or all of dead reckoning, triangulation, or terrain aids or other landmarks are used
- as the dead reckoning element and overall integration element when INS or GPS is used
- as the mechanism for map matching in mapping applications
- as the identification element in adaptive control applications

It can perform these functions individually or all at once - subject to the availability of sensors and the underlying requirements of observability.

Section 1: State Space Kalman Filter

The fundamentals of the state space Kalman filter are described in the appendices for the benefit of readers unfamiliar with the basic concept. The equations of the appendix can be represented in the following block diagram. The system model used is identical to the model used in the planner. It is expressed here as the transition matrix O^{\wedge} . Conceptually, the system model estimates vehicle state based solely on a measurement of time because it expresses the differential equations of motion and the state vector provides all required initial conditions.



1.1 Positioning Requirements

The basic requirements on position estimation for the purposes of implementing autonomous local navigation are as follows:

1.1.1 Bandwidth

Position updates must be available at 100 Hz in order to remove image distortions. This can be achieved by running the system model equations of the filter at this rate because the system model update rate is completely independent of the measurement arrival rate.

1.1.2 Relative Accuracy

Relative position accuracy must limit accumulated position error to one terrain map cell over a distance of the maximum planner lookahead over the mission. At 20 mph, this implies a requirement of about 1% of distance travelled. Odometry is typically available on autonomous ground vehicles and can meet this requirement if properly calibrated.

1.1.3 Absolute Accuracy

While the strategic layer of the **standard model** may require bounded absolute position accuracy, the only absolute accuracy requirement of the tactical layer is to measure attitude (pitch and roll) to within about 10% of the maximum expected over rough navigable terrain. In order to avoid tipover, the navigator must know “which way is up” to about 1° absolute accuracy.

1.2 Design Decisions and Assumptions

For the requirements mentioned above, an AHRS dead reckoning Kalman filter promises to meet all requirements - assuming that the system model cycles much faster than the sensors can provide measurements and the attitude and heading measurements are sufficiently accurate. The filter is formulated for a general redundant asynchronous sensor suite. In order to achieve the 100 Hz update rate, the asynchronous model permits many measurement models to be expressed in closed form as scalar equations, and this reduces the matrix computations to a minimum. A few key assumptions permit the filter to perform as required:

1.2.1 Low Dynamics Assumption

This assumption that linear and angular velocity are mostly constant between measurements makes it possible to reduce the state vector dimensionality to a minimum.

1.2.2 Taylor Remainder Theorem

Under this basic theorem of calculus, provided the low dynamics assumption holds, uncertainty models for sensors and states can be generated from the last neglected term in the relevant Taylor series.

1.2.3 Principle Motion Assumption

This assumption that the vehicle moves primarily along the body y direction and rotates primarily about body z permits the filter to be formulated in observable form without the need for any landmark damping. It also eliminates two linear and two angular velocities from the state vector.

Section 2: AHRS Dead Reckoning

AHRS is an acronym for Attitude and Heading Reference System which is used in the navigation industry to mean a suite of components used for measuring the attitude of a vehicle and sold as an integrated package. The primary distinction between the AHRS and the INS is the absence of accelerometer dead reckoning. The AHRS often provides no position output at all and is best suited for odometric dead reckoning. This section proposes a filter for a mobile robot position estimation system which utilizes an AHRS as the attitude indicator.

2.1 AHRS Components

The AHRS can be considered to be a sensor package consisting of any of the following components:

- integrating gyroscopes
- a magnetic compass
- inclinometers

2.2 Other Sensors

The position estimation system will integrate these attitude indications with:

- wheel or transmission encoder(s)
- Doppler ground speed radar
- steering wheel encoder

2.3 Extended Kalman Filter

In many cases, either the system model or the measurement matrices or both are nonlinear so a linearized Kalman filter of some form is necessary. In order to support wide excursion missions, and because a nominal trajectory is not always available, an EKF will be formulated.

2.4 Unforced Kalman Filter

It is possible to incorporate deterministic control inputs into the Kalman filter, and this requires slight changes to the following equations. It is a simple matter to incorporate steering dynamics via the steering control signal¹. However, in the case of autonomous commercial vehicles, little is known in land-based autonomous vehicle circles about powertrain dynamics, so any such model might be grossly in error to the point of destroying the utility of the filter. For this reason, only the **unforced Kalman filter** is considered here.

A second source of deterministic inputs are the disturbances which result from terrain following in cross-country vehicles. These amount to forces generated by the terrain which prevent the vehicle from sinking into the terrain. For example, a high-speed vehicle pitches violently when it encounters a steep grade. These can be estimated and folded into the filter.

1. Note control signals are distinguished from sensory feedback. In the case of steering, the drive amp current demand is the drive signal.

Section 3: Advantages

It is worth noting the advantages of applying the Kalman filter to the position estimation problem in such a simple case. The main advantage is the improvement in accuracy available due to redundant measurements.

3.1 Redundant Measurements

Sensors such as a transmission encoder and Doppler groundspeed radar provide redundant measurements of velocity which can be used to improve the estimate. The 3 axis AHRS output makes 3D dead reckoning possible and a steering encoder may be able to improve attitude response of the AHRS around the most important axis for dead reckoning purposes since it does not suffer from the settling problems of many commercial AHRS systems.

3.2 Integration of Dead Reckoning and Triangulation

Of course, if any fix information is available, the formulation provides a simple mechanism for improving a dead reckoned estimate considerably.

3.3 Modelling Frequency Response

Most inexpensive AHRS systems are based on fluxgate compasses for heading and inclinometers for attitude. Both of these sensor types can have unacceptably long settling times - they cannot track high dynamics faithfully. State vector augmentation can be used to model the frequency response of such sensors and then, when rate gyros are added to the sensor suite, the overall attitude and heading estimate can be improved.

3.4 Computational Inertial Force Compensation

Another problem of commercial AHRS systems is centrifugal coupling of acceleration and attitude. Inclinometer devices cannot distinguish acceleration from attitude changes, but other measurements of path curvature and linear velocity can be used to compensate for centrifugal accelerations by removing them computationally from the clinometer outputs.

3.5 Nonlinear Error Propagation

Accelerometers or acceleration states would permit modification of the measurement noises to account for settling problems. The advantage of the EKF over the linear KF is that the uncertainty propagation is more accurate. For example, if a poor heading reference is used, it is the direction transverse to the direction of travel which can incur the most error. The cross-coupling terms in the system Jacobian can correctly account for this.

3.6 Noise Immunity

Another advantage, specifically of the state space formulation of the Kalman filter, is its true filtering behavior. Appropriate choice of the system disturbance model will cause the filter to automatically reject erroneous high frequency input. This arises as a side-effect of considering the new estimate to be a weighted sum of the old estimate and the estimate generated by the measurements. In practice, however, a "spiky" sensor is best managed by a true low-pass prefiltering operation.

Chapter 2: Design

Section 1: AHRS Dead Reckoning System Model

This section begins the business of writing out the math. The rest of this part depends heavily on the kinematic notation and relationships set out in the appendices. Most of the measurement and system model is a 3D kinematic model. The reader will have to refer to the appendices in order to closely follow the kinematics here.

1.1 Nav Frame System Model

Under the low dynamics assumption, the discrete system model can be formulated as the six-axis dead reckoning equations. The first question to resolve is the coordinate system in which to represent the linear velocity states. A practical choice depends on the observability issue.

Let x , y , z , θ , ϕ , and ψ denote the vehicle position and attitude in the navigation frame. For notational convenience, translational and attitude variables will be grouped together as follows:

$$\bar{\mathbf{r}} = [x \ y \ z]^T \quad \bar{\mathbf{p}} = [\theta \ \phi \ \psi]^T \quad \bar{\mathbf{x}} = [\bar{\mathbf{r}}^T \ \bar{\mathbf{p}}^T \ \dot{\bar{\mathbf{r}}}^T \ \dot{\bar{\mathbf{p}}}^T]^T$$

The state equations are then:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{r} \\ \mathbf{p} \\ \dot{\mathbf{r}} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{p} \\ \dot{\mathbf{r}} \\ \dot{\mathbf{p}} \end{bmatrix} \quad \dot{\mathbf{x}} = \mathbf{F}\mathbf{x}$$

This matrix is trivial because the assumed dynamics were trivial. It is recommended that the equations be partitioned for efficiency reasons, but they are cast in this form to identify the state variables. Given measurements of the rate variables and the time step, it is clear that the transition matrix is:

$$\bar{\mathbf{x}}_{k+1} = \begin{bmatrix} \mathbf{I} & \mathbf{F} \Delta t \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\mathbf{x}}_k$$

because the \mathbf{F} matrix is constant and vanishes when squared, so that:

$$\bar{\mathbf{D}}_k = \mathbf{I} + \mathbf{F} \Delta t$$

exactly. This defines the $\bar{\mathbf{O}}_k$ matrix in the model, and notice that it makes the constant velocity low dynamics assumption quite explicit. This is also a literal expression of the usual equations of dead reckoning.

1.2 Observability in the Nav Frame

Unfortunately, the above model was attempted and found to have observability problems with the standard suite of dead reckoning sensors. The nav frame model is ideal when an INS or GPS is available to provide measurements in the nav frame, but without such sensors, the filter diverges unacceptably. Even with perfect DR sensors, this formulation does not work as a matter of mathematics.

1.2.1 Entire State Vector Observability

To see this, consider a sensor suite that provides direct measurement of attitude and attitude rate, plus a measurement of the forward component of velocity. Let the vehicle drive along the x axis of the navigation frame - then, for this trajectory:

$$\mathbf{z} = \mathbf{H}\mathbf{x}$$

$$\begin{bmatrix} 0 \\ \mathbf{v} \\ \dot{x} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 6 \\ \bullet \\ \mathbf{v} \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ 6 \\ \bullet \\ \cdot \end{bmatrix}$$

Recall that the observability question rests on the rank of the following matrix being that of the system model, which is 12 in this case:

$$\mathbf{S} = [\mathbf{H}^T \mathbf{O}^T \mathbf{H}^T \dots (\mathbf{\Phi}^T)^{n-1} \mathbf{H}^T]$$

Each vertical partition of the matrix \mathbf{E}^* is of the form²:

$$\mathbf{H}^T (\mathbf{\Phi}^T)^n = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \text{ndt} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \text{ndt} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \text{ndt} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

2. Note that the transpose of the RHS is the result so the partitioning is vertical in this view.

so that each partition contains a column linearly independent on the corresponding columns of all other partitions. Therefore the rank is that of a single partition. Notice that for any partition, columns 1,2,3,8 and 9 are all zeros so that the rank of this matrix is 7. Hence the system is not observable.

1.2.2 Rate Variable Observability

A more careful analysis would reflect on the fact that the position states are never observable in a pure dead reckoning system such as the one proposed here - in either the nav or body frames. While it is the case that the initial conditions can be used to compute the position states from velocity and attitude, the observability criterion does not capture this possibility. However, if the criterion is reformulated for the rate variables only, the system remains unobservable.

To see this practically, notice that either a transmission encoder or Doppler groundspeed radar measures only the component of vehicle velocity which is directed along the body y axis. The measurement matrices for such sensors act so as to project the speed residual onto all three of the state variables \dot{x} , \dot{y} and \dot{z} . However, once the velocity vector is of correct magnitude, all adjustment stops. So if the predicted *speed* is correct but the *velocity vector* is wrong, the filter will not rotate the velocity vector in order to correct it.

The measurement models of the nav frame model generate the body to nav transform in the measurement models of the DR sensors, so provided the measurement models of the forthcoming section are modified, the above nav frame model can be successfully used with INS or GPS, or any sensor suite which measures translation along the nonobservable axes.

Some AHRS provide accelerations in the body frame. These can be integrated in a straightforward manner with either the nav frame model or the body frame model discussed next.

1.3 Body Frame System Model

These problems of observability can be overcome by changing the system model to force the velocity vector to be oriented along the body y axis. However, this would be like having one state variable and faking three others. In truth, only one is independent, so there **is** only one. There is a more elegant way. Specifically, the state variables are reduced in number and the system model is reformulated to *explicitly* assume that:

- the vehicle translates only along the body y axis
- the vehicle rotates only around the body z axis

These are assumptions, of course, which will be violated under certain circumstances. However, there is no choice without some mechanism to measure translation in the vertical and sideways. This will be called the **principal motion assumption**, and it amounts to replacing six legitimate states with two special combinations of themselves. The principal motion assumption solves the observability problem.

1.3.1 State Vector

In this model, the state variables are:

$$\bar{\mathbf{x}} = [\mathbf{x} \ y \ z \ V \ \theta \ \phi \ \psi \ \beta]^T$$

where V is the projection of the vehicle velocity onto the body y axis, and $\dot{\psi}$ is the projection of the vehicle angular velocity onto the body z axis.

The observability problem is fixed by writing the continuous time system differential equations as follows:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ V \\ \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -V s_{\theta} c_{\psi} \\ V c_{\theta} c_{\psi} \\ V s_{\theta} \\ 0 \\ \dot{\theta} \\ -\dot{\psi} c_{\theta} \\ \dot{\psi} c_{\theta} c_{\psi} \\ 0 \end{bmatrix} + \mathbf{A} \begin{bmatrix} x \\ y \\ z \\ V \\ \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

This result incorporates kinematic transforms from the appendices. Technically, this system model is *nonlinear*, therefore it must be linearized according to the rules for an EKF.

1.3.2 System Jacobian

The linearized continuous-time differential equation is:

$$\frac{d}{dt} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta V \\ \Delta \theta \\ \Delta \psi \\ \Delta \dot{\theta} \\ \Delta \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -V s_{\theta} c_{\psi} & V s_{\theta} c_{\psi} & 0 & -V c_{\theta} c_{\psi} & 0 \\ 0 & 0 & 0 & V c_{\theta} c_{\psi} & -V c_{\theta} c_{\psi} & 0 & -V s_{\theta} c_{\psi} & 0 \\ 0 & 0 & 0 & V s_{\theta} & 0 & 0 & V c_{\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dot{\psi} c_{\theta} & 0 & s_{\theta} \\ 0 & 0 & 0 & 0 & -\dot{\psi} c_{\theta} c_{\psi} & \dot{\psi} s_{\theta} & 0 & -\dot{\psi} c_{\theta} c_{\psi} \\ 0 & 0 & 0 & 0 & \dot{\psi} s_{\theta} c_{\psi} & -\dot{\psi} c_{\theta} c_{\psi} & 0 & C_{\theta} c_{\psi} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta V \\ \Delta \theta \\ \Delta \psi \\ \Delta \dot{\theta} \\ \Delta \dot{\psi} \end{bmatrix}$$

which gives the F matrix of the EKF.

1.3.3 Transition Matrix

The above system Jacobian matrix should be distinguished from the transition matrix. Admittedly, \mathbf{O} does not exist for nonlinear plants, but the system differential equation can be approximated by reinvoking the low dynamics assumption. The above nonlinear plant is linearized in time as follows. Under the assumption that \mathbf{V} and \mathbf{p} represent all components of the linear and angular velocity of the vehicle, then the dead reckoning equations are:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ \mathbf{V} \\ \mathbf{e} \\ \mathbf{\phi} \\ \mathbf{V} \end{bmatrix}_{\mathbf{K}+1} = \begin{bmatrix} 1 & 0 & 0 & -s\backslash/c6dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & c\backslash/c6dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & s0dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & s\langle\rangle dt \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -t6c\langle\rangle dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & c\langle\rangle dt/c0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{\mathbf{K}} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ \mathbf{V} \\ \mathbf{6} \\ \mathbf{*} \\ \mathbf{\psi} \end{bmatrix}_{\mathbf{K}}$$

where the kinematic transforms of the appendices were used to compute the projections of the velocity and angular velocity vectors. This transition matrix is just the equations of 3D dead reckoning. It has been generated by re-expressing the nonlinear plant as a matrix function of the states. It is first-order in time because the highest order time derivatives included are order 1.

Section 2: AHRS Dead Reckoning Measurement Model

Certain of the measurement relationships for dead reckoning sensors are nonlinear. Therefore, a linearized Kalman filter is necessary in this application whether or not the system model is nonlinear. A slow moving vehicle assumption has already been adopted in the system model, so it is unnecessary to use a simple linearized filter in order to track high accelerations. Therefore, an extended Kalman filter will be formulated. This section provides the 3D measurement models for such a filter incorporating many of the sensors commonly used on autonomous vehicles.

One of the advantages of the body frame system model is that almost all of the measurement models are trivial. The tradeoff is that now the system model has rotation transforms in it. These rotation matrices cannot be avoided - they can only be moved around.

2.1 Transmission Encoder

While, strictly speaking, a transmission encoder measures differential distance, it can be considered to be integral with the system clock and therefore it is a device which measures velocity. Relative computer clock accuracies are such that they can be considered perfect. This avoids the considerable difficulty associated with the arc length derivative called **curvature** in a model where everything else is a time derivative.

The measurement model is trivial.

$$\mathbf{V}_{\text{enc}} = \mathbf{V} \qquad \mathbf{H}_{\text{enc}} = \frac{\partial \mathbf{V}_{\text{enc}}}{\partial \bar{\mathbf{x}}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

2.2 Doppler Groundspeed Radar

The Doppler sensor has a model similar to the transmission encoder. As a **range rate** measurement device, its output must be calibrated to recover the constant cosine of the slant angle between the direction of the beam and the direction of the body forward axis. This amounts to a scale factor which can be assumed to have been calibrated out, so that \mathbf{V}_{dop} is considered to represent the vehicle speed³.

$$\mathbf{V}_{\text{dop}} = \mathbf{V} \qquad \mathbf{H}_{\text{dop}} = \frac{\partial \mathbf{V}_{\text{dop}}}{\partial \bar{\mathbf{x}}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

3. A second Doppler radar mounted transverse near the front of an Ackerman steer vehicle should provide a good redundant measurement of yaw rate. Note however, that the measurement must be adjusted for pitch and roll.

2.3 Compass

Compasses differ in their ability to reduce heeling error, and may have nontrivial dynamics. The fluxgate can sometimes sustain as much as 45° of heel without significant error. Newer three coil devices are immune to heeling error. With no information on compass dynamics or construction, the following model can be adopted. Under the assumption that **variation** and **deviation** are already accounted for in the measurement:

$$\psi_{\text{com}} = \psi \quad H_{\text{com}} = \frac{\partial \psi_{\text{com}}}{\partial \bar{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

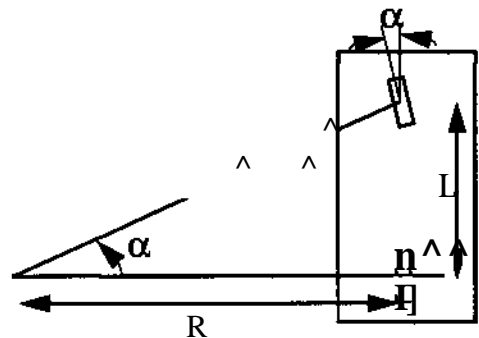
2A AHRS

The clinometers, compass, and gyros in an AHRS can be assumed to measure the vehicle attitude directly so their measurement matrices are the identity matrix. For example:

$$\tilde{p}_{\text{ahrs}} = \begin{bmatrix} I \end{bmatrix} \tilde{p} \quad H_p = \frac{\partial \tilde{p}_{\text{ahrs}}}{\partial \tilde{p}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.5 Steering Wheel Encoder

A steering wheel encoder provides a low fidelity measurement of trajectory curvature, and hence, when multiplied by the speed, a measurement of the rate of rotation of the vehicle about the body z axis⁴. The 3D formulation of this relationship is surprisingly difficult. Let the angular velocity vector directed along the body z axis be called $\dot{\beta}$. Using the bicycle model approximation, the path curvature κ , radius of curvature R , and steer angle α are related by the wheelbase L .

$$\kappa = \frac{1}{R} = \frac{\tan \alpha}{L} = \frac{d\alpha}{ds}$$


Where $\tan \alpha$ denotes the tangent of α . Rotation rate is obtained from the speed V as:

$$\dot{\beta} = \frac{d\beta}{ds} \frac{ds}{dt} = \kappa V = \frac{V \tan \alpha}{L}$$

4. Note that this is not yaw. Yaw is measured about the z axis of the navigation frame. Consider driving up a steep hill. The vehicle steers in the plane of the hill, not about the gravity vector.

Finally, then, the steer angle a is the quantity indicated by the encoder. It is an indirect measurement of the ratio of $\dot{\alpha}$ to velocity through the measurement function:

$$a = \operatorname{atanl} \left(\frac{L\dot{\beta}}{V} \right) = \operatorname{atari} (KL)$$

The Jacobian is the measurement matrix. It is easier to formulate the measurement matrix by utilizing the chain rule and considering the product KL to constitute a single variable.

$$H_a = \begin{bmatrix} 0 & 0 & 0 & \frac{\partial \alpha}{\partial V} & 0 & 0 & 0 & \frac{\partial \alpha}{\partial \dot{\beta}} \end{bmatrix}$$

$$\frac{da}{dv} = \left(\frac{1}{V + (KL)^2 X} \right) \left(-\frac{KL}{V} \right) \qquad \frac{da}{a\dot{p}} = \left(\frac{1}{1 + (KL)^2} \right) \left(\frac{KL}{\dot{p}} \right)$$

Note that V is derived from the states and not from the encoder or radar. In general, any measurement model uses only the current state and parameter estimates and never any other measurement. The partial derivatives will approach infinity if they are evaluated in this form near zero linear or angular velocity. Since $K = p/V$, they can usefully be rewritten as:

$$\frac{\partial \alpha}{\partial V} = \frac{-L\dot{\beta}}{1 + (L\dot{\beta})^2} \qquad \frac{\partial \alpha}{\partial \dot{\beta}} = \left(\frac{LV}{1 + (L\dot{\beta})^2} \right)$$

where the apparent singularities have been removed. Provided at least one of $\dot{\alpha}$ and V are nonzero, the filter will do the right thing⁵. However, when both are zero, the Jacobian is filled with zeros and this may cause a singular matrix in the computation of the Kalman gain. Physically, the steer angle is irrelevant when the vehicle is not moving, so the measurement must be discarded.

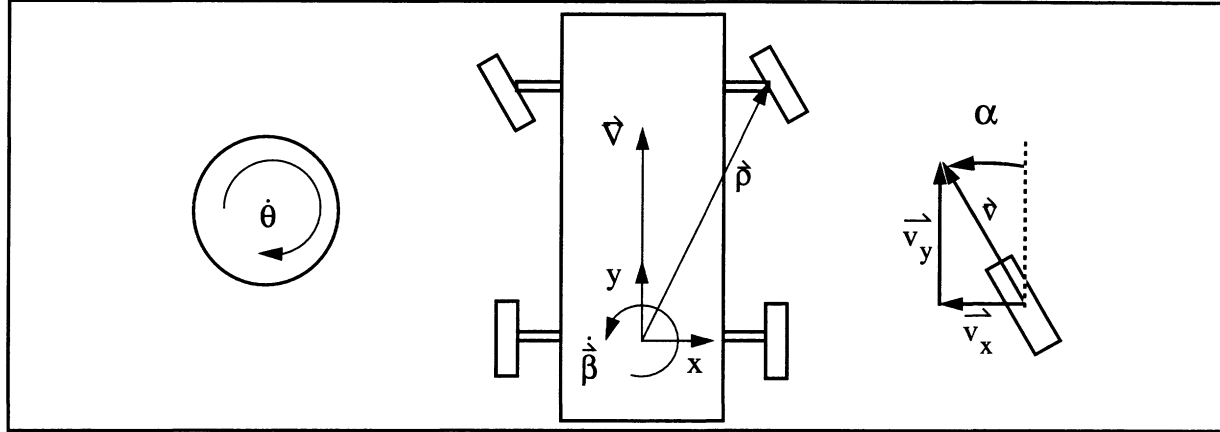
The steering measurement model is also unique in that it is the only DR measurement model for which the predictive measurement is not directly equal to a state. This results simply from the choice of the steer angle as the measurement instead of the angular velocity state.

2.6 Wheel Encoders

It is more convenient on many vehicles to measure wheel rotations rather than engine or transmission rotations. Indeed, if the wheels are actuated by electric motors, there is no engine or central transmission from it to the wheels at all. It is important to distinguish several kinds of measurements that may be available from instrumented wheels. A fixed wheel is permitted to rotate about a single axis - the one associated with forward motion. A free wheel may rotate about a vertical axis as well as the axis associated with forward motion. Either of these two degrees of freedom may be powered or not and either may be instrumented.

5. For an Ackerman steer vehicle, if the vehicle speed is zero and the angular velocity is not, something else is probably wrong.

Consider a single wheel on a vehicle that has two degrees of rotational freedom as shown below. Let \vec{p} be the position vector of the wheel relative to the vehicle control point. Let the wheel radius



be r . It is simplest to *formulate the measurements in the body frame*. The velocity of the end of the wheel axle relative to the world is available from vector algebra as:

$$\begin{aligned}\vec{v} &= \vec{V} + \dot{\vec{p}} \times \vec{p} = V\hat{j} + \dot{\beta}\hat{k} \times (\rho_x\hat{i} + \rho_y\hat{j}) \\ \vec{v}_x &= -\dot{\beta}\rho_y\hat{i} \quad \vec{v}_y = (V + \dot{\beta}\rho_x)\hat{j}\end{aligned}$$

2.6.1 Steer Angle

Now the available measurements actually invert these relationships. First the steer angle α of the wheel and its gradient are:

$$\begin{aligned}\alpha &= \text{atan}(\sigma) = \text{atan}(v_y/v_x) \\ \frac{\partial \alpha}{\partial V} &= \frac{\partial \alpha}{\partial \sigma} \frac{\partial \sigma}{\partial V} = \left(\frac{1}{1 + \sigma^2} \right) \frac{1}{v_x} = - \left(\frac{1}{\dot{\beta}\rho_y} \right) \left(\frac{1}{1 + \sigma^2} \right) \\ \frac{\partial \alpha}{\partial \dot{\beta}} &= \frac{\partial \alpha}{\partial \sigma} \frac{\partial \sigma}{\partial \dot{\beta}} = \left(\frac{1}{1 + \sigma^2} \right) \left(\frac{\rho_y v_y + \rho_x v_x}{v_x^2} \right) = (V\rho_y) \left(\frac{1}{1 + \sigma^2} \right)\end{aligned}$$

This is a measurement of the ratio of angular to linear velocity and hence is a measure of curvature just as is the Ackerman steer angle.

2.6.2 Free Wheel Velocity

A free wheel will rotate automatically about the body z axis by the necessary steer angle due to friction. Its measurement relationship in radians is:

$$\dot{e} = -rU^v_x + \sqrt{U[(\dot{\beta}\rho_y)^2 + (\dot{\beta}\rho_x)^2]}$$

$$\frac{\partial \dot{e}}{\partial \mathbf{v}} = \frac{1}{r} \frac{1}{2v} \frac{1}{\sin(a)} \quad \frac{\partial \dot{e}}{\partial \mathbf{p}} = \frac{1}{r} \left(\frac{2v_x \rho_y + 2v_y \rho_x}{2v} \right) = \frac{1}{r} (v_x \cos \alpha + v_y \sin \alpha)$$

This is a measurement that responds to both the linear and angular velocity of the vehicle but they cannot be distinguished from a single measurement. The filter will automatically distinguish linear and angular velocity when two or more wheel velocities are measured.

2.6.3 Fixed Wheel Velocity

A fixed wheel will not rotate automatically about the body z axis. Its measurement relationship in radians is:

$$\dot{\theta} = \dot{\mathbf{v}} \cdot \hat{\mathbf{j}} = v_y / r = \frac{1}{r} (\mathbf{v} + \dot{\beta} \rho_x)$$

$$\frac{\partial \dot{\theta}}{\partial \mathbf{v}} = \frac{1}{r} \quad \frac{\partial \dot{\theta}}{\partial \dot{\beta}} = \frac{\rho_x}{r}$$

Again, this is a measurement that responds to both the linear and angular velocity of the vehicle but they cannot be distinguished from a single measurement. The filter will automatically distinguish linear and angular velocity when two or more wheel velocities are measured.

2.7 Complete PR Measurement Matrix

The complete measurement matrix for the 3D AHRS dead reckoning (without wheel encoders) system is then:

$$\bar{\mathbf{z}} = [\mathbf{v}_{enc} \ \mathbf{v}_{dop} \ \mathbf{w}_{com} \ \phi_{ahrs} \ \theta_{ahrs} \ \psi_{ahrs} \ \mathbf{a}]^T$$

$$\bar{\mathbf{x}} = [\mathbf{x} \ \mathbf{y} \ \mathbf{z} \ \mathbf{V} \ \mathbf{G} \ \phi \ \psi \ \mathbf{p}]^T$$

$$\frac{\partial \mathbf{a}}{\partial \mathbf{v}} = \frac{-\mathbf{L} \dot{\mathbf{p}}}{1 + (\mathbf{L} \dot{\mathbf{p}})^2}$$

$$\frac{\partial \mathbf{a}}{\partial \mathbf{p}} = \frac{\mathbf{L} \mathbf{V}}{1 + (\mathbf{L} \dot{\mathbf{p}})^2}$$

$$\mathbf{r}^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{\mathbf{d} \mathbf{a}}{\partial \mathbf{v}} & 0 & 0 & 0 & \frac{\partial \mathbf{a}}{\partial \mathbf{p}} \end{bmatrix}$$

If direct access to the attitude package is available, additional states can be added to the filter to model the dynamics of these sensors. As currently formulated, there is little coupling between the attitude reference and the odometry.

Section 3: AHRS Dead Reckoning Uncertainty Model

In theory, one must estimate every element of a covariance matrix in order to provide the filter with the information it needs. In practice, there are often correlated and systematic error sources which are roughly known in magnitude to far exceed any random errors but which are nonetheless not known well enough to model. Often, there is also no knowledge of correlation of error sources, and both the Q matrix and the R matrix are assumed to be diagonal.

$$Q = \text{diag} \begin{bmatrix} \sigma_y^2 & a_z^2 & a_v^2 & a_\theta^* & a_\phi^* & a_\psi^* & \sigma_\beta^2 \end{bmatrix}$$

$$R = \text{diag} \begin{bmatrix} a_{\text{enc}}^2 & a_{\text{dop}}^2 & a_{\text{com}}^2 & a_{\text{pitch}}^* & a_u^2 & a_n^2 & a_{\text{roll}}^2 & a_{\text{yaw}}^2 & a_{\text{steerj}}^2 \end{bmatrix}$$

The P matrix will evolve off diagonal terms naturally as the filter runs. Having made the **uncorrelated measurement error assumption**, the remaining issue is the estimation of systematic error sources for each state and each measurement.

A Kalman filter is a mathematical idealization that happens to be useful in practice. However, it is important to note that *there is a big difference between an optimal estimate and an accurate estimate*. In practical use, the uncertainty estimates take on the significance of *relative weights* of state estimates and measurements. So it is not so much important that uncertainty be absolutely correct as it is that it be relatively consistent across all models.

3.1 State Uncertainty

The state uncertainty model represents the disturbances which excite the linear system. Conceptually, it estimates how bad things can get when the system is run open loop (i.e with no sensors) for a given period of time. It is difficult to estimate these white sequences which drive the system model.

Given the assumption of low dynamics, and the neglecting of the entire control input, the errors associated with these assumptions far outweigh the effects of any truly random forcing function. In the absence of any other information, a plausible approach is to estimate error as the Taylor remainder⁶ in the dead reckoning equations, because, after all, dead reckoning is a truncated Taylor series in time. The Q_k matrix can be assumed diagonal, and its elements⁷ set to the predicted magnitude of the truncated terms in the constant velocity model. They can arise from:

- disturbances such as terrain following loads
- neglected control inputs such as sharp turns, braking or accelerating
- neglected derivatives in the dead reckoning model
- neglected states

6. Clearly, this approach is equivalent to associating the magnitude of the truncated term with the spectral amplitude of a white sequence and integrating over the time step to get the variance.

7. Remember that the elements in the covariance matrix are the variances, not the standard deviations, so the following expressions must all be squared when used.

3.1.1 The Gamma Matrix

The F matrix is convenient because the uncertainties of some of the variables can be represented in the body frame where they can be determined by intuition and they will be automatically converted as necessary. Let the T matrix be given by:

$$\mathbf{r} = \begin{bmatrix} \begin{bmatrix} \mathbf{R}_{b_i}^n \\ 0 \end{bmatrix} & \begin{bmatrix} \mathbf{O} \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \end{bmatrix} \quad \mathbf{Q} = \text{diag} \left[a^2, a^2, \sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_\phi^2, \sigma_\psi^2, \sigma_\beta^2 \right]$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{c}\phi & 0 & \mathbf{s}\phi \\ \mathbf{s}\phi & 0 & \mathbf{c}\phi \\ \mathbf{c}\phi & 0 & \mathbf{s}\phi \\ \mathbf{s}\phi & 0 & \mathbf{c}\phi \end{bmatrix} \quad \mathbf{r}_b^n = \begin{bmatrix} (\mathbf{C}\backslash/\mathbf{C}\backslash\mathbf{j})-\mathbf{S}\backslash(/ \mathbf{S}\mathbf{0}\mathbf{S}\mathbf{C}()) & -\mathbf{s}\backslash(/ \mathbf{c}\mathbf{0} & (\mathbf{C}\backslash(/ \mathbf{S}()) + \mathbf{S}\backslash(/ \mathbf{S}\mathbf{0}\mathbf{C}())) \\ (\mathbf{S}\mathbf{V}(/ \mathbf{C}()) + \mathbf{C}\backslash(/ \mathbf{S}\mathbf{0}\mathbf{S}()) & \mathbf{c}\backslash(/ \mathbf{c}\mathbf{0} & (\mathbf{S}\backslash(/ \mathbf{S}())-\mathbf{C}\backslash(/ \mathbf{J}/ \mathbf{S}\mathbf{0}\mathbf{C}())) \\ -\mathbf{c}\mathbf{0}\mathbf{s}\mathbf{c}() & \mathbf{s}\mathbf{0} & \mathbf{c}\mathbf{0}\mathbf{C}() \end{bmatrix}$$

The \mathbf{Q} matrix contains infinite entries as the vehicle approaches 90° of pitch because the "yaw" and "roll" axes become coincident then. This is an essential singularity of the Euler angle definition of 3D attitude.

3.1.2 Linear Position States

The translational uncertainty can be set to one-half the maximum acceleration times the square of the time step. This is the error expected when constant velocity is assumed. This gives:

$$\sigma_x = \sigma_y = \sigma_z = \frac{a_{\max} (\Delta t)^2}{2}$$

This error source alone is expected to grow roughly with the square root of the number of observations because:

$$\sigma_{\text{total}}^2(t) = \sum_{i=1}^n \sigma_i^2 = \frac{a_{\max}^2 (\Delta t)^2 n}{2}$$

For a sampling rate of 10 Hz, this gives about 10 meters of accumulated error per g of acceleration per hour of operation. For a rate of 1 Hz it is 300 meters per g per hour⁸. A very rough number for the maximum acceleration is about 1/10 g on roads and perhaps 1 g on rough terrain. The number is not truly acceleration - it represents the best guess for wheel slip, vehicle acceleration, and terrain disturbances.

8. This is the formal reason why velocity dead reckoning must cycle quickly on an accelerating vehicle.

A better model would also account for the fact that the errors vary greatly with their direction with respect to the vehicle. Let the maximum acceleration be a vector quantity. Its components directed along the body x, y, and z axes are roughly weighted as follows:

$$\bar{a}_{\max} = \begin{bmatrix} 0.1a_{\max} & 1.0a_{\max} & 0.3a_{\max} \end{bmatrix}^T$$

A basic assumption of encoder dead reckoning is that vehicle motion is always aligned with the forward body axis. This has been called the **principle motion assumption**. It is violated in situations where the wheels slip laterally and it is violated when the local terrain tangent plane is not aligned with the body frame (i.e when the vehicle fails to follow the terrain). One important special case of the latter is the deflections of the vehicle suspension. This is a very large error source which is appropriately modelled in the state equations and not in the sensors, since it amounts to a modelling assumption. A nav frame formulation would not suffer from this type of error.

Nevertheless, it is generally the case that a vehicle does not slide sideways or accelerate upward very quickly, so this model takes this into account. The uncertainties expressed in the nav frame are then automatically computed by the F matrix.

3.1.3 Angular Position States

For the angular position states, there are no measurements of velocity in the DR equations for all axes, so the truncation error is a velocity term. These uncertainties cannot be left at zero because the filter will first not project forward in the state equations, and secondly, compute zero gains for the measurements. The states will not move at all. When the measurement residuals are computed for these, the state vector is one cycle old and has not moved. A very rough error estimate for these is:

$$\sigma_0 = \sigma_{\phi} = \sigma_{\psi} = \Omega_{\max} \Delta t$$

$$\sigma_{\text{total}}(t) = J \dot{t} \sigma = \frac{\Omega_{\max} \Delta t \sqrt{t/(\Delta t)}}{2} = \frac{27 t f \Delta t V t/(\Delta t)}{2}$$

Where f is the estimated highest frequency of attitude changes of the vehicle body. Using a cycle rate of 10 Hz and a frequency of 1/24 Hz or 15° = 1/4 rad per second, this gives about 4 rads per hour of operation. This may sound unreasonable, but the issue at hand is the error in predicting constant attitude for a vehicle which can rotate at 15° per second over an hour with no measurement. This simply tells the filter to trust the readings of the sensors and forces the angular states to track the sensors.

A more refined model could make explicit use of the assumption that the vehicle cannot pitch or roll more than about 15°, but this is not really necessary unless the sensors have drift problems⁹. One addition to the model is to account for the fact that the errors vary greatly with their direction with respect to the vehicle. Let the maximum angular velocity be a vector quantity. Its components directed along the body x, y, and z axes are roughly weighted as follows:

$$\Omega_{\text{max}} = \begin{bmatrix} 0.3\Omega_{\text{max}} & 0.3\Omega_{\text{max}} & 1.0\Omega_{\text{max}} \end{bmatrix}^T$$

where it is assumed that the primary direction of rotation is about the body z axis.

3.1.4 Linear Velocity States

In the case of linear velocity, there are no acceleration states which propagate it forward in time via the transition matrix, so this state will not move if its uncertainty is set to zero. Again, using the remainder theorem:

$$\begin{aligned} {}^\circ V &= {}^a \text{max}^{At} \\ {}^\circ \text{total}^{(t)} &= \wedge {}^\circ i = \frac{{}^a \text{max}^{At} A/t/ \sqrt{<At>}}{2} = \frac{{}^a \text{imax} \Delta t \sqrt{t/(\Delta t)}}{2} \end{aligned}$$

This gives 10 meters of error for an hour of operation at 10 Hz cycling rate using 1 g.

3.1.5 Angular Velocity States

Following the technique, an estimate of the maximum angular acceleration of the vehicle is needed in order to estimate the truncation error. By Euler's equation, this is the ratio of the applied torque to the moment of inertia, assuming a diagonal inertia dyadic. Numbers for these quantities are hard to come by. One way to get a reasonable number is to assume that the angular velocity cannot change any faster than zero to the maximum in less than some number of seconds, which amounts to an assumption about the magnitude of the disturbance loads. This would give:

$$\alpha_{\text{max}} = \Omega_{\text{max}}/\tau$$

where **T** is an adjustable "time constant". Then the angular velocity state is uncertain at:

$$\sigma_{\dot{\beta}} = \alpha_{\text{max}} \Delta t$$

9. As an idea for dealing with drift, it may be possible to add fake measurements of zero attitude with high uncertainty. This may have the useful side effect of enforcing level indications over the long term while still allowing the attitude sensors to capture dynamics. This idea is used in inertial systems which slave the gyros to the accelerometers because they indicate the direction of gravity.

3.2 Measurement Uncertainty

The measurement uncertainties are far more critical to the filter operation, because, after all, the whole system is considered to fail if sensors are lost for only a few seconds and filter optimality is not an issue. For sensors models, some of the error sources to be estimated include:

- bias and scale instability
- neglected sensor dynamics
- noise, vibration, backlash, EMI, and compliance, etc.

This section will model all sources of error as if they were random. A better form of model would augment the state vector to *include bias and scale factor states* for some of these sensors to allow the filter to tune itself. Of course, such states will require appropriate sensor redundancy, so they are not included in this general formulation.

3.2.1 Encoder

The random error in the encoder measurement can be estimated by driving the vehicle at constant velocity and plotting the encoder output. Since the vehicle is a filter on speed inputs, any vibrations or electrical noise will manifest itself in such an experiment.

However, there are far more important systematic sources. It is expected that scale errors will predominate because the wheel radius actually varies with time and temperature, compliance has been an important historical issue, and there is of course, longitudinal wheel slip. Since the assumed error is a scale factor error, it is modelled as a fraction of the measurement itself. This has the property that the uncertainty is correctly increased when the time step increases.

A final issue with encoders is that encoder readings must be either time-stamped or read synchronously. Otherwise the velocity computed will be in error by the degree to which the filter cycle time is asynchronous¹⁰. In a non real-time testing scenario, this error can be several hundred percent, so it must be addressed. With these caveats, the uncertainty model is:

$$\sigma_{\text{enc}} = \text{SFE}_{\text{enc}} V_{\text{enc}}$$

From experience, about 5% of total distance travelled is appropriate for the position states. However, the issue here is the error in the encoder estimate of the forward component of velocity. So something less than 5% is appropriate unless there are significant errors beyond scale errors.

3.2.2 Doppler

Without more information about a specific Doppler device, little can be said about the nature of its systematic error sources after calibration. There is likely a velocity dependence, and hence a scale error, and it would be surprising if it performs significantly better than a well-calibrated encoder on pavement. Hence, tentatively set:

$$\sigma_{\text{dop}} = \text{SFE}_{\text{dop}} V_{\text{dop}}$$

¹⁰. The encoder is the only sensor for which this is necessary. The dt used by it must be the real dt between when the sensor was physically read. For the state equations, it is appropriate to read the clock just before execution and subtract it from last time around to get the dt.

3.2.3 Attitude

All of the attitude sensors will be considered as a unit. It is difficult to quantify the attitude uncertainty without more knowledge of the components inside. It is expected that considerable systematic error will be present which will have complicated dynamics, so an output noise test may not be very meaningful.

In the absence of information, the best that can be done is to assume some constant uncertainty. However, it is common to find that yaw uncertainty is worse than that of the other two angles, because the former is often based on the gravity vector, and the latter on the weak and unreliable local magnetic field so tentatively set:

$\sigma_{\theta} = \sigma_{\phi} = \sigma_{ATT}$	$\sigma_{\psi} = 2\sigma_{ATT}$
--	---------------------------------

Some well known issues for attitude sensors are listed below for consideration in the estimates.

3.2.3.1 Accelerometers and Inclinometers

Both of these devices function by measuring the deflection of a mass attached to a calibrated restraint where the indicated degree of freedom is either linear or rotary. As a matter of basic physics¹¹, no instrument can instantaneously distinguish acceleration from gravity.

Inclinometers will measure vehicle linear acceleration as well as the direction of gravity, so their dynamics may have to be considered depending on their transfer function. Oftentimes, inclinometers intended for use on vehicles are constructed as low-pass filters for this reason. The filtering of inclinometer outputs has the side effect of filtering vehicle attitude dynamics from the inclinometer signal as well. Based upon the time constant of the inclinometer filter, a Taylor remainder analysis may be appropriate.

Accelerometer measurements of acceleration are corrupted by gravity and inertial forces due to the earth’s rotation if they are low-pass devices. The technology of accounting for this leads to the inertial navigation system before the accounting is complete.

It is possible to account for the corruption of attitude by centrifugal acceleration in inclinometers by changing the measurement relationship to reflect that the measurement is the sum of either pitch and forward acceleration or roll and lateral acceleration.

3.2.3.2 Gyroscopes

The rate gyro is a nice complement to the inclinometer because it responds well to high frequency inputs. Gyro drift is the major source of error and it arises from very small disturbance torques generated on the device rotary bearings, whatever their nature, whether the devices are strapdown or not. Frequency response and drift models may be available from the manufacturer. Gyro drift rate is well modelled by a combination of random bias, exponentially-correlated error, and random walk. It has also been suggested that a random ramp is necessary.

11. Specifically, Einstein’s principle of relativity implies that a man in an elevator in free-fall in a gravitational field can never know he is falling based on any measurement conducted in the elevator frame. Thus, accelerometers confuse gravity for acceleration, and inclinometers confuse acceleration for gravity.

3.2.3.3 *Compasses*

The magnetic compass comes in various forms, but regardless of the form, there are three issues to consider. First, the **variation** of the local geomagnetic field and geographic north is a constant for small excursions that can be easily calibrated out¹². Second, the term **deviation** is reserved for the effects of the vehicle residual magnetic field. The geomagnetic field is fixed in the earth frame whereas the vehicle field is fixed in the vehicle frame. Hence, the device measures the time-varying sum of the two. A short Fourier series in vehicle heading can be constructed to calibrate this out against a reference by spinning the vehicle one full turn in azimuth.

Heeling error is the term for the error induced when the vehicle approaches vertical because the local field no longer projects onto the sensor sensitive axis. For fluxgates, heeling error is not an issue. However, fluxgates require nontrivial settling times, so they have the same filtering property on heading as clinometers do on pitch and roll. Therefore, a heading gyro is often necessary to augment a compass in even moderate angular velocity vehicles.

3.2.4 **Steering**

Steering wheel position can be measured with any number of simple transducers. It is typically a very low-fidelity measurement that does not benefit from overly precise error characterization. Let:

$$a_{\text{steer}} = \text{constant}$$

where the constant is determined from a linearity test.

12. The variation for the eastern seaboard of the United States is about 15° east of north. However, this is an outdoor assumption for an isolated compass. Any magnetic sources close to the vehicle will distort the local field. My thanks to R. Coulter for educating me on compasses.

Section 4; Aided AHRS Dead Reckoning

The pure dead reckoning filter of the previous section is unlikely to achieve an accuracy which exceeds a few percent of the distance travelled. This is because of the essential integration of errors in the process of dead reckoning. In particular, notice that none of the measurement matrices of the previous section have a nonzero element in the first three columns. That is, *none of them measures the vehicle position* - even indirectly. They all measure attitude or derivatives of position.

Whatever the fidelity of the measurements used in practical dead reckoning, a fix is needed at regular intervals to damp the DR and the mechanism for doing this is the subject of this section. The Kalman filter is an ideal formalism for integration of dead reckoning and position fixes because fixes are simply additional measurements which can be folded into the equations in like manner to the DR measurements.

4.1 Fixes in the Navigation Frame

The simplest form of position fix is a direct measurement of the vehicle position in the navigation frame. In practice, the **survey point** is the only such fix available because position indicating devices cannot usually be mounted at the center of the body frame. Here, the vehicle is positioned at a point which has been presurveyed in the nav frame. Once the filter is told that this is the case, it can use its stored knowledge of the true coordinates of the survey point to generate the fix. Survey points are useful for development as well as operational purposes since they can be used to calibrate the system models.

When generating survey points using a positioning device, it is important to place it in exactly the same place on the vehicle each time it is used, and to account for its position explicitly.

A distinctive turn in the trajectory may constitute a survey point. Such **path features** can be used provided:

- there is some mechanism to ensure that the vehicle is actually on the stored trajectory (say, a human driver)
- there is some mechanism to recognize when the feature is encountered

The measurement matrix for survey points is trivial:

$$\bar{\mathbf{z}} = \begin{bmatrix} x_{sp} & y_{sp} & z_{sp} \end{bmatrix}^T \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4.2 Fixes in a Positioner Frame

Sensors which can provide fixes on their own position include the GPS receiver¹³ and the inertial navigation system. The measurement matrix for such a sensor is relatively trivial:

$$\bar{z} = T_b^p \begin{bmatrix} x_{\text{gps}} & y_{\text{gps}} & z_{\text{gps}} \end{bmatrix}^T \quad H = T_b^p \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Estimation of GPS uncertainty is a difficult matter because it depends on such matters as the presence or absence of Selective Availability (the dominant source of error), various technological error sources such as receiver noise, dynamics, multipath, etc., and the satellite geometry. Ideally, a receiver would provide the covariance matrix in the navigation frame directly. Without such information, the only option is to assume a constant covariance which is relatively conservative. In the case of civilian GPS receivers, position error is highly correlated, and a correlation model is appropriate.

Estimation of INS uncertainty is even a harder problem because of the Schuler dynamics involved. Again, a large number is better than no number. Technically, the INS is a DR device, and it does not actually generate a “fix”. The implication of this is that INS uncertainty models must have a time growth associated with them.

13. In truth, the GPS receiver has a complicated measurement model which is four-dimensional range triangulation. It is possible to ignore this if an estimate of uncertainty in the GPS output is available. Even the GDOP output of a receiver can be used somewhat.

Section 5: Roadfollower Aiding

A roadfollowing system provides an observable which can be described as **relative crosstrack error**. It is measured relative to the path in view (the road), more or less transverse to the path. It is usually measured in a visual feedforward sense (i.e it is the crosstrack error in front of the vehicle).

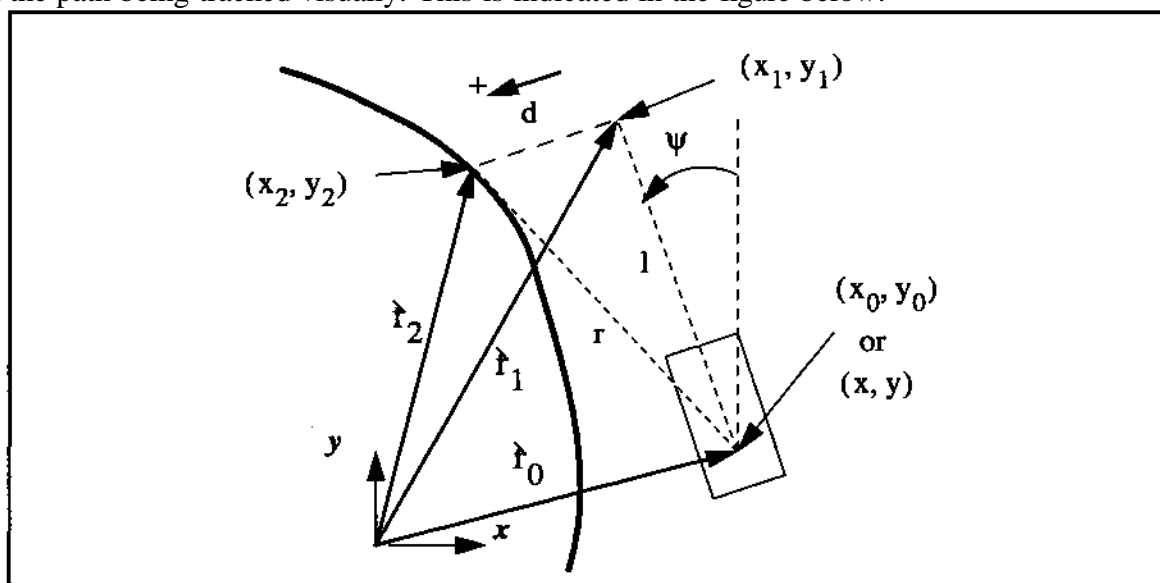
An EKF can accept any indirect measurement of state and since the road deviation in an image is dependent on the vehicle position and attitude, it provides an indirect measurement of state. Integration of a road follower into the EKF amounts to a continuous landmark observable. Note however that this will not improve the performance of the roadfollower - it has direct visual feedback already, but it should improve the position estimate of the vehicle.

5.1 Measurement Model

The measurement model must be formulated in an indirect sense - that is, the relationship between the path and the vehicle state which generates the observable is required. In reality, the observable depends on the complete six-axis pose of the vehicle, but because a typical roadfollower does not make such distinctions and typically does not store a 3D path, a 2D formulation will be used.

5.1.1 Observable

Let **l** be the **lookahead distance**, and let the roadfollower generate the crosstrack error **d** at the lookahead distance that *would be observed if the vehicle travelled in a straight line on its current heading*. Let point 1 be at the end of the lookahead vector and point 2 be the corresponding point on the path being tracked visually. This is indicated in the figure below:



Define the vectors:

$$\mathbf{r} = \mathbf{r}_{20} = \mathbf{r}_2 - \mathbf{r}_0 \quad \mathbf{l} = \mathbf{r}_{10} = \mathbf{r}_1 - \mathbf{r}_0 \quad \mathbf{d} = \mathbf{r}_{21} = \mathbf{r}_2 - \mathbf{r}_1$$

and the letters r , l and d will refer to the magnitudes of these vectors except that the latter will be a signed quantity. The vector l will be called the **lookahead vector**, d will be called the **crosstrack vector**, and i will be called the **goal vector**. The observable is clearly:

$$h(\bar{x}) = d = \pm |r_{21}| = \pm \sqrt{r_{21} \bullet r_{21}} = \pm \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

The crosstrack observable d will be defined to be positive *when the path is to the left* of the end of the lookahead vector as shown in the figure. This depends on the sign of the angle *from* the lookahead vector *to* the goal vector, and it is positive when this angle is counterclockwise. This can be determined without using the arctangent by a vector cross product as follows:

$$\begin{aligned} \text{signof}(lx^*) &= \text{signof}(l_{Ay}r_{y} - l_{y}r_{Ax}) \\ \text{signof}(lx^{\wedge}) &= \text{signof}[(x^{\wedge}x)(y_2 - y) - (y^{\wedge}y)(x_2 - x)] \end{aligned}$$

The coordinates of point 1 depend on the vehicle state vector thus:

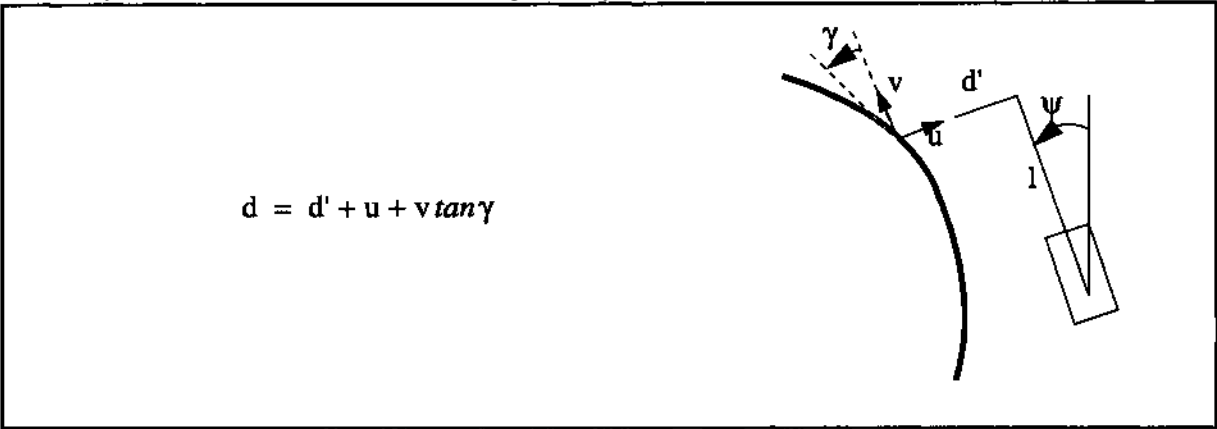
$$\begin{aligned} x_1 &= x - l \sin ||f \\ y_1 &= y + l \cos ||f \end{aligned}$$

This gives the observable as an indirect measurement of vehicle position and orientation in 2D.

5.1.2 Measurement Jacobian

The measurement Jacobian is available through partial differentiation - which is a difficult problem in this case. Note that as the vehicle moves in x, y, or yaw, both the end of the lookahead vector and the point (x₂,y₂) move in an uncorrelated manner. It is useful to establish a (u,v) coordinate system at the current point (x₂,y₂) and consider both of these effects separately.

Consider a differential motion of the vehicle in x and y and yaw which causes the crosstrack point to separate from the path. The differential change in the crosstrack observable is the change in length required to reattach it. If the old crosstrack is d^f, and the tangent of the path relative to the v axis is γ, then the new crosstrack d is given by:



The (u,v) coordinates of the vehicle are:

$$\begin{aligned} u &= (\mathbf{x} - \mathbf{x}_2) \cos \gamma + (\mathbf{y} - \mathbf{y}_2) \sin \gamma \\ v &= -(\mathbf{x} - \mathbf{x}_2) \sin \gamma + (\mathbf{y} - \mathbf{y}_2) \cos \gamma \end{aligned}$$

So the measurement Jacobian is given by:

$$\begin{aligned} \frac{\partial d}{\partial \mathbf{x}} &= \frac{\partial d}{\partial u} \frac{\partial u}{\partial \mathbf{x}} + \frac{\partial d}{\partial v} \frac{\partial v}{\partial \mathbf{x}} = \cos \gamma - \tan \gamma \sin \gamma = \cos(\gamma + \psi) / \cos \gamma \\ \frac{\partial d}{\partial \mathbf{y}} &= \frac{\partial d}{\partial u} \frac{\partial u}{\partial \mathbf{y}} + \frac{\partial d}{\partial v} \frac{\partial v}{\partial \mathbf{y}} = \sin \gamma + \tan \gamma \cos \gamma = \sin(\gamma + \psi) / \cos \gamma \\ \frac{\partial d}{\partial \psi} &= \frac{\partial d}{\partial u} \frac{\partial u}{\partial \psi} + \frac{\partial d}{\partial v} \frac{\partial v}{\partial \psi} = -1 - d \tan \gamma \quad \text{Because:} \\ \frac{\partial u}{\partial \psi} &= -(\mathbf{x} - \mathbf{x}_2) \sin \gamma + (\mathbf{y} - \mathbf{y}_2) \cos \gamma = v \\ \frac{\partial v}{\partial \psi} &= -(\mathbf{x} - \mathbf{x}_2) \cos \gamma - (\mathbf{y} - \mathbf{y}_2) \sin \gamma = -u \end{aligned}$$

5.2 Predictive Measurement

Recall the state update equation of the EKF:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)]$$

This equation runs in the filter when a measurement arrives. It amounts to the formation of a measurement residual - the difference between what is observed \mathbf{z}_k and what was expected based on the current state estimate $\mathbf{h}(\hat{\mathbf{x}}_k^-)$.

In order to generate the prediction, the system must:

- know the coordinates of every point on the path in the navigation frame
- be able to search the path forward to generate the crosstrack prediction

Let $\hat{\mathbf{u}}$ be a unit vector oriented along the lookahead vector:

$$\mathbf{u} = [-\sin\psi \quad \cos\psi]$$

Then the lookahead distance for any point (x_2, y_2) on the path is:

$$l = \mathbf{r} \cdot \hat{\mathbf{u}} = (x_2 - x) (\sin\psi) + (y_2 - y) (\cos\psi)$$

The predictive measurement can be obtained by searching forward from a point on the path near the vehicle until the above expression is equal to the required lookahead distance. Then the formula for d gives the predictive measurement.

5.3 Advantages

The primary advantage of the road-following observable is that it constitutes the only landmark observable in the absence of GPS or visual landmarks. This element of the Kalman filter will force the state estimate to track the known or recorded position of the path.

Section 6: Terrain-Aided AHRS Dead Reckoning

Vehicle perception sensors provide many mechanisms for generating a fix. Landmark recognition can be folded into the filter, but the measurement model is nontrivial because it must account for the imaging process. This section and the following three present a general model of the perception measurement process which is applicable to many different scenarios.

6.1 Terrain Aids

Terrain-aiding schemes can be distinguished and classified along several independent dimensions:

6.1.1 Image Dimension

The image dimension amounts to the size of the measurement vector. Some options include:

- 3D scanning laser rangefinder
- 2D scanning laser rangefinder
- 3D stereo range images
- 2D color images
- 2D intensity images

An important distinction of the last two is that their imaging transform is not invertible. In this case, it is not possible to use a single image to convert the image coordinates of a landmark into its coordinates in the nav frame. However, because the Kalman filter is formulated in terms of sensing and not perception, this is not overly important. Underdetermined measurements of state are quite legal.

6.1.2 Image Information Density

Schemes can also be classified based on the amount of information extracted from a single image:

- **correlation** schemes match pieces of images in order to generate a single vector measurement
- **feature-based** schemes match a small number of "interesting" points to generate a small number of vector measurements
- **iconic** schemes match every pixel in the image in order to generate a large number of vector measurements

6.1.3 Absolute and Differential Observations

Schemes can be classified based upon whether landmarks are considered known in the nav frame or whether they are considered unknown in the nav frame and observed to change from image to image:

- **absolute observations** match observations against the known position of a "landmark" in the nav frame
- **differential observations** match observations in the current image against either their position in the last image or their predicted position in the current image

6.2 Measurement Uncertainty

Uncertainty models for perception sensors can be generated from knowledge of the physics of the operation of the sensor and the conditioning of the embedded computations, if any. Measurement uncertainty is modelled in the R_x matrix. Some specific cases are:

6.2.1 AM Laser Rangefinders

These devices have a well-known variation in range accuracy which varies directly with the square of the range and inversely with the angle of incidence and the reflectance of the surface. Angular uncertainty can be estimated as the width of a pixel but may also include an accounting for the fidelity of the mirror drive control laws.

6.2.2 TOF Laser Rangefinders

For time of flight devices, variation with range is probably a function of the range quantization and terrain incidence angle. Variation with reflectance and angle of incidence can be accounted for by considering the operation of the thresholding electronics. Similar comments about angular uncertainty apply.

6.2.3 Stereo Vision Rangefinders

Stereo uncertainty can be generated from a knowledge of the particular algorithm used, the baseline, and the sensory hardware. It can sometimes be produced as a by-product of the stereo algorithm.

6.2.4 CCD Images

For CCD images, a model of the transformation between intensity noise and the noise on the measured angle to a feature is required. The basic issue is the fidelity of localization of the landmark in the image. Accuracies larger or considerably smaller than a pixel are possible depending on the feature extraction algorithm used.

Section 7: Absolute Landmark Recognition

This section and the following two will present models for various terrain-aiding techniques. In all cases, matrix notation is used to abstract away the details of the kinematics of various sensors. These transforms and their Jacobians are given in the appendices.

7.1 Absolute Landmarks

Absolute landmark recognition is distinguished by the true knowledge of the landmark position in the nav frame. This distinction appears in the interpretation of the predictive measurement $h(\hat{\mathbf{x}}_k^-)$ in the state update equation. In this case, the predictive measurement arises from “simulating” the generation of the landmark position in the image according to the measurement model evaluated at the current vehicle position and the known landmark position. Many special cases of absolute aiding can be distinguished.

7.1.1 3D Landmark Recognition with a Rangefinder

If a rangefinder is used, the rangefinder imaging Jacobian is used. In this case, the sensor to image transform is spherical polar and the measurement is 3D, so it is $3 \times n$.

7.1.2 2D Landmark Recognition with a Video Camera

If a camera is used, the camera imaging Jacobian applies. In this case, the sensor to image transform is a perspective projection and the measurement is 2D, so it is $2 \times n$.

7.1.3 Video Crosstrack Observations

A degenerate example of the above landmark recognition is the use of the crosstrack error observable generated by a road-follower. This case was presented earlier in detail.

7.1.4 Convoy Image Sequences

If sufficient radio bandwidth is available, following vehicles in a convoy can maintain a kind of **video lock** by computing the position of the correlation peak over a small window between the current image and the image generated by the front vehicle when it was close to the current position. Relative crosstrack and alongtrack error should be easy to compute in this case¹⁴.

7.1.5 2D Point and Line Feature Matching

For indoor 2D applications, it is possible to match a wall of known shape against the current range image from a single line rangefinder. Based on the current position estimate and a known map of the environment, every pixel in a single line rangefinder constitutes a measurement whose residual can be used to update the position estimate. For 2D problems, the state vector is only 3×1 and no matrix inversion is required.

14. Both of the previous scenarios will benefit if only the lower image pixels are used because the parallax observable will be largest and more useful in this region of the image. Of course, the entire image will not correlate anyway because of the perspective geometry of the imaging process. If they did, stereo would never work.

7.2 Feature-Based Scheme

In a feature-based scheme, an interest operator identifies interesting points in an image and a mechanism is provided which constructs candidate matches of landmarks against their predicted positions in the image.

Let the position of a landmark in the navigation frame be known to be:

$$\bar{\mathbf{r}}_L^n = \begin{bmatrix} x_L & y_L & z_L \end{bmatrix}^T$$

Consider a generalized perception sensor which generates a 3D image of which most real sensors are special cases. Such a sensor can be modelled as generating the landmark position somewhere in the image through the image formation process expressed in terms of a concatenation of transformations through intermediate frames.

The transformation from the navigation to the sensor frame is:

$$\bar{\mathbf{r}}_L^s = \begin{bmatrix} x_L^s \\ y_L^s \\ z_L^s \end{bmatrix} = \mathbf{T}_b^s \mathbf{T}_n^b(\bar{\mathbf{x}}) \bar{\mathbf{r}}_L^n$$

Where $\mathbf{T}_n^b(\bar{\mathbf{x}})$ is the nav frame to body frame homogeneous transform, and \mathbf{T}_b^s is the body frame to sensor frame homogeneous transform. The measurement matrix for the transform is evaluated as the product of the constant body to sensor transform, the Jacobian tensor of the nav to body transform, and the landmark position vector.

$$\mathbf{H}_L^s = \frac{\partial}{\partial \bar{\mathbf{x}}} \left(\bar{\mathbf{r}}_L^s \right) = \mathbf{T}_b^s \frac{\partial}{\partial \bar{\mathbf{x}}} \left(\mathbf{T}_n^b(\bar{\mathbf{x}}) \right) \bar{\mathbf{r}}_L^n$$

Now to express the generated image, let a generalized nonlinear imaging function map a point in the sensor frame into image coordinates:

$$\bar{\mathbf{r}}_L^i = \mathbf{f} \left(\bar{\mathbf{r}}_L^s \right) = \mathbf{f} \left(\mathbf{T}_b^s \mathbf{T}_n^b(\bar{\mathbf{x}}) \bar{\mathbf{r}}_L^n \right)$$

where \mathbf{f} is the triple (range, azimuth, elevation) for a rangefinder or the pair (row, column) for a video camera. The complete measurement Jacobian is then given by the chain rule:

$$\mathbf{H}_L = \left| \frac{\partial}{\partial \bar{\mathbf{x}}} \left(\bar{\mathbf{r}}_L^i \right) \right| = \mathbf{H}_s^i \mathbf{H}_L^s = \mathbf{H}_s^i \mathbf{T}_b^s \frac{\partial}{\partial \bar{\mathbf{x}}} \left(\mathbf{T}_n^b(\bar{\mathbf{x}}) \right) \bar{\mathbf{r}}_L^n$$

This is the general case for any sensor.

Recall that the measurement Jacobian provides the information necessary to project the residual onto the state vector. The measurement uncertainty itself arises in the R_k matrix. So the analysis thus far has nothing to do with the sensor itself. Rather, it answers the question of how an error in vehicle position relates to an error in the position of a landmark in the image for a perfect sensor.

This will be called the **landmark Jacobian**. In order to use this formula for any particular sensor, the imaging Jacobian must be substituted for the particular sensor used.

The matrix partial is a tensor. Let it be $4 \times 4 \times n$. Its second index is matched with the row index of the landmark position vector to generate the $4 \times n$ matrix:

$$H_{i,k} = \left[\frac{\partial T}{\partial \bar{x}} \right]_{i,j,k} \bar{r}_j \quad \begin{matrix} i=1,4 \\ j=1,4 \\ k=1,n \end{matrix}$$

Notice that the fact that the filter is using the difference between the projected position of the landmark and the actual position of the landmark in the image is not explicit. The operation of the filter is such that it automatically computes what the differential change in the state vector has to be in order for the observed measurement to be made.

7.3 Uncertain Absolute Landmarks

The above treatment of landmark recognition assumes that the landmark position is known precisely in the navigation frame. However, in some cases, the landmark has its own uncertainty. This issue can be managed in the general formulation because the R_k matrix exists to quantify measurement uncertainty. However, there is some difficulty involved because the R_k matrix is expressed in image coordinates. Recall that the transformation of the landmark position into the image plane is given by:

$$\bar{r}_L^i = f(\bar{r}_L^s) = f(T_b^s T_n^b(\bar{x}) \bar{r}_L^n)$$

Thus, the uncertainty in the image plane arising from an uncertain landmark can be computed as

$$\begin{aligned} \Delta \bar{r}_L^i &= \frac{\partial}{\partial \bar{r}_L^n} \left[f(T_b^s T_n^b(\bar{x}) \bar{r}_L^n) \right] \Delta \bar{r}_L^n \\ \Delta \bar{r}_L^i &= H_s^i T_b^s T_n^b(\bar{x}) \Delta \bar{r}_L^n \end{aligned}$$

where the chain rule has been used and it was noticed that the argument to the imaging function is linear in the landmark position in the nav frame. Now from the expectation operator:

$$\begin{aligned} E[(\Delta \bar{r}_L^i)(\Delta \bar{r}_L^i)^T] &= E\left[\left(H_s^i T_b^s T_n^b(\bar{x}) \Delta \bar{r}_L^n\right)\left(H_s^i T_b^s T_n^b(\bar{x}) \Delta \bar{r}_L^n\right)^T\right] \\ \text{Cov}[\Delta \bar{r}_L^i] &= H_s^i T_b^s T_n^b(\bar{x}) \text{Cov}[\Delta \bar{r}_L^n] \left(H_s^i T_b^s T_n^b(\bar{x})\right)^T \end{aligned}$$

So that if C_L is the covariance of the landmark expressed in the navigation frame, then the R_k matrix for the measurement is:

$$R_k = H C_L H^T$$

$$H = H_s^i T_b^s T_n^b(\bar{x})$$

This analysis produces the contribution of landmark uncertainty to the R_k matrix. The effect of sensor uncertainty must be (literally) added to this, and the effect of vehicle position uncertainty is already accounted for in the measurement Jacobian.

In this formulation of the uncertain landmark, the landmark position is not considered part of the state vector and, hence, is not updated. It is possible to augment the state vector to include the landmark position. This will be discussed in a later section.

7.4 Iconic Scheme

In an iconic absolute landmark scheme, the “landmarks” are really a continuous geometric description of all or part of the environment which is known *a priori*. In 2D, a line segment world description or a collection of occupancy points suffices. In the latter case, and in general, it may be necessary to interpolate a sparse *a priori* model of world geometry in order to generate the predictive measurements. The predictive measurements are generated by “simulating” the image that would be expected if evaluated at the current position estimate and the known world geometry model.

Such a scheme implemented in 3D amounts to a least squares continuous match of the entire terrain map from a single image against the known continuous world model.

Section 8: Differential Landmark Recognition

Differential aids have a very close connection to **mapping**¹⁵ applications. The distinction is based on whether the vehicle position or the landmark position (or both) is considered unknown. That is, whether one or the other or both appear in the state vector. It is possible to consider both unknown if the uncertainties are treated correctly and the state vector is augmented. This is considered in the next section.

In differential applications, if the vehicle position estimate is updated *before* the image is transformed into a map section, then *mapping is performed simultaneously* with position estimation. After all, the filter is attempting to minimize the residual in order to produce the optimal position estimate, but in this case, the residual is the map mismatch, so it is directly minimized. This can only be done, of course, when an invertible sensor transform is used, but this is necessary to form the terrain map anyway, so it is not an extra requirement.

8.1 Differential Landmarks

When landmarks are observed differentially, the predictive measurement is used to generate a residual between *where it should appear* based on the last time it was seen, and *where it does appear* in the current image. Under the measurement models for sensors used here, the residual is formed in image space. Nav frame schemes are possible too, but they are a mathematical equivalent where the transforms in the measurement models are just moved around from the H_k matrix to the R_k matrix.

Again, iconic and feature-based schemes are possible and these can be distinguished by the interpretation of the predictive measurement $h[\hat{x}_k^-]$ in the state update equation:

$$\hat{x}_k = \hat{x}_k^- + K_k [z_k - h(\hat{x}_k^-)] = \hat{x}_k^- + K_k [z_k - \hat{z}_k^-]$$

8.2 Feature-Based Scheme

Here, discrete features are used and matched from image to image. This is done by tracking the nav frame positions of the landmarks and transforming them into image space as needed.

After dead reckoning has supplied a position estimate, the predictive measurement can be simply generated through the measurement model evaluated at the current position estimate:

$$\hat{z}_k = T_L \left(T_b^s T_n^b (\bar{x}_k) \hat{r}_L^n \right) \Big|_k$$

8.3 Iconic Scheme

Iconic schemes *require* a map unless images are acquired fast enough that flow information is high enough in fidelity to justify the linear assumption of the filter. When a map is available, the predictive measurement for every pixel in the image is generated by "simulating" what the sensor should see in the current position given by the current position estimate. This simulated image is generated from the evolving map as it exists before the current image is incorporated.

¹⁵ The *map matching* of local terrain map navigation is a trivial kind of mapping.

Thus, considering the computed point of intersection of the image ray with the terrain map to constitute the "landmark" $\hat{\mathbf{r}}_L$, the measurement model is:

$$\hat{\mathbf{z}}_k = \hat{\mathbf{r}}_L^i = \mathbf{f}\left(\hat{\mathbf{r}}_L^s\right) = \mathbf{f}\left(\mathbf{T}_b^s \mathbf{T}_n^b(\bar{\mathbf{x}}) \hat{\mathbf{r}}_L^n\right)$$

where the notation suggests that the landmark position is predicted.

Complete iconic schemes are probably only feasible in 2D worlds with 2D sensors because the simulation is expensive computationally. However, it is also possible to undersample the image to reduce the load in the 3D case..

8.4 Examples of Differential Aids

8.4.1 3D Terrain Aided Navigation

In a manner very similar to video lock, a rangefinder navigation system can use the mismatch difference between two sequential terrain maps in order to update the vehicle position estimate. If the range images are close together in time, the computation may be poorly conditioned, so it is probably best to compute the mismatch over the largest excursion for which two images still overlap. In this scenario, the position of a feature from the last image forms the landmark position and the position in the current image forms the measurement. It would be very computationally intensive to process every range pixel in this manner, so a feature extraction process will be a practical necessity.

8.4.2 2D Point and Line Feature Matching

For indoor 2D applications, the evolving map may be a set of geometric primitives such as lines or simply a set of occupancy points. When the map is interpolated, a mechanism for simulating the image expected is available. Based on the current position estimate and the evolving map of the environment, every pixel in a single line rangefinder constitutes a measurement whose residual can be used to update the position estimate.

Section 9: Simultaneous Mapping and Position Estimation

With an augmented state vector, it is possible to "distort" an evolving map in order to produce a better estimate of both the vehicle position and the environmental map. This is achieved, of course, with state vector augmentation.

9.1 State Vector Augmentation

The augmented state vector includes the vehicle position estimate as well as the position of the landmark in the nav frame:

$$\bar{\mathbf{x}}_n = [\mathbf{x} \ \mathbf{y} \ \mathbf{z} \ \mathbf{V} \ \theta \ \phi \ \psi \ \dot{\beta}]^T \quad \bar{\mathbf{x}}_L = [\mathbf{r}_x^n \ \mathbf{r}_y^n \ \mathbf{r}_z^n]^T$$

The system model then becomes:

$$\frac{d}{dt} \begin{bmatrix} \bar{\mathbf{x}}_n \\ \bar{\mathbf{x}}_L \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}) \\ 0 \end{bmatrix}$$

9.2 Measurement Model

The measurement model now generates a better reflection of the uncertainties in both subvectors of the state. Now that the landmark position is part of the state vector, the measurement model becomes:

$$\bar{\mathbf{r}}_L^i = \mathbf{f}(\bar{\mathbf{r}}_L^s) = \mathbf{f}(\mathbf{T}_b^s \mathbf{T}_n^b(\bar{\mathbf{x}}_n) \bar{\mathbf{x}}_L)$$

9.3 Measurement Jacobian

The measurement Jacobian is available by **stacking** the Jacobians of the state subvectors:

$$\Delta \bar{\mathbf{r}} \mathbf{I} = \mathbf{H}_I \mathbf{A}^f \mathbf{L}$$

$$A_{il} = \mathbf{T}_b^s \left(\frac{\partial}{\partial \bar{\mathbf{x}}_n} \mathbf{T}_n^b(\bar{\mathbf{x}}_n) \bar{\mathbf{x}}_L \Delta \bar{\mathbf{x}}_n + \mathbf{T}_n^b(\bar{\mathbf{x}}_n) \Delta \bar{\mathbf{x}}_L \right)$$

$$\mathbf{H}_L^T = \mathbf{H}_S^T \mathbf{T}_b^s \begin{bmatrix} \mathbf{a}_i \mathbf{r}_n^b(\bar{\mathbf{x}}_n) \bar{\mathbf{x}}_L & 0 \\ 0 & \mathbf{T}_n^b(\bar{\mathbf{x}}_n) \end{bmatrix}$$

where the usual assumption of uncorrelated measurement error has been made. This formulation will both estimate vehicle position and produce an optimal map. The extension to differential observations is straightforward.

Section 10: Real Time Identification

The filter formulation can be modified to recover important system calibration constants using state vector augmentation techniques. Consider, for example, that the most important calibration constants in a rangefinder perception system are:

- the height c_b^s of the sensor
- the tilt angle Θ_b^s of the sensor
- the range image bias R_{bias}
- the range image scale factor K_{range}

This section is the clearest example of why the tensor notation has been adopted. The measurement models are very involved and it would be hopeless to attempt to formulate them without the compactness afforded by the tensor notation.

10.1 State Vector Augmentation

The state vector can be augmented to include these variables. Formulating in general, let the additional state variables be denoted:

$$\bar{\mathbf{x}}_n = [\mathbf{x} \ y \ z \ \mathbf{V} \ \theta \ \phi \ \psi \ \dot{\beta}]^T \quad \bar{\mathbf{x}}_b = [c_b^s \ \Theta_b^s]^T \quad \bar{\mathbf{x}}_s = [R_{bias} \ K_{range}]^T$$

where the notation suggests parameters of a particular transformation, and introduce them into the system model through the additional differential equations:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{h} \\ \bar{\mathbf{x}}_s \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

Then the system model has been appropriately modified. The only measurements which involve these new states are the landmarks, so they can only be observed with landmark-aided navigation.

10.2 Measurement Model

In the simplest formulation of the problem, consider that landmarks are known in the navigation frame and they generate point features in a range image.

An interest operator is used to identify features in a range image, and for each feature, the image position forms the z_k observation vector. The predictive measurement $h(\bar{\mathbf{x}})$, where the augmented state vector is used, computes the predicted position of the features in each image based on their known positions in the navigation frame:

$$\hat{\mathbf{r}}_L^i = f(\hat{\mathbf{r}}_L^s, \bar{\mathbf{x}}_s) = f(T_b^s(\bar{\mathbf{x}}_b) T_n^b(\bar{\mathbf{x}}_n) \hat{\mathbf{r}}_L^n, \bar{\mathbf{x}}_s)$$

10.3 Measurement Jacobian

The measurement Jacobian is available from the matrix interpretation of the chain rule and the mechanism of **stacking**. This formulation will require modifications to the imaging Jacobian to include differentials with respect to the parameters \bar{x}_s . It also requires the formulation of the Jacobian tensor of the body to sensor transform which is straightforward.

Consider that the expression for the landmark position in the image is a nonlinear function of three vector variables. Then the total derivative is given by the chain rule:

$$\Delta \hat{f} = H_s^j(\bar{x}_s) \Delta \hat{r}_L^s + \frac{\partial f}{\partial \bar{x}_s} \Delta \bar{x}_s$$

$$\Delta f = \frac{\partial}{\partial \bar{x}_b} T_b^s(\bar{x}_b) T_n^b(\bar{x}_n) \bar{r}_L^n + T_b^s(\bar{x}_b) \frac{\partial}{\partial \bar{x}_n} T_n^b(\bar{x}_n) \bar{r}_L^n \Delta \bar{x}_n$$

$$H = \begin{bmatrix} \frac{\partial f}{\partial \bar{x}_s} & 0 \\ 0 & H_s^j(\bar{x}_s) H^j \end{bmatrix}$$

$$H^j = \begin{bmatrix} H_s^j(\bar{x}_s) \frac{\partial}{\partial \bar{x}_b} T_b^s(\bar{x}_b) T_n^b(\bar{x}_n) \bar{r}_L^n & 0 \\ 0 & H_s^j(\bar{x}_s) T_b^s(\bar{x}_b) \frac{\partial}{\partial \bar{x}_n} T_n^b(\bar{x}_n) \bar{r}_L^n \end{bmatrix}$$

This formulation will both estimate vehicle position and calibrate the system kinematic models. The extension to differential observations is straightforward.

Chapter 3: Implementation

This section overviews the implementation of the position estimation system. The position estimation system converts a diverse set of readings from sensors which are:

- redundant
- inaccurate
- asynchronous

into a single consistent estimate of vehicle state of the required temporal resolution and of optimal accuracy in the Kalman filter sense of optimality.

Section 1: Position Estimator

The three main components of the Position Estimator are the System Model, the Sensor Model, and the Kalman Filter. These elements implement two independent update loops which continuously update the vehicle state. The System Model cycles at a nominal rate of 100 Hz. The Sensor Model and Kalman Filter both cycle together at a rate determined by the frequency response and update rate of each sensor involved. The data flow diagram for the Position Estimator is given below. The mathematics of these transforms are documented in the appendices and in this part.

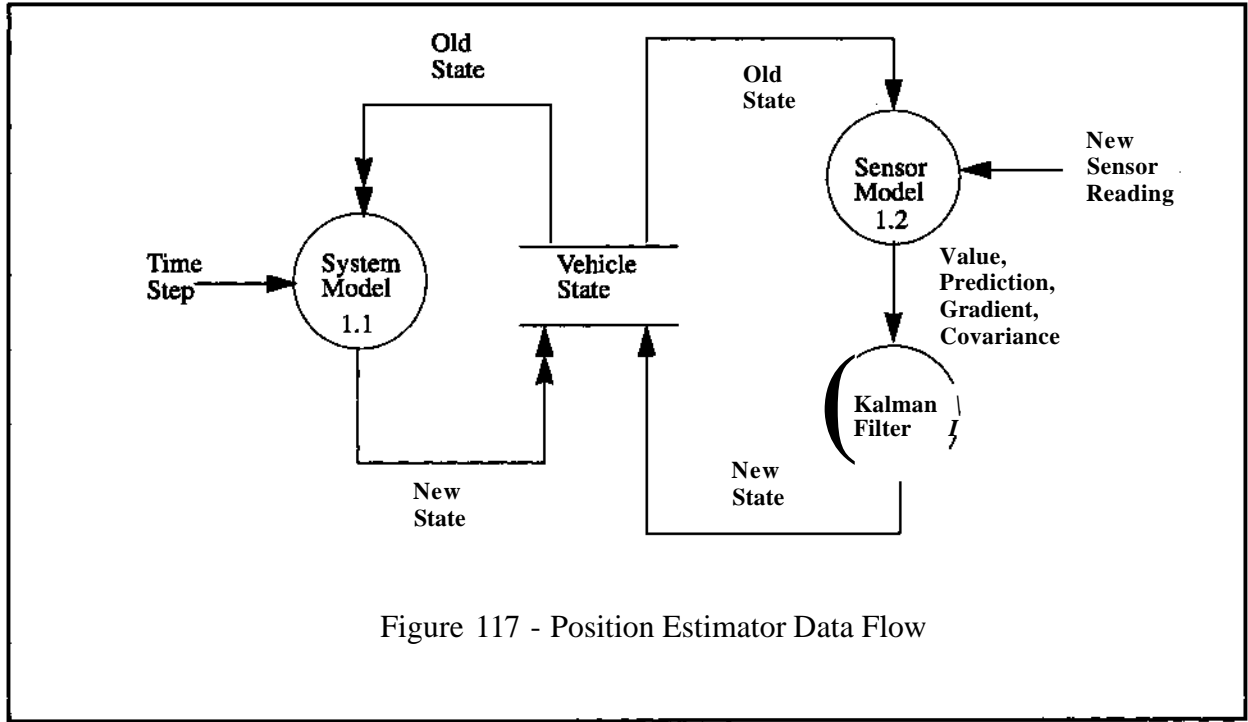


Figure 117 - Position Estimator Data Flow

1.1 System Model

The System Model implements the equations of 3D dead reckoning under a low dynamics assumption. It is an embodiment of the state update equation based on the transition matrix and time step, and the state uncertainty update equation.

1.2 Sensor Model

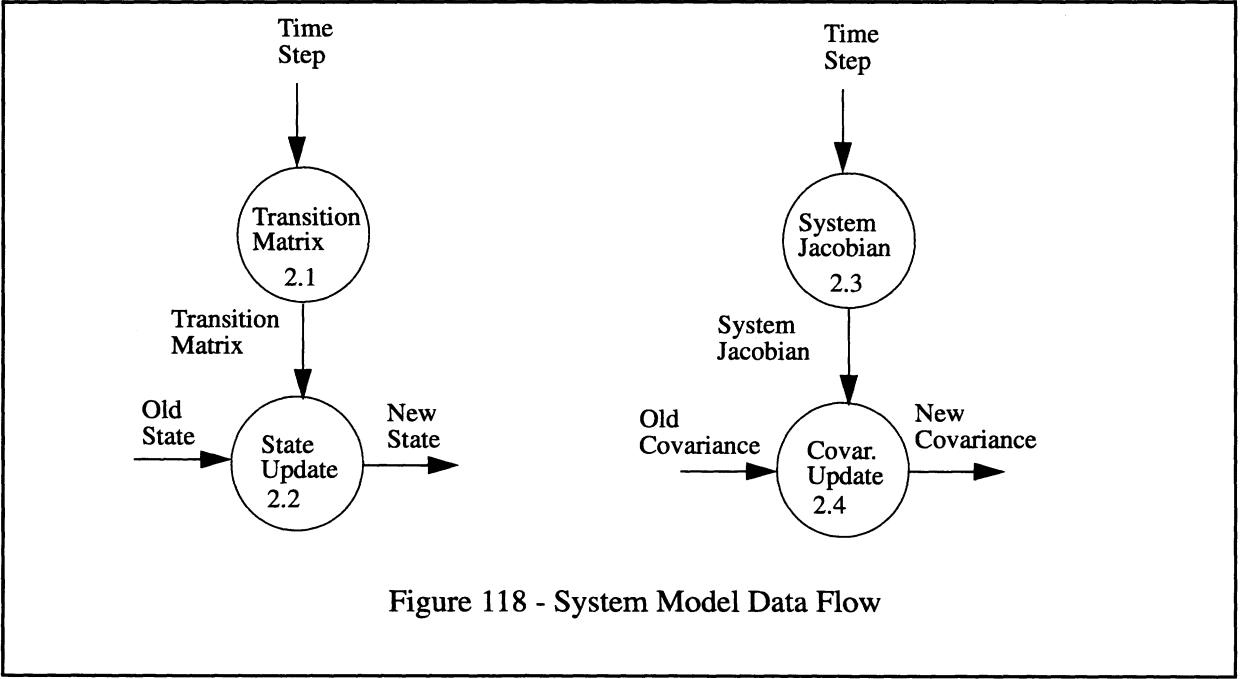
The Sensor Model computes the **predictive measurement** based on the current state estimate and the explicitly coded model. It also computes the **measurement gradient** with respect to the state vector.

1.3 Kalman Filter

The Kalman Filter computes the difference between the predictive measurement and the actual measurement. Based on this **measurement residual**, and the **Kalman gain**, it updates the state estimate.

Section 2: System Model

The System Model is responsible for interpolation of the state estimate between the arrival of measurements. The data flow diagram for this component is given below:



2.1 Transition Matrix

This module computes the elements of the transition matrix based on the explicitly coded 3D dead reckoning model, the current linear and angular velocity, and the time step.

2.2 State Update

This module simply multiplies the state vector by the transition matrix to get the new state.

2.3 System Jacobian

This module computes the system Jacobian of the nonlinear plant based on the first order models discussed in the appendices.

2.4 Covariance Update

This module computes the uncertainty of the new state estimate based on the standard matrix quadratic form for uncertainty update.

Section 3: Sensor Model

The Sensor Model is responsible for computing the nonlinear measurement matrices which correspond to each measurement, their uncertainty, and the predictive measurement based on the current state. The data flow diagram for this component is given below:

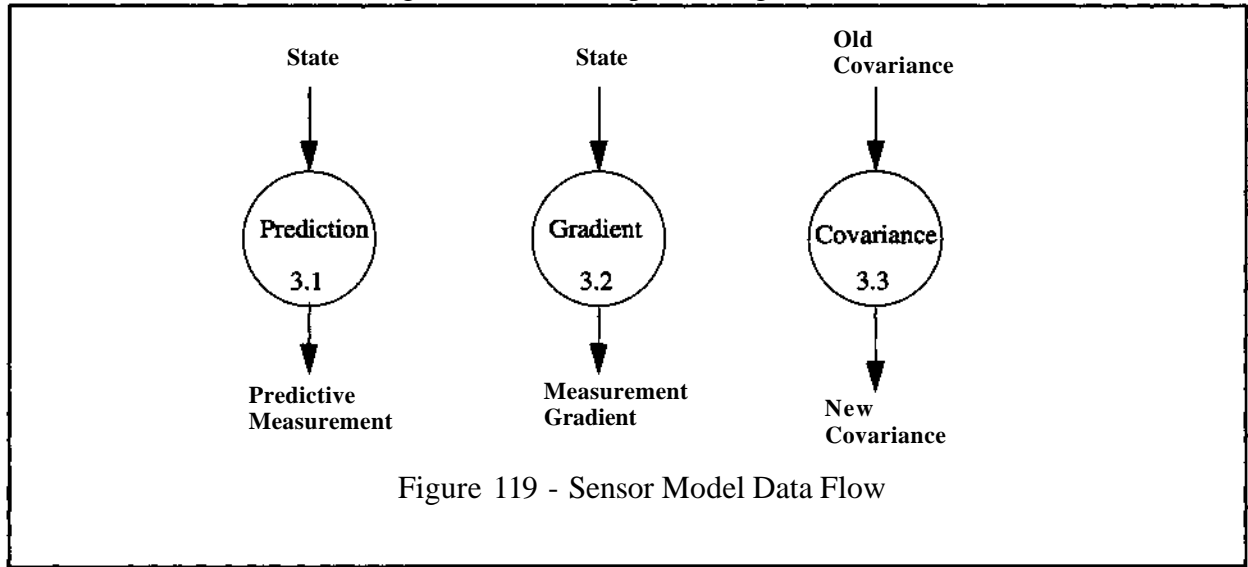


Figure 119 - Sensor Model Data Flow

3.1 Prediction

This module computes the prediction of the measurement based on the current state.

3.2 Gradient

This module computes the gradient of the measurement with respect to the state vector at the current state.

3.3 Covariance

This module updates the state covariance based on the uncertainty in the old state and the uncertainty in the measurement.

Section 4: Kalman Filter

The Kalman Filter is responsible for merging individual measurements into the state estimate. The data flow diagram for this component is given below:

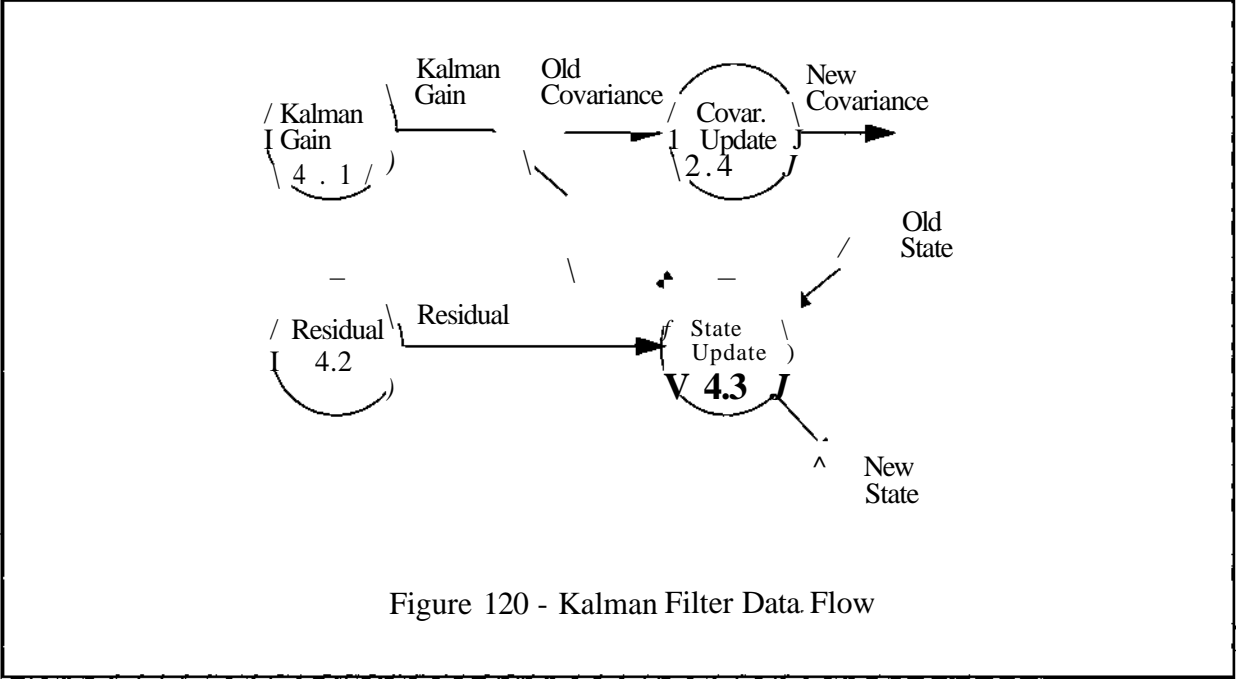


Figure 120 - Kalman Filter Data Flow

4.1 Kalman Gain

This module computes the Kalman gain matrix based on the state uncertainty, the measurement matrix, and the measurement uncertainty.

4.2 Residual

This module computes the measurement residual from the actual and predicted values.

4.3 Covariance Update

This module updates the state covariance based on the uncertainty in the old state and the uncertainty in the measurement.

4.4 State Update

This module implements the actual Kalman state update equation which recomputes state from the measurement residual and the Kalman gain.

Section 5: Calibration and Tuning

The Kalman filter is a mechanism that accounts for random errors in such a way as to generate an optimal estimate. If nonrandom errors exist in the model or the measurements, the filter cannot do much about them unless it is told about them. Further, the estimate is only as good as the uncertainty estimates that are given to the filter. The mathematical ideal of the zero mean white sequence noise source must be stretched significantly in practice. This section discusses these and other matters that are important to the implementation of a working filter.

Once the filter equations are implemented, the remaining practical issues include:

- the calibration of systematic errors
- the characterization of remaining error sources, *as if they were random*, in terms of their first order statistics and correlation models

A real filter *must be tuned* very precisely for “optimal performance”, and it may not work at all if this is not done properly. This section outlines some techniques used to achieve these goals.

The significance of calibration to Kalman filtering is that zero mean noise sources are assumed, and any bias in the distribution will result in suboptimality. In practical terms, most sensors in use are nominally linear, and the measurement model is a straight line. For such a model, the calibration problem becomes one of estimating **bias** and **scale factor**, which are just the y intercept and the slope of the measurement model. Any remaining misfit in the model can be regarded, in practice, as uncertainty to be given to the Kalman filter.

On occasion, the model parameters are known to have stability problems (i.e. to vary with unmodelled parameters such as time) and adaptive techniques can be used to estimate them in real time. Constant bias occurs in several sensors in the suite. The compass for instance may not be calibrated for the local region. Gyros may exhibit slight biases which may or may not be constant.

It can be shown that even very small systematic error in the attitude indications can ultimately cause enormous error in the filter output position. This is because these errors act forever. It is very important to remove systematic attitude error through calibration techniques.

When **bias stability** is known or suspected to be a problem, the following techniques are used.

5.1 Relative Navigation

Relative navigation avoids the bias issue completely because it cancels out in the mathematics of the application. For example, bias in the compass heading can be ignored if no measurements of landmarks are referenced to a geocentric frame of reference. This is possible since high-pass gyros will not require knowledge of the transform to the geocentric earth frame, and thus, vehicle azimuth can be treated as a relative quantity. The east and north directions are arbitrary. In inertial navigation systems, this idea is called **wander azimuth** but it is done for different reasons. However, if any device such as GPS is used which reports position in a geographic frame, the details of the alignment to that frame will have to be addressed. Differential GPS is another form of relative navigation, as is traditional encoder dead reckoning when the start point is taken at zero.

5.2 Zups

Another type of bias removal is the zero velocity update, or **zup**. For terrestrial vehicles, sustained speeds below a few inches per second are uncommon enough to be considered to result from sensor biases and not vehicle motion. The zup can be engaged after sufficiently long periods of very slow velocities are observed coming from the sensors. Attitude zups are possible too, for if the velocity is zero, it is likely that the attitude is not changing either.

Zups are useful and sometimes necessary for other reasons. Specifically, when the Q matrix is not zero, the P matrix grows in magnitude with each iteration of the system model. This may be unrealistic and even harmful to the operation of the filter. Once the filter enters zup mode, this can be checked until motion begins again. Zups must be triggered by the measurements and not the state vector. Otherwise, there is no simple way to exit zup mode.

5.3 State Vector Augmentation

Another mechanism for bias removal, state vector augmentation, was discussed in an earlier section and is elaborated in the appendices. Analogous techniques are available for adaptive measurement of scale factor.

Section 6: Initialization and Reconfiguration

The filter equations make no distinction between the first cycle and any other, so any choice of the initial state vector and its uncertainty can be made *provided it is consistent*. Also, sensors can be added and removed dynamically in some cases, provided the filter observability criterion is not compromised.

6.1 Initialization

Two general cases are possible and they can be mixed across the elements of the state vector and covariance matrix.

6.1.1 Relative Navigation

Set the initial position to zero and its uncertainty to zero. Provided the appropriate elements of Q_k are nonzero, the state estimate will evolve normally in time as if the initial position was known perfectly. This technique applies mostly to linear position states. Since it requires that landmarks be known with respect to the initial position, it may be difficult to use with any landmark observations unless they are observed differentially.

6.1.2 Absolute Navigation

Set the initial state value to zero and its uncertainty to a large number. The first measurement of that state will naturally replace the initial value when it is incorporated into the estimate. Note however, that the filter will run open loop with respect to that state until a measurement is received. Depending on the time interval involved, this may or may not be a problem.

6.2 Reconfiguration

Certain issues arise during the development of a filtering algorithm for which a small number of simple techniques are useful.

6.2.1 Removing Sensors

When for some reason, a sensor becomes unavailable, it may be necessary to change the operation of the filter. The simplest case is when the filter is implemented for asynchronous measurements and the appropriate variables can be set to indicate that the measurement is no longer available. If this is not possible, the uncertainty of the measurement can be set to a large number in the R_k matrix. If the removed sensor is the last sensor which measured a particular state, then that state may need to be removed as well because cross-coupling in the matrices will generate fictitious values for the state which will then begin to corrupt other states. In the event that observability is compromised, the whole filter is not viable without the lost sensor.

6.2.2 Removing States

In order to remove a state, the numerical operation of the filter must be as if the filter was originally formulated without it. It is not enough to set the state uncertainty to a large number; the value of the state must be set to zero every cycle in order to damp the errors introduced by every iteration.

Section 7: Bandwidth and Efficiency

7.1 Asynchronous Implementation

One matter to be addressed in implementation is the asynchronous nature of the measurements. For example, Doppler readings may not be available as frequently as encoder readings. GPS measurements cannot be generated faster than 2 Hz whereas inertial systems can be 100 times faster than this. The state equations are run at about 100 Hz while measurements are incorporated at whatever rate they are generated by the sensors. The basic algorithm is given below:

```
State_Update() /* entered every cycle */
{
    update state estimate for a time step of dt
    via the transition matrix(dt);
    if( Doppler measurement available)
        run Kalman() on Doppler;
    if( Encoder measurement available)
        run Kalman() on encoder;
    if( AHRS measurement available)
        run Kalman() on AHRS;
    if( Compass measurement available)
        run Kalman() on compass;
    if( Steering measurement available)
        run Kalman() on steering;
}

Kalman()
{
}
```

7.2 Efficiency

In implementing the filter equations, it is typical to find that most of the time is spent computing the uncertainty matrices, and the Kalman gain, and in inverting and multiplying matrices. Some simple steps can be taken to reduce this problem. The following two steps make it possible to run the filter at 100 Hz.

7.2.1 Kalman Gain

The matrix Kalman gain equation for an EKF is:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$$

Let a single measurement arrive for integration with the state estimate. Then, the R matrix is a scalar:

$$R = [r]$$

Let the measurement project onto a single state whose index is s with a coefficient of unity. Then the H matrix is a row vector with a single unit element in the s 'th position:

$$H = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$$

The expression $P_k^{-T} H_k^T$ is the s 'th column of P_k^{-T} and the expression $H_k P_k^{-T} H_k^T$ is the (s,s) element of P_k^{-T} . Define:

$$p = P_{ss}^{-T}$$

Finally, the Kalman gain is a column vector equal to a constant times the s 'th column of P_v^{-T} :

$$K = \left(\frac{1}{p+r}\right)P_{is}^{-T} \quad \forall i$$

7.2.2 Uncertainty Propagation

The matrix uncertainty propagation equation for an EKF is:

$$P = [I - (KH)]P$$

This can be computed many times faster as:

$$P = P - K(HP)$$

and this rewrite is valid regardless of the form of the measurement matrix H .

Further simplification is possible in the case of a single scalar measurement. Let a single measurement arrive for integration with the state estimate. Again, let the measurement project onto a single state whose index is s with a coefficient of unity. Then the H matrix is a row vector with a single unit element in the s 'th position:

$$H = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$$

The expression HP is then the s 'th row of P_k^{-T} . Reusing the last result, the expression KHP is simply a constant times the outer product of the s 'th column and the s 'th row of P_v^{-T} :

$$(KHP)_{ij} = \left(\frac{1}{p+r}\right)P_{is}^{-T}P_{sj}^{-T} \quad \forall i \forall j$$

Section 8: Outlier Rejection

Although a Kalman filter can handle random measurement noise, a single outlier can cause a large erroneous change in the state estimate unless it is rejected before it is used in the state update equation. This section describes some techniques used to achieve outlier rejection.

8.1 Angular Addition

Notice that the Kalman state update equation and the system model constitute two places where angles are added. After each iteration of these equations, and during the formation of the measurement residual, the angular states *must* be forced into some canonical sector of the unit circle such as $-n$ to 71 . If this is not done, the measurement residuals will begin, apparently randomly, to overestimate the mismatch by a full revolution and the state vector will rotate one revolution over a few cycles. This is usually quite a disaster if the filter output forms the sensor for a control loop.

8.2 Reasonableness Checks

At other times, a sensor that is not wired correctly, not turned on, poorly calibrated or in any other way not functioning correctly can cause a computational tug of war in the filter that may lead to divergence. One good generic reasonableness check is to compare the Kalman residual for each measurement with its own nominal uncertainty. If the residual is much larger than the uncertainty, then something is probably wrong and the measurement can be discarded. This is a good way of eliminating spikes and outliers. If a sensor fails to meet this criterion consistently, then it can be disabled forever and an error message issued.

8.3 Legitimate State Discontinuities

When high quality measurements, like fixes, arrive infrequently, it is possible for the state estimate to undergo a rapid change in a short period of time. This can wreak havoc in any closed loop system since it appears as a large deviation between the reference input and the feedback signal.

One technique for dealing with this is presented in the appendices in the form of the simple linearized Kalman filter which computes the state error vector instead of the state. Another technique is to low-pass filter the state output but this has the disadvantage of delaying control response to legitimate high dynamic feedback if the entire state vector is filtered.

Since discontinuities will only arise in practice in the position states, it is appropriate to slowly filter the x , y , and z components of the state discontinuity *only* into the state estimate used as position feedback by control algorithms. An appropriate choice of time constant in the filter will prevent unusual behavior. This approach becomes increasingly less viable as the frequency of fix information is reduced.

Chapter 4: Results

Section 1: AHRS Dead Reckoning

The 3D dead reckoning AHRS filter has been implemented and tested in the field. Tuning of the state uncertainties was required to optimize the tradeoff between data smoothing and sensor tracking. A zup mode was included in order to check undesirable growth of the state uncertainty when the vehicle was stopped. It was observed that heading uncertainty was treated much more accurately in EKF mode. Specifically, uncertainty grows primarily in the direction of travel in LKF mode, whereas when the coupling terms of the EKF are introduced, uncertainty grows transverse to it. The latter is more realistic in situations when the heading indication is poor. Indeed, the overwhelming component of position error was due to poor attitude indications.

The sensors used were a steering wheel potentiometer for "yawrate"¹⁶ measurement, transmission encoder and redundant Doppler radar for measurement of velocity, and an AHRS for three-axis attitude. Time stamps were used to compute encoder velocity precisely. Significant systematic errors were easily identified from the signal plots which were not resolved before this writing.

One of many test runs will be used to illustrate performance. In this run, winding mountainous city streets were driven. The total excursion was about 4 Km in the horizontal plane and 200 meters vertically. Aggressive braking and accelerating maneuvers were common. On many occasions, the vehicle stopped for traffic lights, and the filter idled in zup mode until motion resumed. The following comments apply to Figure 121.

The qualitatively correct growth of uncertainty is illustrated because the uncertainty ellipses touch the path when it was driven in the other direction. Point repeatability less than 1% of the travelled distance was normally achieved. Relative and absolute accuracy were not quantified.

One of the advantages of the 3D formulation is the availability of the z coordinate. Zups appear as flat regions because the abscissa is time. Notice that the start and end do not agree. This is likely due to the high vehicle accelerations corrupting the inclinometers used. In fact, the plotted pitch signal shows significant periods when the output during the downhill leg was positive. However the relative accuracy for a few seconds is very good. This suggests that the z output should be good enough to avoid map matching problems in a mapping perception system.

The position output in the plane is illustrated. Notice that the return path from the turn point could be rotated through a small angle at the turn point and the graphs would overlap almost perfectly. A residual systematic error in heading is responsible for this.

The yaw output indicates the success of the reasonableness checks in angle addition in the filter. Both the yaw sensor and the filter state are plotted but they cannot be distinguished beyond the first few cycles.

16. The measurement of body vertical angular velocity is not really yawrate because yawrate is measured about global z, not body z.

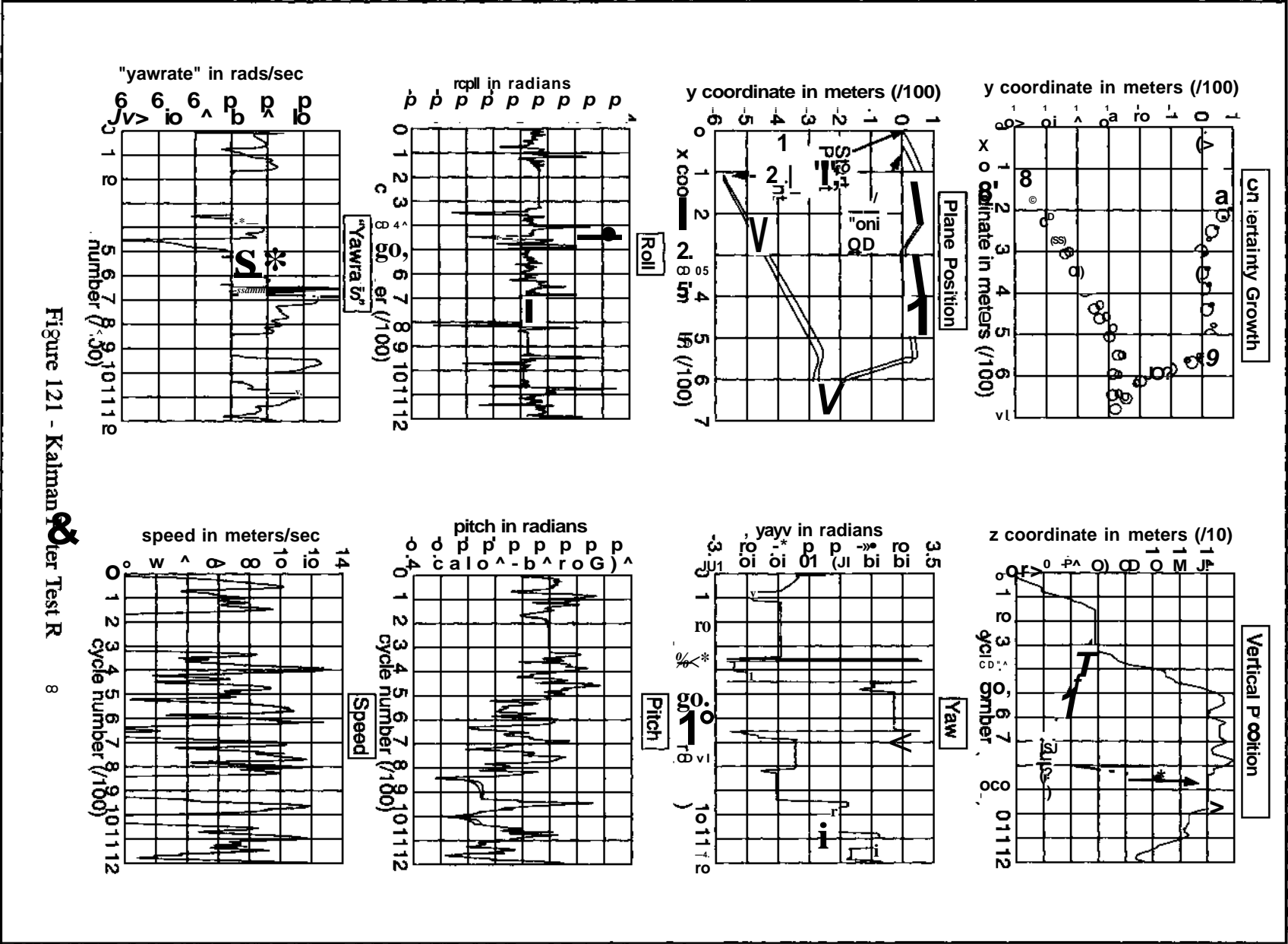


Figure 121 - Kalman Filter Test Results

The roll state and sensor reading are plotted. Again, the state simply tracks the single sensor. Corruption via vehicle acceleration is likely because the signal is not symmetric.

The pitch signal shows how vehicle acceleration during braking maneuvers causes the state to lag the sensor output. Curiously, the output is again more positive than it should be. This is a calibration issue which cannot be accounted for in the filter itself, at present.

The yawrate state is available from the steering wheel encoder. A nonlinearity in the steering sensor causes a bias near zero which causes the graph to be predominantly positive, when, in fact, it should be roughly skew symmetric about a vertical line.

The speed output shows how both redundant sensors are used to determine the state. The Doppler and encoder readings agree so well that the averaging of the filter is hard to see. On magnification, it is clear that the filter is using both and determining an estimate which is a weighted average when they disagree slightly.

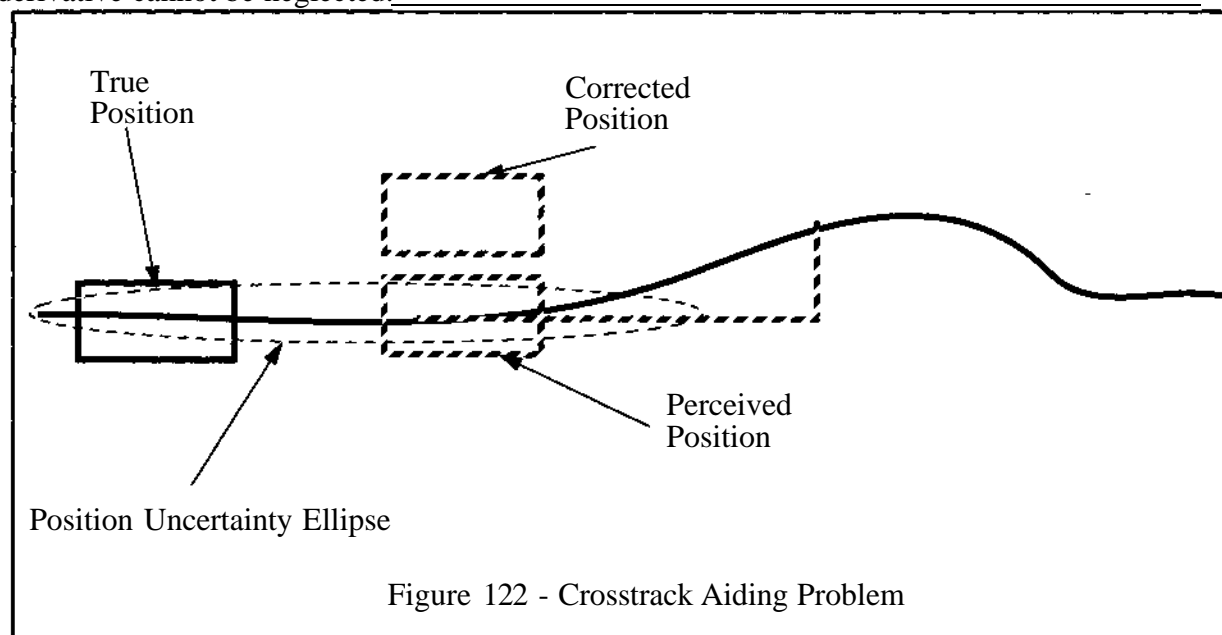
Section 2: Crosstrack Aiding

The ALVINN system [69] was used to generate a crosstrack observable for testing purposes and the overall effect on the position estimate was a negative one. While, at times, the system seemed to work, there were many times when large jumps in the state estimate occurred that no amount of tuning could remove. Oftentimes, the state jumps occurred near turns or where high curvature path points occurred. These were the very points where the crosstrack observable was potentially the most useful.

The conclusion that can be drawn is that while a crosstrack observable may be useful in theory, it had two severe problems in practice on the test vehicle. First, because crosstrack is a ID measurement, it can only reduce uncertainty in the crosstrack direction. On long straight path segments, in the absence of other fix information, uncertainty is unbounded in the alongtrack direction.

Second, when the alongtrack uncertainty is large, *the fundamental linearity assumption of the EKF is violated*. This is illustrated in the following figure. Suppose the vehicle position estimate is somewhat ahead of the true vehicle position. Then, the predictive crosstrack measurement will have a substantial positive value. However, the vehicle is really several lengths behind the estimate, so the true observed crosstrack is approximately zero.

On this basis, the filter will compute the position error which would have caused the deviation *based on the measurement Jacobian at the current state estimate*. This will cause the updated position to be to the left of the current estimate as shown. Fundamentally, the problem is that the measurement Jacobian changes direction rapidly over the region of state uncertainty, so the second derivative cannot be neglected.



This problem with crosstrack aiding is really a universal problem with using fix information when the dead reckoning error is already substantial.

Chapter 5: Summary and Conclusions

This part described the state space Kalman Filter which forms the core of the position estimation system. These algorithms form the basis of RANGER'S Position Estimator object. The most important decisions in the filter design problem are the **modelling** decisions which outline the states, the measurements, the noise models, coordinate systems, linear approximations, etc. Thus, the purpose of this part was to outline a set of plausible modelling decisions for the navigation problem for autonomous vehicles which have led to a working Kalman filter.

Section 1: Concept

This section outlined an approach to autonomous vehicle position estimation based on the state space Kalman filter. This particular filter formulation is quite general. The formulation has wide-ranging uses.

LI State Space Kalman Filter

The fundamentals of the state space Kalman filter are described in the appendices for the benefit of readers unfamiliar with the basic concept. The system model estimates vehicle state based solely on a measurement of time because it expresses the differential equations of motion and the state vector provides all required initial conditions.

1.1.1 Positioning Requirements

The basic requirements on position estimation for the purposes of implementing autonomous local navigation are as follows:

- Position updates bandwidth must equal or exceed 100 Hz.
- Relative position accuracy must equal or exceed 1% of distance travelled.
- Absolute attitude accuracy must equal or exceed 1°.

1.1.2 Design Decisions and Assumptions

For the requirements mentioned above, an AHRS dead reckoning Kalman filter promises to meet all requirements. In order to achieve the 100 Hz update rate, the asynchronous model permits many measurement models to be expressed in closed form as scalar equations, and this reduces the matrix computations to a minimum. A few key assumptions permit the filter to perform as required:

- The **low dynamics assumption** asserts that velocity is nearly constant between measurements of velocity.
- The **Taylor remainder theorem** is used to estimate the system model uncertainties necessary to implement the filter.
- The **principle motion assumption** asserts that linear velocity occurs only along the forward body axis and angular velocity occurs only along the body vertical axis.

1.1.3 AHRS Dead Reckoning

AHRS is an acronym for Attitude and Heading Reference System which is used in the navigation industry to mean a suite of components used for measuring the attitude of a vehicle and sold as an integrated package. A filter for a mobile robot position estimation system is proposed which utilizes an AHRS as the attitude indicator.

The AHRS can be considered to be a sensor package consisting of integrating gyroscopes, a magnetic compass, and inclinometers. The position estimation system integrates these attitude indications with wheel or transmission encoder(s), Doppler ground speed radar, and a steering wheel encoder.

In order to support wide excursion missions, and because a nominal trajectory is not always available, an extended Kalman filter was formulated.

1.1.4 Advantages

Some of the advantages of applying the Kalman filter to the position estimation problem are:

- Redundant odometric and attitude measurements can be integrated together.
- Dead reckoning and triangulation measurements can be integrated together.
- Frequency response of sensors can be modelled.
- Inertial force compensation can be performed.
- Nonlinear error propagation is more accurate.
- Noise immunity is a characteristic of the position estimate.

Section 2: Design

2A AHRS Dead Reckoning System Model

This section begins the exposition of the modelling decisions. It was shown that the formulation of the system model in the navigation frame leads to an observability problem because a measurement of forward velocity cannot be used to estimate the entire 3D vehicle velocity vector.

This problem can be eliminated by reformulating the filter system model in the body frame. The state variables are reduced in number and the system model is reformulated to explicitly assume that:

- the vehicle translates only along the body y axis
- the vehicle rotates only around the body z axis

This will be called the **principal motion assumption**, and it amounts to replacing six legitimate states with two special combinations of themselves. The principal motion assumption solves the observability problem.

Having made this assumption, the body frame model was elaborated in more detail. The state vector includes the position and attitude of the body and the components of linear and angular velocity mentioned above. This gives an eighth-order Kalman filter. The system model is elaborated as the nonlinear equations of 3D dead reckoning. The model nonlinearity is a consequence of the body frame model. Therefore the must be linearized according to the rules for an extended Kalman filter. The system Jacobian and transition matrix were elaborated for this EKE

2.2 AHRS Dead Reckoning Measurement Model

Measurement models were provided for all of the following sensors:

- Transmission encoder
- Doppler groundspeed radar
- Fluxgate compass
- AHRS
- Steering wheel encoder
- Wheel encoders

2.3 AHRS Dead Reckoning Uncertainty Model

An **uncorrelated measurement error assumption** is adopted to render the associated covariance matrices diagonal.

2.3.1 State Uncertainty

The state uncertainty model represents the disturbances which excite the linear system. In the absence of any other information, a plausible approach is to estimate error as the Taylor remainder¹⁷ in the dead reckoning equations, because, after all, dead reckoning is a truncated Taylor series in time.

2.3.2 Measurement Uncertainty

The measurement uncertainties are far more critical to the filter operation, because, after all, the whole system is considered to fail if sensors are lost for only a few seconds and filter optimality is not an issue. For sensors models, some of the error sources to be estimated include:

- bias and scale instability
- neglected sensor dynamics
- noise, vibration, backlash, EMI, and compliance, etc.

This section modelled all sources of error as if they were random. Models were elaborated for all of the sensors mentioned above.

1A Aided AHRS Dead Reckoning

The pure dead reckoning filter of the previous section is unlikely to achieve an accuracy which exceeds a few percent of the distance travelled. This is because of the essential integration of errors in the process of dead reckoning. The Kalman filter is an ideal formalism for integration of dead reckoning and position fixes because fixes are simply additional measurements which can be folded into the equations in like manner to the DR measurements. Some of the fix-generating mechanisms discussed were survey points, path features, GPS, and roadfollower crosstrack observations.

17. Clearly, this approach is equivalent to associating the magnitude of the truncated term with the spectral amplitude of a white sequence and integrating over the time step to get the variance.

2.5 Terrain-Aided AHRS Dead Reckoning

Vehicle perception sensors provide many mechanisms for generating a fix. Landmark recognition can be folded into the filter, but the measurement model is nontrivial because it must account for the imaging process. This section and the following three presented a general model of the perception measurement process which is applicable to many different scenarios.

2.6 Simultaneous Mapping and Position Estimation

With an augmented state vector, it is possible to "distort" an evolving map in order to produce a better estimate of both the vehicle position and the environmental map. This is achieved with state vector augmentation.

2.7 Real Time Identification

The filter formulation can be modified to recover important system calibration constants using state vector augmentation techniques. A model was proposed for the recovery of calibration constants in a rangefinder perception system.

Section 3: Implementation

This section overviews the implementation of the position estimation system. The position estimation system converts a diverse set of readings from sensors which are:

- redundant
- inaccurate
- asynchronous

into a single consistent estimate of vehicle state of the required temporal resolution and of optimal accuracy in the Kalman filter sense of optimality.

3.1 Position Estimator

The three main components of the Position Estimator are the System Model, the Sensor Model, and the Kalman Filter. These elements implement two independent update loops which continuously update the vehicle state. The System Model cycles at a nominal rate of 100 Hz. The Sensor Model and Kalman Filter both cycle together at a rate determined by the frequency response and update rate of each sensor involved.

3.2 System Model

The System Model implements the equations of 3D dead reckoning under a low dynamics assumption. It is an embodiment of the state update equation based on the transition matrix and time step, and the state uncertainty update equation.

3.3 Sensor Model

The Sensor Model computes the **predictive measurement** based on the current state estimate and the explicitly coded model. It also computes the **measurement gradient** with respect to the state vector.

3.4 Kalman Filter

The Kalman Filter computes the difference between the predictive measurement and the actual measurement. Based on this **measurement residual**, and the **Kalman gain**, it updates the state estimate.

3.5 Calibration and Tuning

The Kalman filter is a mechanism that accounts for random errors in such a way as to generate an optimal estimate. If nonrandom errors exist in the model or the measurements, the filter cannot do much about them unless it is told about them. Further, the estimate is only as good as the uncertainty estimates that are given to the filter. This section discussed these and other matters that are important to the implementation of a working filter.

Once the filter equations are implemented, the remaining practical issues include:

- the calibration of systematic errors
- the characterization of remaining error sources, *as if they were random*, in terms of their first order statistics and correlation models

The significance of calibration to Kalman filtering is that zero mean noise sources are assumed, and any bias in the distribution will result in suboptimality.

3.6 Initialization and Reconfiguration

The filter equations make no distinction between the first cycle and any other, so any choice of the initial state vector and its uncertainty can be made *provided it is consistent*. Also, sensors can be added and removed dynamically in some cases, provided the filter observability criterion is not compromised.

3.6.1 Initialization

Two general cases are possible and they can be mixed across the elements of the state vector and covariance matrix. In relative navigation, set the initial position to zero and its uncertainty to zero. In absolute navigation, set the initial state value to zero and its uncertainty to a large number.

3.6.2 Reconfiguration

Certain issues arise during the development of a filtering algorithm for which a small number of simple techniques are useful. The problems associated with removing sensors and states were discussed.

3.7 Bandwidth and Efficiency

A few techniques were discussed which lead to significant improvement in the bandwidth of the filter. Efficient scalar equations were given for computation of the Kalman gain and the propagation of uncertainty.

3.8 Outlier Rejection

A single outlier can cause a large erroneous change in the state estimate unless it is rejected before it is used in the state update equation. This section described some techniques used to achieve outlier rejection including proper angular addition, reasonableness checks, and filtering legitimate state discontinuities.

Section 4: Results

4.1 AHRS Dead Reckoning

The 3D dead reckoning AHRS filter has been implemented and tested in the field. The sensors used were a steering wheel potentiometer for "yawrate" measurement, transmission encoder and redundant Doppler radar for measurement of velocity, and an AHRS for three-axis attitude.

One of many test runs was used to illustrate performance. In this run, winding mountainous city streets were driven. The total excursion was about 4 Km in the horizontal plane and 200 meters vertically. Aggressive braking and accelerating maneuvers were common. On many occasions, the vehicle stopped for traffic lights, and the filter idled in zup mode until motion resumed.

4.2 Crosstrack Aiding

The ALVINN system [69] was used to generate a crosstrack observable for testing purposes and the overall effect on the position estimate was a negative one. While, at times, the system seemed to work, there were many times when large jumps in the state estimate occurred that no amount of tuning could remove.

When the alongtrack uncertainty is large, *the fundamental linearity assumption of the EKF is violated* and this seems to be the reason for the failure of the crosstrack observable to amp position errors. This problem with crosstrack aiding is really a universal problem with using fix information when the dead reckoning error is already substantial.

PART VII: Summary and Conclusions

This part will summarize the thesis in terms of its basic arguments, results, and conclusions.

Section 1: Problem Statement

The problem addressed by this thesis is **intelligent autonomous mobility**. That is, the problem of moving a robot vehicle around in an environment while remaining safe. This problem classification includes the problems of obstacle avoidance, perception, position estimation, coordinated actuator control, and goal-seeking but does not include the problem of strategic motion planning.

Section 2: RANGER Navigator

The navigation system called RANGER can be classified as an *intelligent, predictive controller* for autonomous vehicles. It is intelligent because it uses range images that are generated from either a laser rangefinder or a stereo triangulation rangefinder in order to perceive the immediate environment of the vehicle. It is predictive because it reasons based on its knowledge of its own capability to react to hazards in real time. The system closes the overall perceive-think-act loop for a robot vehicle at relatively high update bandwidth and incorporates both somatic and environmental feedback.

The goal of the project has been to increase speed and enhance the reliability of robotic vehicles in rugged outdoor settings. The navigator has navigated over distances of 15 autonomous kilometers, moving continuously, and has at times reached speeds of 15 km/hr. The system has been used successfully on a converted U.S. Army jeep called the NAVLAB II and on a specialized Lunar Rover vehicle that may, one day, explore the moon.

2.1 Operational Modes

The system can autonomously seek a predefined goal or it can be configured to supervise remote or in-situ human drivers and keep them out of trouble.

2.2 Goal-Seeking

The system can follow a predefined path while avoiding any dangerous hazards along the way or it can seek a sequence of positions or a particular compass heading. In survival mode, seeking no particular goal, it will follow the natural contours of the surrounding terrain.

2.3 World Model

A computerized terrain map data structure is maintained which models the geometry of the environment. It is an array of elevations that represents the world as a 2-1/2 D surface where the vertical direction is aligned with the gravity vector. This representation, combined with a model of vehicle geometry, permits a robust assessment of vehicle safety.

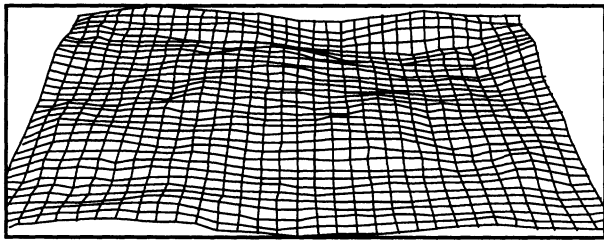
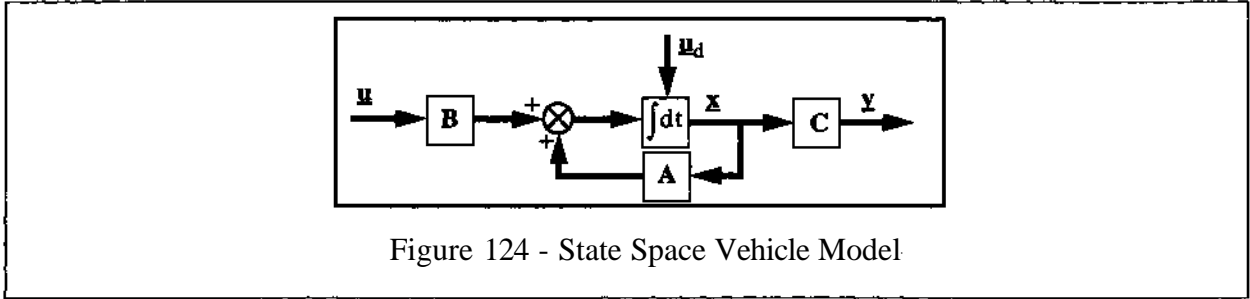


Figure 123 - Terrain Map

1A Vehicle Model

The system is a solution to the difficulties of autonomous control of land vehicles based on a tightly-coupled, adaptive feedforward control loop. It incorporates measurements of both the state of the vehicle and the state of the environment and maintains high fidelity models of both that are updated at very high rates.

At sufficiently high speeds, it becomes necessary to explicitly account for the difference between the ideal response of the vehicle to its commands and its actual response. The vehicle is modelled as a dynamic system in the sense of modern control theory. Although the system uses a nonlinear model, the linear system model expressed in the following generic block diagram provides a sense of the important signals and transformations involved.



FIFO queues and time tags are used to model the delays associated with physical i/o and to register contemporary events in time. The command vector \underline{u} includes the steering, brake, and throttle commands. The disturbances \underline{u}^{\wedge} model the terrain contact constraint. The state vector \underline{x} includes the 3D position and 3 axis orientation of the vehicle body as well as its linear and angular velocity. The system dynamics matrix A propagates the state of the vehicle forward in time. The output vector \underline{y} is a time continuous expression of predicted hazards where each element of the vector is a different hazard.

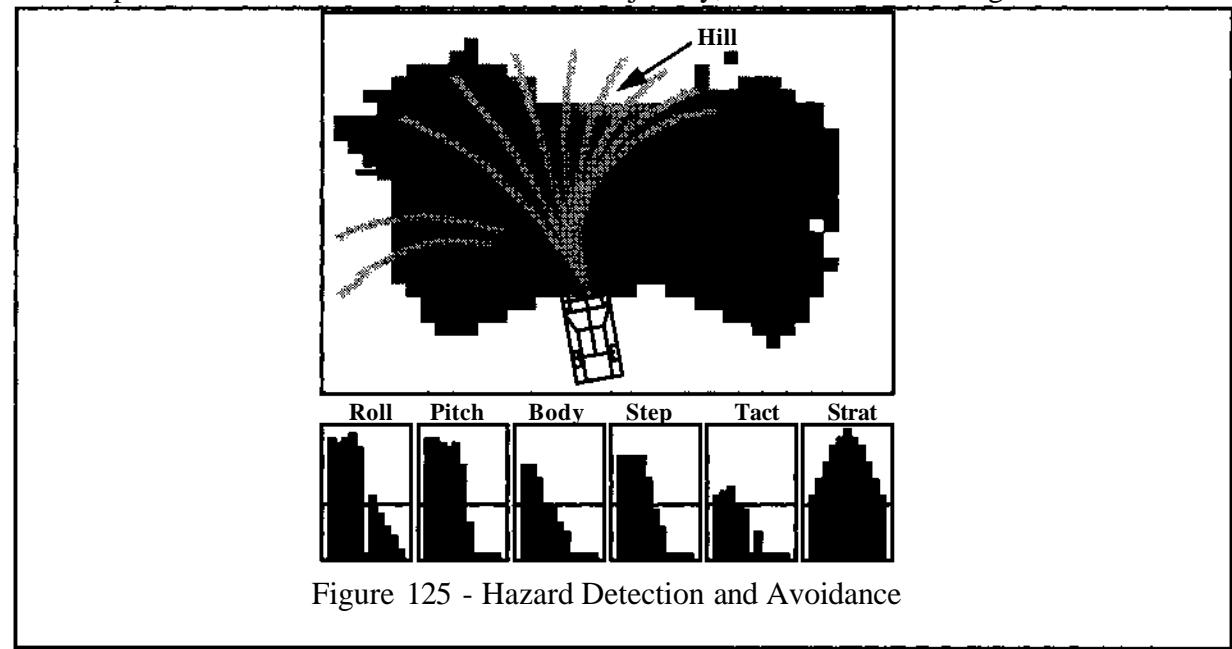
2.5 Hazard Assessment

Hazards include regions of unknown terrain, hills that would cause a tip-over, holes and cliffs that would cause a fall, and small obstacles that would collide with the vehicle wheels or body.

The process of predicting hazardous conditions involves the numerical solution of the equations of motion while enforcing the constraint that the vehicle remain in contact with the terrain. This process is a feedforward process where the current vehicle state furnishes the initial conditions for numerical integration. The feedforward approach to hazard assessment imparts high-speed stability to both goal-seeking and hazard avoidance behaviors.

System components above the state space model in the software hierarchy translate the hazard signals $\underline{x}(t)$ into a vote vector. This is accomplished by integrating out the time dimension to generate a vote for each steering direction based on a normalization of the worst case of all of the considered hazards.

In the figure, the system issues a left turn command to avoid the hill to its right. The histograms below represent the votes for each candidate trajectory, for each hazard. Higher values indicate



safer trajectories. The hazards are excessive roll, excessive pitch, collision with the undercarriage, and collision with the wheels. The tactical vote is the overall vote of hazard avoidance. It wants to turn left. The strategic vote is the goal-seeking vote. Here it votes for straight ahead.

2.6 Arbitration

At times, goal-seeking may cause collision with obstacles because, for example, the goal may be behind an obstacle. The system incorporates an arbiter which permits obstacle avoidance and goal-seeking to coexist and to simultaneously influence the behavior of the host vehicle. The arbiter can also integrate the commands of a human driver with the autonomous system.

2.7 Sensors

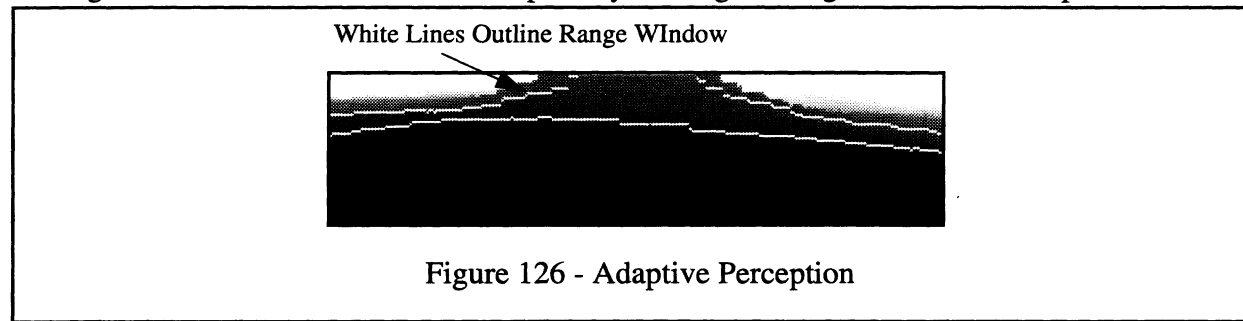
The navigator accommodates both laser rangefinder and stereo perception systems and it incorporates its own integrated stereo correlation algorithm. In either case, the design achieves significant increases in vehicle speeds without sacrificing either safety or robustness.

2.8 Adaptive Perception

Perception has long been acknowledged as the bottleneck in autonomous vehicle research. Yet, a moving vehicle generates images which contain much redundant information. Removal of this redundancy is the key to fast moving robot vehicles.

A new range image perception algorithm has been developed for the navigator. It selectively extracts a very small portion of each range image in order to reduce the perceptual throughput to a bare minimum. In this way, vehicle speed is less limited by the computer speed.

The algorithm searches each image for a band of geometry that is between two range extremes, called the range window as shown in the figure. Only the data between the white lines is processed. The algorithm also accounts for vehicle speed by moving the range window out as speeds increase.



This approach also stabilizes the sensor in software because the search for the data of interest adapts automatically to both the shape of the terrain and the attitude of the vehicle. It is up to 6000 times faster than traditional approaches and it achieves the throughput necessary for 20 m.p.h. motion on an ordinary computer workstation.

2.9 Position Estimation

The navigator incorporates a Kalman Filter algorithm that merges the indications of all of the navigation sensors into a single consistent estimate of the vehicle position, attitude, and velocity. Any number of sensors in any combination can be accommodated including, wheel or transmission encoders, compasses, gyroscopes, accelerometers, doppler radar, inclinometers, terrain aids such as landmarks and beacons, and inertial and satellite navigation systems.

2.10 Implementation

While the real-time core of the system can be expressed in about 1000 lines of C, the navigator includes a simulation and development environment incorporating a data logger and simulators for natural terrain, vehicles, sensors, and pan/tilt mechanisms. Real-time animated graphics provide feedback to the human supervisor. A custom C language interpreter is used to configure and control the system at run-time.

2.11 Architecture

Implemented in C language, the system is composed at the highest level of four objects. The Map Manager integrates all environmental sensor images into a single consistent world model. The Controller implements hazard detection and avoidance, goal-seeking, and arbitrates between them. The Vehicle encapsulates the state of the vehicle and provides dynamics simulation and feedforward. The Position Estimator implements the position estimation Kalman filter system.

Section 3: Perspectives on Intelligence

There exists in the literature at this point in the history of the field a clear spectrum of approaches to intelligent mobility from minimalist reactive approaches [8],[15] to more deliberative reasoned approaches [51],[77], [54]. Few groups propose that one approach or the other solves all problems, and, of course, there is considerable value in asking the question of how much can be accomplished within a single paradigm. Rather than taking an extreme position on the reactive-deliberative spectrum, the thesis argues that all of these elements have their place, and indeed are necessary, to solve the problem.

3.1 On Memory

While it may be possible to continually measure the environment at high rate and simply react to what is seen in each frame independent of other frames, there are reasons why memory is necessary in high-speed navigators. The perceptual complexity analysis demonstrates that high-speed navigators cannot afford the computation necessary to continually process the same scene geometry at sufficiently high resolution. Thus, for high-speed navigators, the memory involved in the mapping of the environment is an essential system capability.

3.2 On Deliberation

A high-speed vehicle incorporates its own memory as a matter of basic physics. There are limits on its maneuverability which arise fundamentally from the state memory which is the defining characteristic of dynamic systems. It has been shown that a mobility system which fails to model this memory will be unreliable. Unlike the environmental geometry memory mentioned above, the impact of this form of memory is that it requires some form of deliberation. In particular, for high-speed navigators, models of dynamics take the place of models of logical precedence used in AI in that they limit the states reachable in a small period of time from any given state. In such navigators, the deliberative reasoning about the future impact of current actions (that is implemented in feedforward models) is an essential capability.

3.3 On Reaction

No one would argue that a high-speed vehicle need not respond to external stimuli within finite time. This is the essence of obstacle avoidance. Yet, the latency models developed in the thesis and the precision timekeeping that has been incorporated seem more applicable to the control systems of fighter aircraft than to autonomous vehicles. For high-speed navigators, real-time response means more than reacting fast. Indeed, high-speed navigators must *reason about* their ability to respond and, at least on the current generation of vehicle testbeds, this reasoning (about response) is an essential capability.

Section 4: Fundamental Arguments

This section interprets the impact of the analysis on the design problem. It summarizes the most important arguments and the logic behind the design of the navigator.

4.1 Guaranteed Safety

One of the primary goals of the navigator has been to maximize speed. Driving fast requires:

- looking far ahead and/or reacting fast
- reducing the amount of scene geometry processed and/or processing it fast
- a sufficiently accurate and precise model of the environment and the vehicle

The design of the navigator starts with the adoption of the policy of **guaranteed safety** - the policy of guaranteeing that all of the above things remain true at all times while the system is in operation. Later analysis of this policy lead to the classification of the problem of driving fast as a real-time problem for two reasons:

- Response time is a major concern and must be reduced.
- Throughput is a major concern and must be reduced.

4.2 Response

The design takes the point of view that response time must be reduced. The problem of minimizing response time leads into a quest to reduce throughput, reduce acuity (resolution) and increase fidelity (accuracy).

4.3 Throughput

Throughput must be reduced because it affects response. Higher throughput takes more time to achieve in software so higher throughput causes longer response times. An effect called the **computational spiral effect** expresses the intuitive logic that a system that must react fast must cycle fast and one that must cycle fast must minimize its computational throughput. This logic is the organizing principle behind the design of the adaptive perception algorithm for perception, and the adaptive regard algorithm for planning. Both algorithms concentrate only on a focus of attention of minimal size in order to reduce throughput.

4.4 Acuity

Resolution affects throughput and therefore response. In the case of planning the area of the focus of attention divided by the resolution is the throughput required in range pixels. In the case of planning, the number of trajectory alternatives considered divided by the resolution is the throughput required in configuration checks. The previous paragraph argued that the area of the focus of attention must be minimized, whereas this paragraph argues that the resolution with which this area is checked must also be minimized.

The mapping from image space to world coordinates is nonlinear and it generates pixel footprints that are neither homogeneous nor isotropic on the groundplane. While this appears at first glance to be a major problem, when it is considered along with response and throughput requirements, this

sampling problem is revealed as much less severe. The important points here are that range must be large because of response and the width of the focus of attention must be narrow because of throughput.

4.5 Fidelity

Fidelity must be maximized for three separate reasons:

- Poor fidelity causes thrashing in the standard architectural model and hence affects throughput and therefore response.
- Poor fidelity is a major cause of collisions because the system has an inadequate understanding of its ability to control the vehicle.
- Poor fidelity causes instability in obstacle avoidance and goal seeking.

Given that fidelity must be maximized, the design was driven toward a state space model of vehicle dynamics. Another issue that arose was the fact that this model, necessary in its own right, was not solvable in reverse. For this reason, a formulation of planning in terms of optimal control and a generate and test search was introduced to allow using the forward model only. This approach ultimately had a beneficial affect on response because it allows a low resolution search of the feasible set without explicitly considering feasibility.

Section 5: Key Assumptions

Some of the most important assumptions used in the navigator are outlined below. This list is provided in order to:

- be explicit about the important assumptions
- identify the major failure modes of the system

5.1 Continuity Assumption

Many aspects of adaptive perception and feedforward control involve circular logic. For example, it is impossible to know where the vehicle will go until the terrain is known and it is impossible to know what image data to use until the vehicle trajectory is known. Luckily, conservative approximations can often be made to break this circle. When this is not feasible, information from the last cycle of computations can be used as an approximation for the current cycle. This assumption that, for example, system cycle time is a smooth function of time will be called the **continuity assumption**.

5.2 Terrain Smoothness Assumption

Some degree of **terrain smoothness assumption** must always be adopted. Use of a particular map resolution is tantamount to adopting this assumption. This assumption is critical to the successful operation of autonomous navigators but it also, unfortunately, implies that systems will fail with some regularity when this assumption is violated.

5.3 Obstacle Sparsity Assumption

The navigator minimizes its search through the use of a forward model of the vehicle and a gross sampling of the commands available at any instant of time. This sample of commands is kept small in order to minimize processing time, but it also implies that convoluted trajectories and trajectories that squeeze between narrowly spaced hazards will not be searched. In effect, real-time performance has lead to adopting the **obstacle sparsity assumption** as a pragmatic matter to maximize speed.

5.4 Small Incidence Angle Assumption

The **small incidence angle assumption** together with the 2-1/2D world assumption and the static environment assumption decouples the problem of selection from the problem of perception and makes adaptive perception possible. However, the first of these assumptions is not a strong one because it is always possible to use a more accurate kinematic model to transform a entire region of interest from vehicle coordinates to the image plane.

5.5 2-1/2D World Assumption

One of the most important assumptions of all is the assumption that the environment can be modelled as a sampled surface of the form $z = f(x,y)$. This assumption is implicit in most implementations of terrain maps. It is called the **2-1/2D world assumption**. If a terrain map is used, the range values can be *consistently* assumed to be monotonic in elevation angle and this can

be used to dramatically reduce perceptual processing because some pixels need never be visited at all. This **monotone range assumption** follows from the 2-1/2D world assumption and also provides the basis for ambiguity removal in phase ambiguous sensors like AM rangefinders.

5.6 Static Environment Assumption

Another very important assumption is the assumption that obstacles stay put as the vehicle drives past them. This permits the perception system to localize and resolve hazards only once and position them in the navigation frame. Without this assumption, it is necessary to continually reassess the geometry of the entire field of view.

Section 6: Conclusions

This section presents a short list of conclusions which seem most significant to the problem and most relevant to the more general problem of autonomous mobility.

- **Spiral Effect.** The computational spiral effect measures the cost of being smart in terms of its effect on reaction time. A tradeoff between throughput and response exists which can be managed by minimizing throughput in order to minimize reaction time.
- **Real-Time Approach.** Minimum reaction time will, in turn, give the highest possible speeds and the highest quality environmental sensing for any given sensor.
- **Complexity of Geometric Perception.** The policy of guaranteed vehicle safety implies a computational complexity of $O([TV]^N)$ for range image processing where T is the vehicle reaction time and V is the velocity. This result implies that increased vehicle speed will require nonlinear growth in computational bandwidth.
- **Fundamental Tradeoff.** This analysis identifies the fundamental tradeoff of finite computing resources as one of speed for either resolution or reliability.
- **Adaptive Perception.** The throughput problem of autonomous navigation can be eliminated at contemporary speeds by computational stabilization of the sensor sweep.
- **Computational Image Stabilization.** Adaptive perception techniques which computationally stabilize the vertical field of view provide the best of both worlds. They provide the high throughput necessary for high-speed motion and the wide field of view necessary for rough terrain.
- **Adaptive Stereo Perception.** The techniques of computational image stabilization have been applied to the stereo vision problem as it occurs in mobility scenarios. In rough numbers, adaptive sweep, implemented through a disparity window and appropriate postprocessing, improves throughput by a factor of 10. Similarly, adaptive scan, implemented through a vertical baseline and appropriate angular resolution promises to improve throughput by another factor of 10, although this has not been demonstrated conclusively here.
- **Adaptive Regard.** In a manner similar to the use of a focus of attention in perception, a focus of attention can be computed for obstacle avoidance that reflects the capacity of the vehicle to react at any given time. **Adaptive regard** places a limit on how close a vehicle should look for hazards because it cannot react inside of some distance. Thus, adaptive regard calls for all data inside some lower limit to be ignored and limits are placed on the extent of data processed beyond this minimum limit by considerations of both minimum planning throughput and range data quality.
- **Tactical Control Layer.** A modification to the standard architectural model of robotic systems has been proposed which connects strategic geometric reasoning to dynamic reactive control in an effective manner when the system under control exhibits poor command following. The problem is solved in this intermediate layer between AI and control, between reactive and deliberative approaches. An optimal solution to the problem includes elements of deliberation and elements of reactivity both.

- » **Actuation Space Search.** Dynamics of many kinds imply that the local planning problem is actually relatively easy from the point of view of search complexity. The planning "state space"¹ of the high-speed Ackerman vehicle is degenerate. Ordering heuristics generally optimize search by imposing the most constraining limits first and reducing the size of the search space as fast as possible. This principle is used here because once dynamically infeasible paths are eliminated, only a few remaining alternatives are spatially distinct enough to warrant consideration.
- **Dynamic Models.** The incorporation of dynamics models has generated a local navigation system which remains stable past the limits beyond which kinematically modelled systems became unstable. The use of dynamic models also makes obstacle avoidance more reliable in general by imparting to the system a more accurate understanding of its ability to respond.
- **Forward Modelling.** Physical dynamics amounts to an overwhelming constraint on the maneuverability of a high-speed vehicle. For a vehicle or operating regime for which classical path generation based on via points becomes difficult, forward modelling has the advantage that generated trajectories are feasible by construction.
- > **Arbitration.** The simultaneous satisfaction of hazard avoidance and goal seeking can cause contention for the absolute control of vehicle actuators, and in a practical system, this contention must be resolved through some arbitration mechanism. The problems of goal seeking, local path planning, and hazard avoidance have been unified into an optimal control context. In this context, a functional computed over the feasible set of response trajectories serves as the quantity to be optimized and the hazard avoidance mechanism specifies the confines of the feasible set.
- > **Hazard Space.** In strict mathematical terms, the configuration space (C-space) of AI planning is a subset of the state space (S-space) of multivariate control. It is a well-established technique to abstract a mobile robot into a point in six-dimensional position and attitude coordinates. The significance of hazard space (H-space) is that it performs the same function for dynamic planning that C-space performs for kinematic planning.
- **State Space Kalman Filter.** A definitive formulation of position estimation has been posed which is specifically designed for autonomous vehicles based on the state space Kalman filter. The major architectural advantage of this formulation is that state estimation is a completely separate process from the measurement model so it is inherent and straightforward to supply an interpolated position estimate at significantly higher rates than can be supplied by the underlying sensory hardware. Additionally, state vector augmentation techniques permit modelling the dynamics of sensors, removal of known systematic sensory errors, and implementation of self-calibration routines based on landmark observations.

1. In AI, the term "state space" has a slightly different connotation than in control theory.

PART VIII: Contributions

At the time of this writing, autonomous mobility is a field in its infancy. This is clear because the number of systems developed and documented to drive themselves outdoors can be counted on a single hand. In such a wide open field, opportunities to make a difference and influence the future are many. This section will present the work in the context of its contributions to the research.

Section 1: General Claims

1.1 Codification

The analysis section of this document is probably the most complete description of the problems, subproblems, issues, tradeoffs, design rules, and terminology of off-road mobility in print. This matters because any design process requires both requirements and constraints as a starting point and these two elements were not so comprehensively elaborated anywhere prior to this work.

1.2 Classification

It is significant that the problem has simply been classified as a real-time control problem. This matters because competing perspectives have been held in the past and a balanced view must weigh all perspectives. Regardless of whether any particular view becomes the dominant one, the importance and quantitative implications of response and throughput on the problem, and the significance of forward modelling to path planning will remain important.

1.3 Demonstration

Of course, two of the most important questions to ask before attempting any work are "is it worth doing at all" and "is it even possible". The system developed here has shown that off-road mobility is possible at 10 mph for usefully long excursions and because of this, the work has redefined, however humbly, the line between the possible and the impossible.

Section 2: Specific Claims

2.1 **Perspective**

The perspective of the work can be summarized in two sentences. First, high speed mobility is a problem with stringent real-time constraints. Second, in order to meet those constraints, an adaptive approach to perception and a control approach to obstacle avoidance are indicated. Simply put, this perspective has lead to a certain approach, and that approach is directly responsible for an improvement in the capabilities of autonomous navigators.

2.2 **Approach**

The computational image stabilization algorithms and the feedforward control algorithms are the key elements of the approach. These algorithms are new and are implemented in a variation on the classical hierarchical robotic control architecture.

2.3 **Generality**

Simply because current autonomous navigators must attempt to survive with inadequate perceptual acuity, adaptive perception will be important for the immediate future until such time as computer speed catches up. State space modelling is a type of model for vehicle dynamics that is far more general than competing approaches. It is likely to remain an important aspect of autonomous navigation whenever a vehicle exhibits deterministic behavior.

Section 3: Limitations / Future Work

While there are many remaining problems on the testbed vehicles that limit performance, some problems are clearly caused by software and ought to be solved in software.

3.1 Strategic Planning

While the problem of choosing the strategic goal to be sought has been clearly stated to be outside the scope of the work, it must be also stated that strategic planning is a critical system component in practice.

3.2 Stereo Vision

Much of the work undertaken here in stereo vision has been speculative and experimental in character. The emphasis has been to demonstrate feasibility but the considerable effort required to construct robust stereo ranging has not been expended.

3.3 Identification

The latency and actuator dynamics models used in the navigator suffer from the same problems of all models that attempt to be accurate - they need to be calibrated. In the long run, it will make sense to construct a system identifier that watches the vehicle being driven in order to calibrate these models most accurately.

3.4 Coordinated Control

Experience on the RATLER vehicle has pointed to a clear problem in the implementation of the navigator in that it assumes that coordinated control loops exist which minimize command following error after the system has issued its current command. If the system makes the claim of being integrated and comprehensive, then such loops must also be included in the package.

3.5 Map Matching

The algorithm used for merging images together is rudimentary - primarily because more sophisticated approaches actually resulted in worse performance. Nonetheless, it should be possible to incorporate at least the scatter matrix information to avoid the problems of registration at vertical surfaces.

3.6 Kalman Filter

The Kalman filter work undertaken has been only a small part of what should be done. Perhaps the most interesting area of future work is the use of visual landmarks to quantify the vehicle position. At the very least, feature tracking from image to image or the placement of landmarks in the test area could be investigated to quantify their value in outdoor settings.

PART IX: Appendices

The following appendices include detailed mathematical analysis and results that were deemed too detailed for the body of the document. These results will serve as useful references for those readers interested in reusing the details for their own purposes.

Chapter 1: Kinematic Transforms

Kinematic modelling is one of the most essential analytical tools of robotics. It is used for modelling mechanisms, actuators, and sensors for on-line control, off-line programming, and simulation purposes. A short tutorial on Homogeneous Transforms is presented covering the triple interpretation of a homogeneous transform as an operator, a coordinate frame, and a coordinate transform. The operator / transform duality is derived and its use in the Denavit Hartenberg convention is explained. Forward, inverse, and differential kinematics are derived for a simple manipulator to illustrate concepts.

A standard set of coordinate frames is proposed for wheeled mobile robots. It is shown that the RPY transform serves the same purpose as the DH matrix in this case. It serves to interface with vehicle position estimation systems of all kinds, to control and model pan/tilt mechanisms and stabilized platforms, and to model the rigid transforms from place to place on the vehicle. Forward and inverse kinematics and the Euler angle rate to the angular velocity transform are derived for the RPY transform.

Projective kinematics for ideal video cameras and laser rangefinders, and the imaging Jacobian relating world space and image space is derived. Finally, the kinematics of the Ackerman steer vehicle are presented for reference purposes.

Control of autonomous vehicles in 3D can require precise kinematic models of mechanisms, the image formation process, terrain following, steering kinematics, and more. This chapter provides the tools necessary to solve these problems in one place for reference purposes. It is both a tutorial and a reference for the transforms used in the RANGER vehicle controller.

Section 1: Notational Conventions

The 3 X 3 matrix R_a^b denotes a rotation matrix which transforms a displacement from its expression in coordinate system 'a' to its expression in coordinate system 'b'. The 4 X 4 matrix T_a^b denotes the homogeneous transform which transforms a vector from its expression in coordinate system 'a' to its expression in coordinate system 'b\ If \bar{p}^a denotes a point expressed in coordinate system 'a', then the notation for conversion of coordinates to coordinate system 'b' is easy to remember by considering that the a's cancel:

$$\bar{p}^b = T_a^b \bar{p}^a$$

The 4 X 4 matrix P_a^b denotes a nonlinear projection operator *represented* as a homogeneous transform. In such notation, the vector normalization step is implicit in the transform. The 3 X 1 vector f_a^b represents the translation vector from system 'a' to system 'b' expressed in system 'a' coordinates. The matrix J_a^b denotes the Jacobian of the transformation from system 'a' to system 'b\

The specification of derivatives will be necessarily loose. If X and y are scalars, \dot{X} and $\dot{\bar{y}}$ are vectors, and X and Y are matrices, then all of the following derivatives can be defined:

$\frac{\partial y}{\partial x}$	a partial derivative	$\frac{\partial y}{\partial \bar{x}}$	a gradient vector
$\frac{\partial}{\partial \bar{x}y}$	a vector partial derivative	$\frac{\partial y}{\partial \bar{x}}$	a Jacobian matrix
$\frac{dY}{dx}$	a matrix partial derivative	$\frac{\partial Y}{\partial \bar{x}}$	a Jacobian tensor

The notation ca and sa is shorthand for cos(a) and sin(a) respectively. Sums of angles, for example, will be indicated by cascaded subscripts as follows:

$$c_{123} = \cos (\psi_1 + \psi_2 + \psi_3)$$

Section 2: Homogeneous Coordinates and Transforms

Homogeneous coordinates are a method of representing 3D entities by the projections of 4D entities onto a 3D subspace. This section investigates why such an artificial construct has become the cornerstone of robot kinematic modelling.

IX **Points**

A point is a position in space. Points will be represented by column vectors such as:

$$\mathbf{p}_1 = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$$

where the superscript T denotes the matrix transpose and permits writing a column vector as a row vector.

2.2 Operators

An **operator** is any process which maps points onto other points. For the present purpose, operators will be limited to those which can be represented as matrices. The above point \mathbf{p}_1 can be altered in many different ways by multiplying it by a general 3 X 3 matrix T:

$$\mathbf{p}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = T_{p1} = \begin{bmatrix} t_{xx} & t_{xy} & t_{xz} \\ t_{yx} & t_{yy} & t_{yz} \\ t_{zx} & t_{zy} & t_{zz} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1 \mathbf{i} + \mathbf{V}_2 \mathbf{j} + \mathbf{V}_3 \mathbf{k} \\ x_1 + t_{zx}y_1 + t_{zy}z_1 \\ y_1 + t_{yx}x_1 + t_{yy}y_1 + t_{yz}z_1 \\ z_1 + t_{zx}x_1 + t_{zy}y_1 + t_{zz}z_1 \end{bmatrix}$$

Some simple operators are scale, reflection, rotation, projection, and shear. All of these can be generated from the above matrix by a suitable choice of the entries in the matrix.

2.3 Homogeneous Coordinates

Even though the above operator is very general, it is not general enough to be used conveniently in many robotics applications because it cannot represent a translation. That is, there exists no 3 X 3 matrix which adds a constant vector, like \mathbf{p}_3 to \mathbf{p}_1 . Such a translation could be expressed as a vector addition:

$$\mathbf{p}_2 = \mathbf{p}_1 + \mathbf{p}_3 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}$$

This cannot be done by a matrix because \mathbf{p}_3 cannot be represented, in general, as a linear combination of the elements of \mathbf{p}_1 - it is supposed to be independent of \mathbf{p}_1 .

However, the situation can be fixed with a standard trick. In **homogeneous coordinates**, an extra element can be added to each point to represent a kind of scale factor:

$$p_1 = \begin{bmatrix} x_1 & y_1 & z_1 & w_1 \end{bmatrix}^T$$

and it is conventional to consider that this 4-vector is projected into 3D by dividing by the scale factor:

$$Pl = \begin{bmatrix} x_1/w_1 & y_1/w_1 & z_1/w_1 \end{bmatrix}^T$$

Points are typically represented with a scale factor of 1. Thus:

$$p_1 = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \end{bmatrix}^T$$

is a point in homogeneous coordinates. It is also possible to represent a pure direction in terms of a point at infinity by using a scale factor of 0. Thus:

$$q_1 = \begin{bmatrix} x_1 & y_1 & z_1 & 0 \end{bmatrix}^T$$

is a direction in homogeneous coordinates.

2.4 Homogeneous Transforms

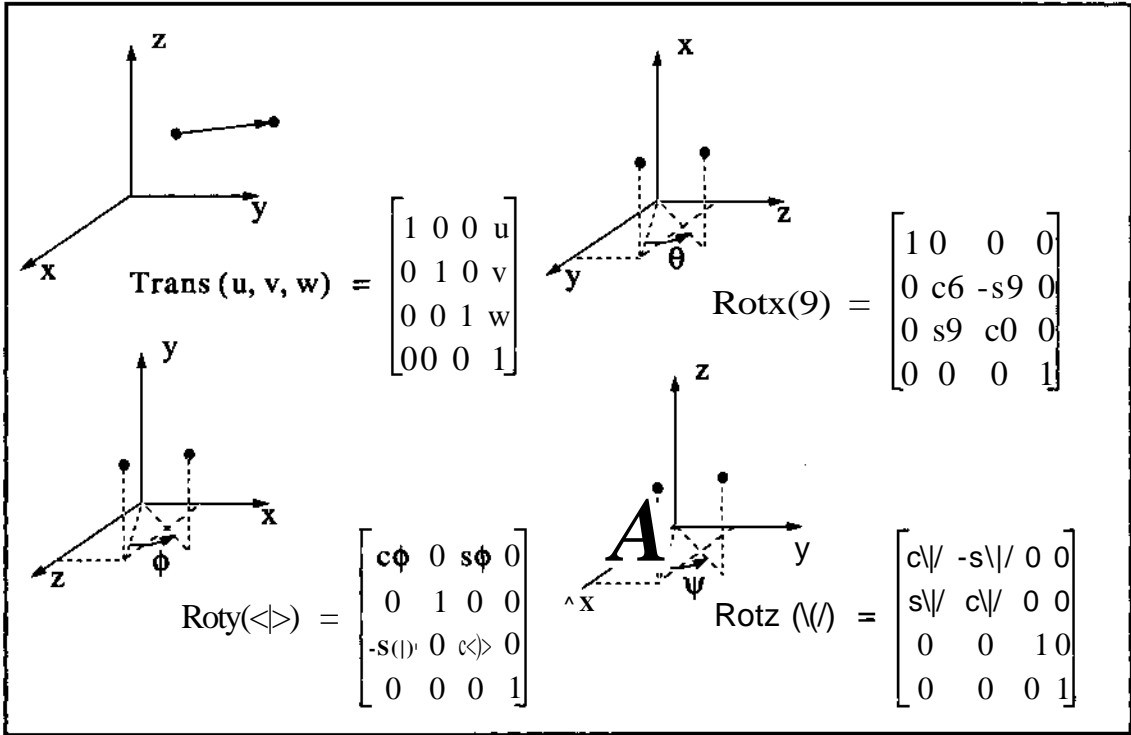
Using homogeneous coordinates, it is now possible to represent the addition of two vectors as a matrix operation, thus:

$$p_2 = p_1 + p_3 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} + \begin{bmatrix} x_3 \\ y_3 \\ z_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_3 \\ 0 & 1 & 0 & y_3 \\ 0 & 0 & 1 & z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \text{Trans}(p_3)p_1$$

where $\text{Trans}(p_3)$ is the homogeneous transform which is equivalent to a translation operator for the translation vector p_3 .

2.5 Homogeneous Transforms as Operators

Suppose it is necessary to move a point in some manner and express the result in the *same* coordinate system as the original point. This is the notion of an operator. The basic operators are translation along and rotation about any of the three axes. The following table gives the four elementary operators which are sufficient for the purposes of the document, and for almost any real problems. Operators are identified by an upper case first letter.



It is important to remember the precise semantics of these operators. They take a point expressed in some coordinate system, operate on it, and supply the result expressed in the same coordinate system.

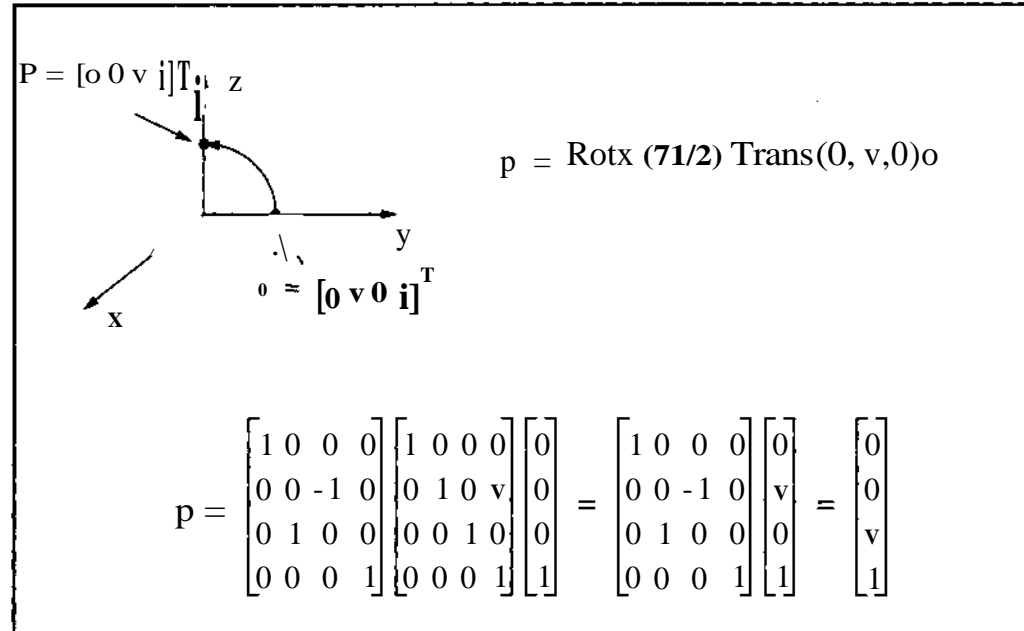
One of the most common operators used in robotics is a rotation followed by a translation. The homogeneous transform can be used to represent this operator as a single matrix. In general, many kinds of operators can be represented by homogeneous transforms. However, the operator interpretation is but one of three possible interpretations of the homogeneous transform.

2.6 Example • Operating on a Point

The homogeneous coordinates of the origin O are clearly:

$$o = [0 \ 0 \ 0 \ 1]^T$$

As an example of applying operators to a point, the following figure indicates the result of translating the origin along the y axis by V units and then rotating the resulting point by 90° around the x axis.



Notice that the operators are written in right to left order because this is the order in which they are applied to the original column vector representing the origin. The order is important because matrix multiplication is not commutative. Also notice that the initial point, intermediate point, and final result are all expressed in terms of the axes of the original coordinate system. Operators have *fixed axis semantics*.

2.7 Homogeneous Transforms as Coordinate Frames

Recall that homogeneous coordinates can be used to represent directions as well as points. This is done by using a zero scale factor. The unit vectors of a cartesian coordinate system can be considered to be directions. The x, y, and z unit vectors can be written as:

$$i = [1 \ 0 \ 0 \ 0]^T \quad j = [0 \ 1 \ 0 \ 0]^T \quad k = [0 \ 0 \ 1 \ 0]^T$$

Therefore, these three columns can be grouped together with the homogeneous coordinates of the origin to form an identity matrix:

$$[\mathbf{i} \ \mathbf{j} \ \mathbf{k} \ \mathbf{o}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{I}$$

Clearly, the columns of the identity matrix can be interpreted as the unit vectors and the origin of a coordinate system. The result of applying an operator to these four vectors has a similar interpretation that will be shown by example.

2.8 Example - Interpreting an Operator as a Frame

These four vectors can be transformed simultaneously by applying the same two operators to the identity matrix:

$$\mathbf{I}^1 = \text{Rotx}(\pi/2) \text{Trans}(0, v, 0) \mathbf{I}$$

$$\mathbf{I}^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This result is just the product of the original two operators. Each column of this result is the transformation of the corresponding column in the original identity matrix. Thus the transformed vectors are:

$\mathbf{i}' = [1 \ 0 \ 0 \ 0]^T$
 $\mathbf{j}' = [0 \ 0 \ 1 \ 0]^T$

$\mathbf{k}' = [0 \ -1 \ 0 \ 0]^T$
 $\mathbf{o}' = [0 \ 0 \ v \ 0]^T$

It has been shown that the column vectors of any compound fundamental operator are just the homogeneous coordinates of the transformed unit vectors and the origin. This is because the operators apply to any vectors - including the origin and the unit vectors. The result can be visualized by drawing the new transformed axes at the new position.

2.9 The Coordinate Frame

So far in the discussion, there are two complementary interpretations of exactly the same 4 X 4 matrix. It can be an operator which moves points and vectors around, or it can be a representation of a cartesian coordinate system positioned somewhere in space relative to another one. Cartesian coordinate systems positioned in space are a central concept in 3D kinematics. They encode both a position and an attitude. With an encoded position and attitude available, it becomes possible to talk about the *state of motion* of the origin in terms of translation and orientation and all of their associated time derivatives. Therefore, this entity embodies the properties of a *frame of reference*.

With a set of three orthogonal unit vectors, it is possible to represent an arbitrary vector quantity in terms of its projections onto these axes. Therefore, this entity also embodies the properties of a cartesian *coordinate system*. This moving set of unit vectors is often called a **coordinate frame** or simply a **frame**. Often, imaginary frames are embedded in the links of a mechanism in order to keep track of its configuration. These *embedded frames* are central to the study of manipulators, legs and other mechanisms.

2.10 Homogeneous Transforms as Coordinate Transforms

There is a third interpretation of a homogeneous transform. It was shown that they move points and move and orient frames and that they directly represent frames. However, if they move frames, one can think about the original frame and the transformed frame as two different frames and then ask about the relationship between the coordinates of any point in each frame. Let the original frame be called 'a' and the transformed one be called 'b'.

Let a general point p be expressed in the coordinates of frame 'b'. This will be represented by a superscript b. Its coordinates in this frame are:

$$p^b = \begin{bmatrix} x^b & y^b & z^b & 1 \end{bmatrix}$$

Alternately, it can be written as a true physical vector thus:

$$\vec{p}^b = x^b \hat{i}_b + y^b \hat{j}_b + z^b \hat{k}_b$$

This vector can be expressed in the coordinates of frame 'a' by expressing the unit vectors of frame 'b' in the coordinates of frame 'a' and adding the position of the origin, thus:

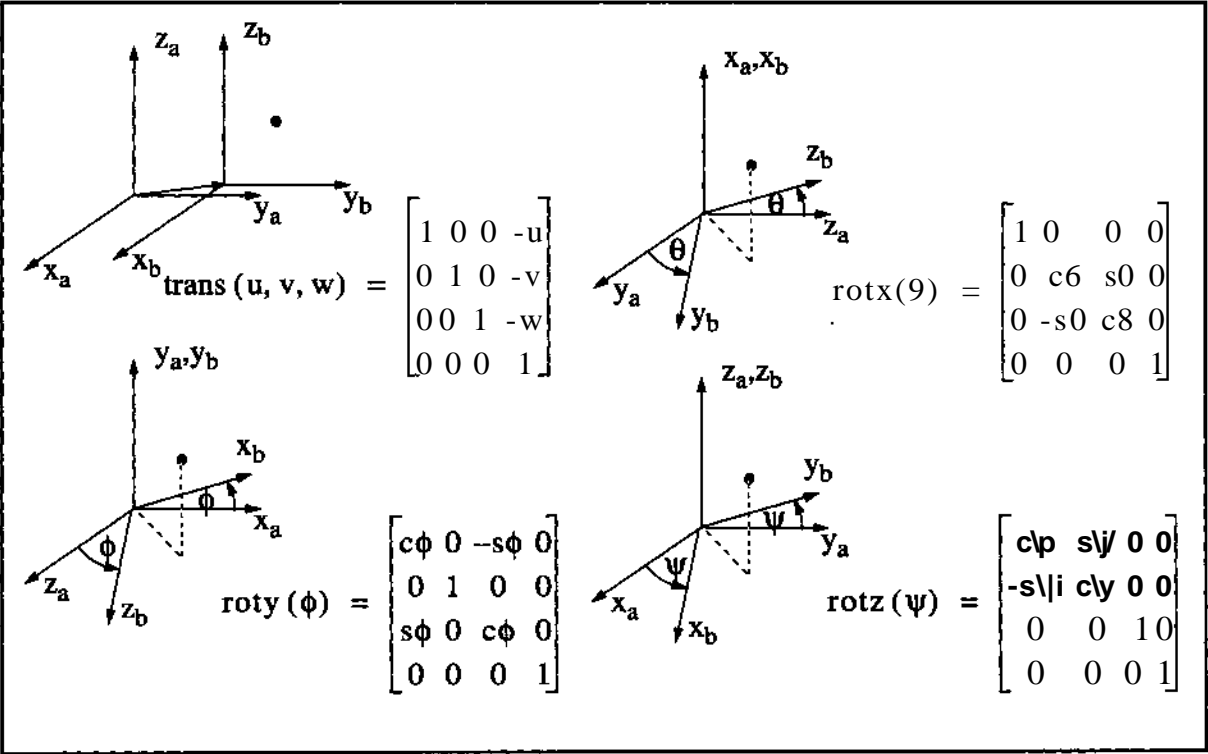
$$\vec{p}^a = x^b \hat{i}' + y^b \hat{j}' + z^b \hat{k}' + o'$$

but this is just the same as if a matrix multiplication were used:

$$p^a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^b \\ y^b \\ z^b \\ 1 \end{bmatrix}$$

It has been shown that the same homogeneous transform that moves frame 'a' into coincidence with frame 'b' also converts the coordinates of points *in the opposite direction* - from frame 'b' to frame 'a'. The reason for the opposite sense in the interpretation is that moving a point "forward" in a coordinate system is completely equivalent to moving the coordinate system "backward".

It is easy to show that the inverse of a rotation matrix is equal to its transpose. This is called the property of *orthogonality*. This leads to a second set of 4 matrices, the fundamental transforms, which are the inverses of the operators and which therefore convert coordinates from 'a' to 'b'. These are given in the table below. In the document, the transforms will be denoted with lowercase first letters in their names.

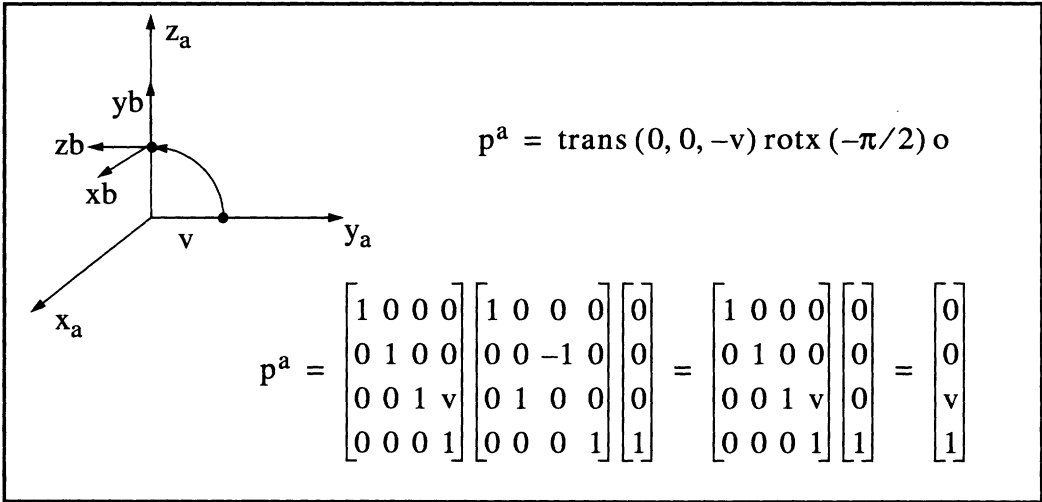


2.11 Example - Transforming the Coordinates of a Point

Again, the homogeneous coordinates of the origin O are:

$$o = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$$

As an example of applying transforms to a point, the following figure indicates the result of transforming the origin of frame 'b' into the coordinates of frame 'a' where the two frames are related by the same sequence of operators used earlier. Frame 'b' is moved into coincidence with frame 'a' by first rotating by -90° around the x axis of frame 'b', and then translating by $-v$ along the z axis of the intermediate frame.



Notice that the transforms are written in right to left order because this is the order in which they are applied to the original column vector representing the origin. The order is important because matrix multiplication is not commutative. Also notice that, for transforms, each new transform is interpreted as if it were applied to the last frame in the sequence of frames that ultimately bring the first into coincidence with the last. Transforms have *moving axis semantics*.

2.12 Inverse of a Homogeneous Transform

The homogeneous transforms that will be used in this document will all be structured according to the following template:

	Rotation Matrix	Position Vector
	Perspective	Scale

The scale factor will almost always be 1 and the perspective part will be all zeros except when modelling cameras. Under these conditions, it is easy to show by multiplying the inverse by the original matrix, that the inverse is:

$$\begin{bmatrix} R & \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R^T & -R^T \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This is very useful for converting from a matrix that converts coordinates in one direction to one that converts coordinates in the opposite direction. Throughout the document, remember that it is trivial to reverse the sense of a coordinate transform.

2.13 A Duality Theorem

The careful reader has probably noticed that operators and transforms are inverses. Whether a particular matrix is considered to be operator or transform is a matter of taste. This has an important implication for kinematic modelling that will be demonstrated by example. Using the previous example, the moving axis operations which bring frame 'a' into coincidence with frame 'b' are:

- translate v units along the z axis
- rotate 90° around the new x axis

The complete transform that converts coordinates from frame 'a' to frame 'b' can be written as:

$$T_a^b = \text{rotx}(\pi/2) \text{trans}(0, 0, v)$$

This can be rewritten in terms of operators as:

$$T_a^b = \text{Rotx}(-\pi/2) \text{Trans}(0, 0, -v)$$

The inverse of this matrix will convert coordinates from frame 'b' to frame 'a', and as was shown earlier, also represents the position and attitude of frame 'b' with respect to frame 'a'. Using the well known method for inverting a matrix product, this inverse matrix can be expressed as follows:

$$\begin{aligned} T_b^a &= \text{Trans}(0, 0, -v)^{-1} \text{Rotx}(-\pi/2)^{-1} \\ T_b^a &= \text{Trans}(0, 0, v) \text{Rotx}(\pi/2) \end{aligned}$$

So that the transform from 'b' to 'a' can be written in terms of transforms from right to left or in terms of operators *in the reverse order*. The latter view is traditional in robotics and it will be used in the next section.¹

1. One implication of this is that any set of rotations about fixed axes is equivalent to the same set executed in the reverse order about moving axes.

Section 3: Forward Kinematics

Forward kinematics is the relatively straightforward process of chaining homogeneous transforms together in order to represent the articulations of a mechanism or simply to represent the fixed transformation between two frames. In this process, the joint variables are given, and the problem is to find the transform.

3.1 Nonlinear Mapping

It is clear that it is possible to transform coordinates between coordinate systems which are fixed in position and attitude with respect to each other. It is also possible to perform elementary operations with homogeneous transforms. Ultimately, this is possible because such operations are *linear*. That is, the result is always a weighted sum of the original point or vector.

Most mechanisms, however, are not linear. They are composed of one or more rotary degrees of freedom. It turns out that the homogeneous transform can still be used to model such complex devices if they are viewed in a different way. In the study of mechanisms, the parameters of the rotations and translations between the frames are considered to be variables. In this sense, the problem is one of understanding the variation in a homogeneous transform when it is considered to be a function of one or more variables.

In the study of mechanisms, the real world space in which a mechanism moves is often called **task space**, and points and vectors are normally expressed in terms of cartesian coordinates. However, the mechanism articulations are most easily expressed in terms of angles and displacements along axes. This space is called **configuration space**. So far, we have studied linear transformations within task space. This section and the following sections will consider the more difficult problem of expressing the relationship between task space and configuration space. This problem is one of multidimensional nonlinear mapping.

3.2 Mechanism Models

It is natural to think about the operation of a mechanism in a moving axes sense - because most mechanisms are built that way. That is, the position and orientation of any link in a kinematic chain is dependent on all other joints that come before it in the sequence.

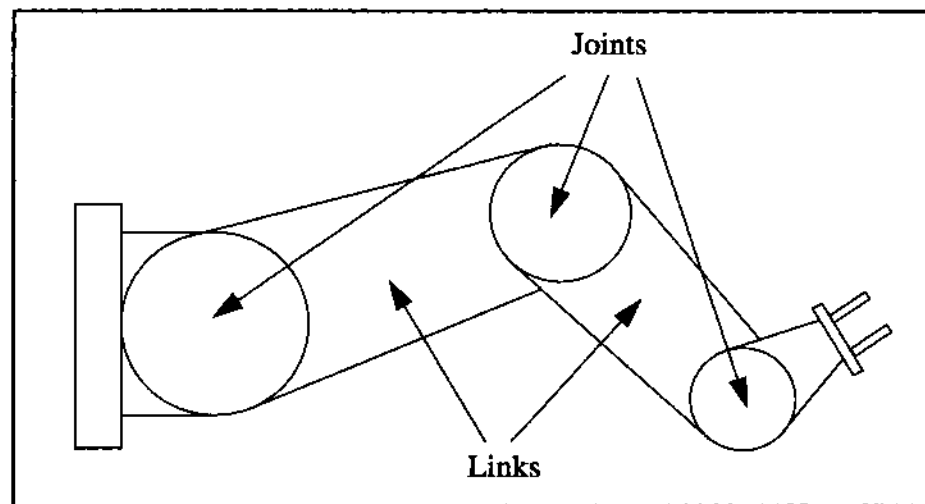
Conventionally, one thinks in terms of moving axis operations applied to frames embedded in the mechanism which move the first frame sequentially into coincidence with all of the others in the mechanism. Then, a sequence of operators is written to represent the mechanism kinematics. The conventional rules for modelling a sequence of connected joints are as follows:

- assign embedded frames to the links in sequence such that the operations which move each frame into coincidence with the next are a function of the appropriate joint variable
- write the operator matrices which correspond to these operations in *left to right order*

This process will generate the matrix that represents the position of the last embedded frame with respect to the first, or equivalently, which converts the coordinates of a point from the last to the first. This matrix will be called the **mechanism model**.

3.3 Modelling a Mechanism

In 1955, J. Denavit and R. S. Hartenberg [14] first proposed the use of homogeneous transforms to represent the articulations of a mechanism, and this form of model has been used almost universally since. A mechanism is considered to be any collection of joints, either linear or rotary, joined together by links. The total number of movable joints is called the number of **degrees of freedom**.



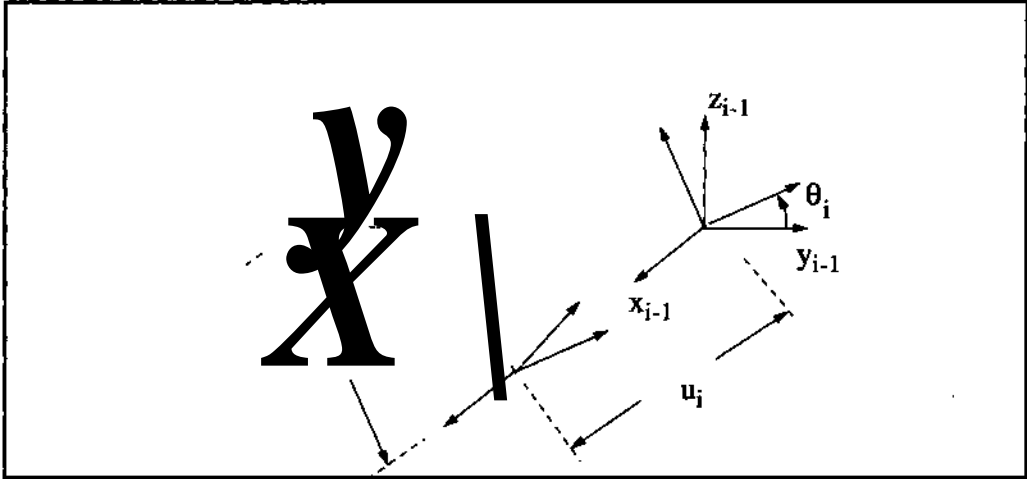
3.4 Denavit Hartenberg Convention for Mechanisms

It is conventional in many aspects of robotics to use a special product of four fundamental operators as a basic conceptual unit. The resulting single matrix can represent the most general spatial relationship between two frames. This convention with a small number of associated rules for assignment of embedded frames has come to be called the Denavit-Hartenberg (DH) convention.

For two frames positioned in space, the directions of, say, the z axes of two frames define two lines in 3D space. For nonparallel axes, there is a well-defined measure of the distance between them given by the length of their mutual perpendicular. Let the first frame be called frame $i-1$ and the second frame i . The first can be moved into coincidence with the second by a sequence of 4 operations:

- rotate around the x^i axis by an angle θ_i
- translate along the x^i axis by a distance a_i
- rotate around the new z axis by an angle ϕ_i
- translate along the new z axis by a distance d_i

The following figure indicates this sequence graphically.



The matrix, conventionally called A_i , that converts coordinates from frame i to frame $i-1$ can be written using the rules for forward kinematic modelling given previously.

$$A_i = T_i^{i-1} = \text{Rot}_x(\theta_i) \text{Trans}(a_i, 0, 0) \text{Rot}_z(\alpha_i) \text{Trans}(0, 0, d_i)$$

$$A_i = T_i^{i-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_i & -s\theta_i & 0 \\ 0 & s\theta_i & c\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\alpha_i & -s\alpha_i & 0 & 0 \\ s\alpha_i & c\alpha_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

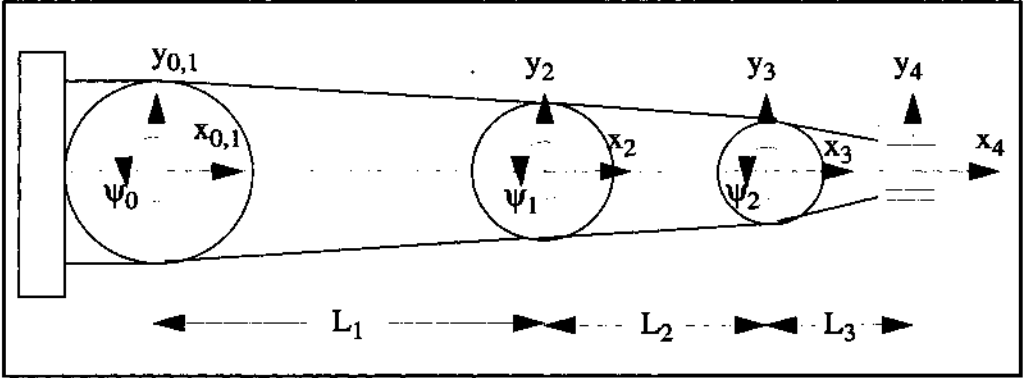
$$A_i = T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_i \\ d_i s\theta_i & d_i c\theta_i & -s\alpha_i & -s\alpha_i d_i \\ s\theta_i c\alpha_i & s\theta_i s\alpha_i & c\alpha_i & c\alpha_i d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- This matrix has the following interpretations:
- its columns represent the axes and origin of frame i expressed in frame $i-1$ coordinates
 - it converts coordinates from frame i to frame $i-1$

3.5 Examples 3-Link Planar Manipulator

Although the homogeneous transform matrices can represent any complex 3D ma. process of writing forward kinematic equations will be illustrated for a simple 2D ma. composed of three rotary joints, three links and an end effector. Coordinate frames are assign, the center of each rotary joint and the datum angle for each is set to zero as illustrated in the figuiv below.

It is conventional to assign the z axis of a frame so that the associated degree of freedom, linear or rotary, coincides with it. Also, the x axis is normally chosen so that it points along the mutual perpendicular. In this case, this means that frame 1 is a rotated version of frame 0:



The joint parameters are indicated in the table below:

Table 19: 3-Link Planar Manipulator

Link	6	u	V	w
0	0	0	V_i	0
1	0	L_i	V₂	0
2	0	L₂	V₃	0
3	0	L₃	0	0

This gives the following Denavit-Hartenberg matrices. Their inverses are computed from the inverse formula because they will be useful later:

$$\begin{aligned}
 A_1 &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_1^{-1} &= \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_2 &= \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_2^{-1} &= \begin{bmatrix} c_2 & s_2 & 0 & -c_2 L_1 \\ -s_2 & c_2 & 0 & s_2 L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_3 &= \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_3^{-1} &= \begin{bmatrix} c_3 & s_3 & 0 & -c_3 L_2 \\ -s_3 & c_3 & 0 & s_3 L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_4 &= \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & V &= \begin{bmatrix} 1 & 0 & 0 & -L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

The position and orientation of the end effector with respect to the base is given by:

$$\begin{aligned}
 T_4^0 &= T_1^0 T_2^1 T_3^2 T_4^3 = A_1 A_2 A_3 A_4 \\
 T_4^0 &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_4^0 &= \begin{bmatrix} c_{123} & -s_{123} & 0 & (c_{123}L_3 + c_{12}L_2 + c_1L_P) \\ s_{123} & c_{123} & 0 & (s_{123}L_3 + s_{12}L_2 + s_1L_P) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Section 4: Inverse Kinematics

Inverse kinematics is the problem of finding the joint parameters given only the numerical values of the homogeneous transforms which model the mechanism. In practice, this is useful because it solves the problem of where to drive the joints in order to get the hand of an arm or the foot of a leg in the right place, at the right orientation.

Proficiency with inverse kinematics requires a degree of skill and practice and there are few general guidelines that can be given. This section will discuss this difficult problem and continue with the simple example.

4.1 Existence and Uniqueness

The inverse kinematic problem is considerably more difficult than the forward one because it involves the solution of nonlinear equations. In general, there is no guarantee that nonlinear equations have a solution, and if they do, it may not be unique. It has been shown theoretically by Pieper [67], that a six degree of freedom manipulator for which the last three joints intersect at a point is always solvable. Most manipulators are constructed in this way, so most are solvable.

Any real mechanism has a finite reach, so it can only achieve positions in a region of space known as the **workspace**. Unless the initial transform to be solved is in the workspace, the inverse kinematic problem will have no solution. This manifests itself, for example, as inverse sines and cosines of arguments greater than 1. Further, mechanisms with many rotary joints will often have more than one solution and these solutions which will exhibit various forms of symmetry. The latter case is known as **redundancy**.

4.2 Technique

This problem would be very difficult to solve without the discipline afforded by the use of homogeneous transforms. Using the DH convention, a mechanism can be "solved" more or less one joint at a time by a process of rewriting the forward kinematics equations in several different ways.

Any DH matrix has only six degrees of freedom. The rotation matrix part is constrained to be orthonormal. That is, the rows or columns must be of unit length and mutually perpendicular. The bottom row is constrained to contain zeros and ones. The position vector contains three independent values.

The inverse kinematic problem is solved by rewriting the forward transform in many different ways to attempt to isolate unknowns. Although there are only 6 independent relationships, they can be rewritten in many different ways.

Using a three degree of freedom mechanism for example, the forward kinematics can be written in all of these ways by premultiplying or postmultiplying by the inverse of each link transform in turn:

$$T_4^0 = A_1 A_2 A_3 A_4$$

$$A_1^{-1} T_4^0 = A_2 A_3 A_4$$

$$A_2^{-1} A_1^{-1} T_4^0 = A_3 A_4$$

$$A_3^{-1} A_2^{-1} A_1^{-1} T_4^0 = A_4$$

$$A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} T_4^0 = I$$

$$T_4^0 = A_1 A_2 A_3 A_4$$

$$T_4^0 A_4^{-1} = A_1 A_2 A_3$$

$$T_4^0 A_4^{-1} A_3^{-1} = A_1 A_2$$

$$T_4^0 A_4^{-1} A_3^{-1} A_2^{-1} = A_1$$

$$T_4^0 A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} = T_1^0$$

These are, of course, redundant specifications of the same set of equations, so they contain no new information. However, they often generate equations which are easy to solve because the DH convention tends to isolate each joint. It is conventional to use the left column of equations, but the right column is easier to follow in our case, so it will be used.

4.3 Example: 3 Link Planar Manipulator

The process is started by assuming that the forward kinematic solution is known. Names are assigned to each of its elements as follows:

$$T_4^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For our example, we know that some of these are zeros and ones, so that the end effector frame is:

$$T_4^0 = \begin{bmatrix} r_{11} & h_i & 0 & P_x \\ r_{21} & r_{22} & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The first equation is already known:

$$T_4^0 = A_1 A_2 A_3 A_4$$

$$\begin{bmatrix} r_{11} & h_i & 0 & p_x \\ r_{21} & x_n & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & s_{123} & 0 & (c_{123}L_3 + 0^2 + 0^2) \\ s_{123} & c_{123} & 0 & (s_{123}L_3 + s_{12}L_2 + s_{11}L_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the (2,1) and (1,1) elements we have:

$$\psi_{123} = \operatorname{atanl}\{x_{2v}x_n\}$$

The next equation is:

$$T_j A_4^1 = A_1 A_2 A_3$$

$$\begin{bmatrix} r_{11} & h_i & 0 & -r_{11}L_3 + p_x \\ r_{21} & x_n & 0 & -r_{21}L_3 + p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & s_{123} & 0 & c_1(c_{12}L_2 + L_1) - s_1(0^2) \\ s_{123} & c_{123} & 0 & s_1(c_{12}L_2 + L_1) + c_1(s_{12}L_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & h_i & 0 & -r_{11}L_3 + p_x \\ r_{21} & x_n & 0 & -r_{21}L_3 + p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & s_{123} & 0 & c_{12}L_2 + c_{11}L_1 \\ s_{123} & c_{123} & 0 & s_{12}L_2 + s_{11}L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the (1,4) and (2,4) elements:

$$k_1 = -r_{11}L_3 + p_x = c_{12}L_2 + c_{11}L_1$$

$$k_2 = -r_{21}L_3 + p_y = s_{12}L_2 + s_{11}L_1$$

These can be squared and added to yield:

$$\begin{aligned} k_1^2 + k_2^2 &= L_2^2 + L_1^2 + 2L_2L_1(c_1c_{12} + s_1s_{12}) \\ k_1^2 + k_2^2 &= L_2^2 + L_1^2 + 2L_2L_1c_2 \end{aligned}$$

And this gives the angle ψ_2 as:

$$\psi_2 = \text{acos} \left[\frac{\left(k_1^2 + k_2^2 \right) - \left(L_2^2 + L_1^2 \right)}{2L_2L_1} \right]$$

The result implies that there are two solutions for this angle which are symmetric about zero. These correspond to the elbow up and elbow down configurations. Now, before the expressions were reduced to include a sum of angles, they were:

$$\begin{aligned} k_1 &= -r_{11}L_3 + p_x = c_1(c_2L_2 + L_1) - s_1(s_2L_2) \\ k_2 &= -r_{21}L_3 + p_y = s_1(c_2L_2 + L_1) + c_1(s_2L_2) \end{aligned}$$

With ψ_2 now known, these can be written as:

$$\begin{aligned} c_1k_3 - s_1k_4 &= k_1 \\ s_1k_3 + c_1k_4 &= k_2 \end{aligned}$$

This is one of the standard forms that recur in inverse kinematics problems for which the solution is:

$$\psi_1 = \text{atan2}[(k_2k_3 - k_1k_4), (k_1k_3 + k_2k_4)]$$

Finally, the last angle is:

$$\psi_3 = \psi_{123} - \psi_2 - \psi_1$$

4.4 **Standard Forms**

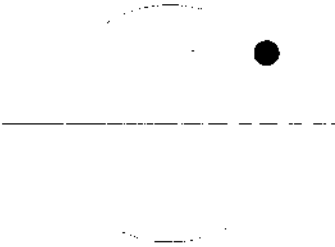
There are a few forms of trigonometric equations that recur in most mechanisms. The following set of solutions is sufficient for most applications. One of the most difficult forms, solved by square and add, was presented in the example. In the following, the letters a, b, and c represent arbitrary known expressions.

4.4.1 **Explicit Tangent**

This form generates a single solution because both the sine and cosine are fixed in value. It can arise in the last two joints of a three-axis wrist, for example. The equation:

$$\begin{aligned} a &= c_n \\ b &= s_n \end{aligned}$$

has the trivial solution:

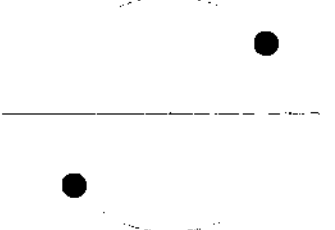
$$\psi_n = \operatorname{atan2}(b, a)$$


4.4.2 **Point Symmetric Redundancy**

This form generates two solutions that are symmetric about the origin. It can arise in the shoulder or hip joint, for example. The equation:

$$s_n a - c_n b = 0$$

can be solved by isolating the ratio of the two trig functions. There are two angles in one revolution which have the same tangent so the two solutions are:

$$\begin{aligned} \psi_n &= \operatorname{atan2}(b, a) \\ \psi_n &= \operatorname{atan2}(-b, -a) \end{aligned}$$


4.4.3 Line Symmetric Redundancy

This form generates two solutions that are symmetric about an axis because the sine or cosine of the deviation from the axis must be constant. The sine case will be illustrated. It can arise in the shoulder or hip joint, or in an elbow or knee joint, for example. The equation:

$$s_n a - c_n b = c$$

can be solved by the trig substitution:

$$a = r \cos (\theta) \qquad b = r \sin (\theta)$$

where:

$$r = \pm \text{sqrt} (a^2 + b^2) \qquad \theta = \text{atan2} (b, a)$$

This gives:

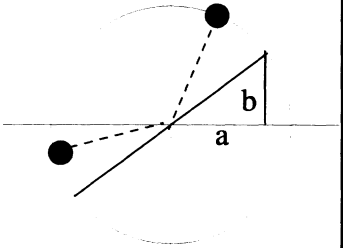
$$\begin{aligned} s_n c \theta - c_n s \theta &= c / r \\ s (\theta - \psi_n) &= c / r \end{aligned}$$

So the cosine is:

$$c (\theta - \psi_n) = \pm \text{sqrt} (1 - (c / r)^2)$$

This gives the result:

$$\psi_n = \text{atan2} (b, a) - \text{atan2} [c, \pm \text{sqrt} (r^2 - c^2)]$$



Section 5: Differential Kinematics

Differential kinematics is the study of the derivatives of kinematic models. These derivatives are called **Jacobians** and they have many uses ranging from:

- resolved rate control
- sensitivity analysis
- uncertainty propagation
- static force transformation .

5.1 Derivatives of Fundamental Operators

The derivatives of the fundamental operators with respect to their own parameters will be important. They can be used to compute derivatives of very complex expressions by using the chain rule of differentiation. For reference, they are quoted below:

$\frac{\partial}{\partial u} \text{Trans}(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\frac{\partial}{\partial \theta} \text{Rotx}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -sG & -cG & 0 \\ 0 & cG & -sG & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$\frac{\partial}{\partial v} \text{Trans}(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\frac{\partial}{\partial \phi} \text{Roty}(\phi) = \begin{bmatrix} -s\phi & 0 & c\phi & 0 \\ 0 & 1 & 0 & 0 \\ -c\phi & 0 & -s\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$\frac{\partial}{\partial w} \text{Trans}(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\frac{\partial}{\partial \psi} \text{Rotz}(\psi) = \begin{bmatrix} -s\psi & -c\psi & 0 & 0 \\ c\psi & -s\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Similar expressions can be generated for the fundamental transforms.

5.2 The Mechanism Jacobian

If a sequence of joints can be represented by the product of a series of homogeneous transforms, it is natural to ask about the effect of a differential change in joint variables on the position and orientation of the end of the mechanism. A frame matrix represents orientation indirectly in terms of three unit vectors, so an extra set of equations is required to extract three angles from the rotation matrix in order to represent orientation. In terms of position, however, the last column of the mechanism model gives the position of the end effector with respect to the base.

In general, let a mechanism have variables represented by the vector \bar{q} , and let the position and orientation, or **pose**, of the end of the mechanism be given by the vector \bar{X} . Then, the end effector position is given by:

$$\bar{x} = F(\bar{q})$$

where the nonlinear multidimensional function F comes from the mechanism model. The Jacobian matrix is a multidimensional derivative defined as:

$$J = \frac{\partial \bar{x}}{\partial \bar{q}} = \frac{\partial F(\bar{q})}{\partial \bar{q}} = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \dots & \frac{\partial x_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial q_1} & \dots & \frac{\partial x_n}{\partial q_n} \end{bmatrix}$$

The differential mapping from small changes in \bar{q} to the corresponding small changes in \bar{X} is:

$$d\bar{x} = Jd\bar{q}$$

The Jacobian also gives velocity relationships via the chain rule of differentiation as follows:

$$\frac{d}{dt}\bar{x} = \left(\frac{\partial}{\partial \bar{q}}\bar{x}\right)\left(\frac{d}{dt}\bar{q}\right)$$

which maps joint rates onto end effector velocity. Note that the Jacobian is nonlinear in the joint variables, but linear in the joint rates. This implies that reducing the joint rates by half reduces the end velocity by exactly half and preserves the direction.

5.3 Singularity

Redundancy takes the form of **singularity** of the Jacobian matrix in the differential kinematic solution. A mechanism can lose one or more degrees of freedom:

- at points where two different inverse kinematic solutions converge
- when joint axes become aligned or parallel
- when the boundaries of the workspace are reached

Singularity implies that the Jacobian loses rank and is not invertible. At the same time, the inverse kinematic solution tends to fail because axes become aligned, and infinite rates can be generated by rate control laws.

5.4 Example: 3-Link Planar Manipulator

For the example manipulator, the last column of the manipulator model gives the following two equations:

$$\begin{aligned} x &= (c_{123}L_3 + c_{12}L_2 + c_1L_1) \\ y &= (s_{123}L_3 + s_{12}L_2 + s_1L_1) \end{aligned}$$

which can be differentiated with respect to θ_1, θ_2 , and θ_3 in order to determine the velocity of the end effector as the joints move. The solution is:

$$\begin{aligned} \dot{x} &= -(s_{123}\dot{\theta}_{123}L_3 + s_{12}\dot{\theta}_{12}L_2 + s_1\dot{\theta}_1L_1) \\ \dot{y} &= (c_{123}\dot{\theta}_{123}L_3 + c_{12}\dot{\theta}_{12}L_2 + c_1\dot{\theta}_1L_1) \end{aligned}$$

which can be written as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} (-s_{123}L_3 - s_{12}L_2 - s_1L_1) & (-s_{123}L_3 - s_{12}L_2) & -s_{123}L_3 \\ (c_{123}L_3 + c_{12}L_2 + c_1L_1) & (c_{123}L_3 + c_{12}L_2) & c_{123}L_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

5.5 Jacobian Determinant

It is known from the implicit function theorem of calculus that the ratio of differential volumes between the domain and range of a multidimensional mapping is given by the Jacobian determinant. This quantity has applications to a technique of navigation and ranging called **triangulation**.

Thus the product of the differentials forms a volume in both configuration space and in task space. The relationship between them is:

$$(dx_1dx_2...dx_n) = |J|(dq_1dq_2...dq_m)$$

5.6 .Tacobian Tensor

At times it is convenient to compute the derivative of a transform matrix with respect to a vector of variables. In this case, the result is the derivative of a matrix with respect to a vector. For example:

$$\frac{\partial}{\partial \vec{q}} [T(\vec{q})] = \left[\frac{\partial T_{ij}}{\partial q_k} \right]$$

This is a three-dimensional cube of numbers which can loosely be called a tensor. The mechanism model itself is a matrix function of a vector.

For example, if:

$$T(\vec{q}) = A_1(q_1) A_2(q_2) A_3(q_3)$$

Then there are three slices of the tensor, each a matrix, given by:

$$\begin{aligned} \frac{\partial T}{\partial q_1} &= \frac{\partial A_1}{\partial q_1} A_2 A_3 \\ \frac{\partial T}{\partial q_2} &= A_1 \frac{\partial A_2}{\partial q_2} A_3 \\ \frac{\partial T}{\partial q_3} &= A_1 A_2 \frac{\partial A_3}{\partial q_3} \end{aligned}$$

Section 6: Vehicle Kinematics

Some aspects of the kinematics of moving vehicles are simpler than that of mechanisms, others are more difficult. This section presents an abbreviated discussion of the kinematic transforms necessary for control of a vehicle in 3D.

6.1 Axis Conventions

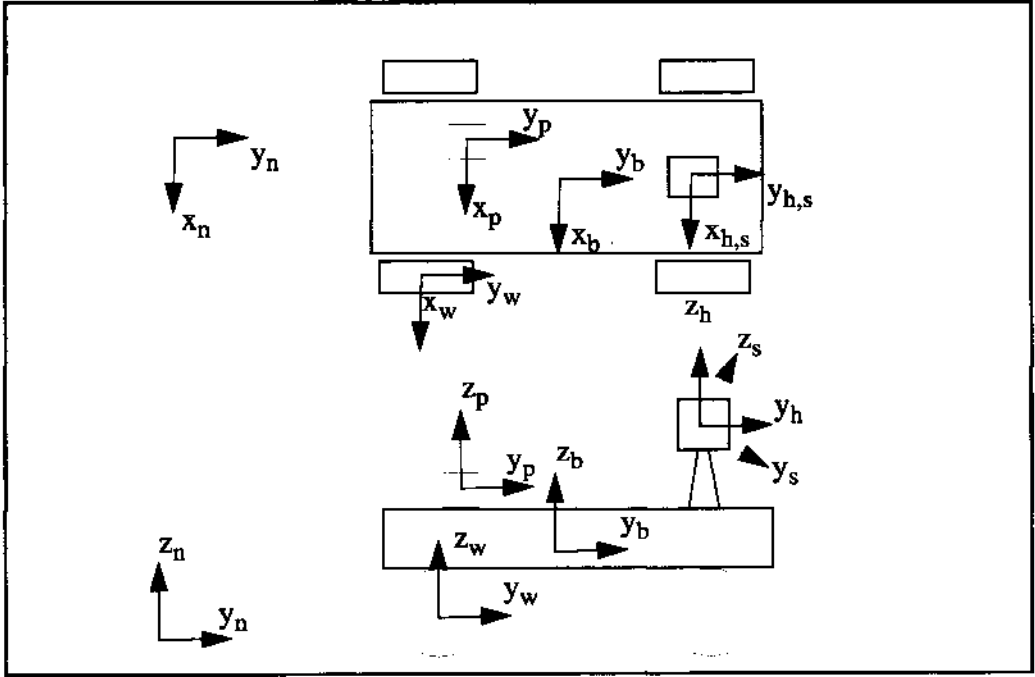
There are at least two prevailing conventions for the assignment of axes for the coordinate systems of vehicles. In aerospace vehicles, it is conventional to point the z axis downward, and this makes it natural to point the x axis forward and the y axis out the right side of the vehicle. The convention used here is that the z axis points up, y forward, and x out the right side. This has the advantage that the projection of 3D information onto the x - y plane is more natural.

It is important to note that the form of rotation matrices depends firstly on the order of their component rotations, and secondly on the linear axis conventions. The convention used here *corresponds to a z - x - y Euler angle sequence*. Therefore, it is not advisable to use the homogeneous transforms developed here until they are verified to be correct for any sensors and actuators that are used.

6.2 Frame Assignment

Several coordinate frames are important for moving vehicles. For legged vehicles, the frames embedded in the legs are assigned according to the DH convention and the frames for the rest of the system are analogous to those for wheeled vehicles. This section will present a set of frames for wheeled vehicles that occur in most applications.

These frames are indicated in the figure below:



6.2.1 The Navigation Frame

This is the coordinate system in which the vehicle position and attitude is ultimately required. Usually, this frame is taken as locally level (i.e. the z axis is perfectly aligned with the local gravity vector, not the local terrain tangent plane). The z, up, or azimuth axis is aligned with the gravity vector; the y, or north axis is aligned with the geographic pole²; and the x axis points east to complete a right-handed system. In some applications, any frame that is fixed on the earth is satisfactory whether or not it is aligned with the earth's fields. This frame is identified by the letter n.

6.2.2 The Body Frame

The body frame is positioned at the point on the vehicle body which is most convenient and is considered to be fixed in attitude with respect to the vehicle body. For Ackerman steer vehicles, the center of the rear axle is a natural place for this frame. In some applications, the best estimate of the position of the center of gravity is more appropriate. The z axis points up, y forward, and x out the right side. This frame is identified by the letter b.

6.2.3 The Positioner Frame

This frame is positioned at the point on or near any position estimation system which reports *its own position*. If the system generates attitude and attitude rates only, this frame is not required because the attitude of the device will also be that of the vehicle. For an INS, this is typically the center of the IMU and for GPS it is the phase center of the antenna³. Axes directions are defined similarly to the body frame directions. There is a different positioner frame for each positioning device. This frame is identified by the letter p.

6.2.4 The Sensor Head Frame

Sometimes, environmental perception sensors are mounted on stabilized platforms or on pan/tilt mechanisms. These provide isolation of the sensor attitude from that of the vehicle and/or the ability of the system to literally point its "head". Axes directions are defined similarly to the body frame directions. In cases where the rotary axes of the device all intersect at a point, this frame is positioned at the common point of intersection of these axes. Axes directions are defined similarly to the body frame directions.

In cases where the environmental perception sensor axes are not aligned with respect to those of the vehicle (for example, when the sensor looks downward), or in cases where there is a misalignment which must be accounted for, a ***rigid sensor head*** can be defined which tilts the body axes into coincidence with those of the sensor. This frame is identified by the letter h.

6.2.5 The Sensor Frame

This frame is positioned at the center of the environmental perception sensor with axis definitions similar to the body frame when the sensor points forward. For video cameras, it is positioned on the optical axis at the image plane behind the lens. For stereo systems, it is positioned either between both cameras or is associated with the image plane of one of them. For imaging laser

2. The geographic pole is determined by the earth's spin axis, not the magnetic field.
3. The antenna may be nowhere near the GPS receiver.

rangefinders, it is positioned as the average point of convergence of the rays through each pixel. Axes directions are defined similarly to the body frame directions. There is a different sensor frame for each sensor.

6.2.6 The Wheel Frame

This frame is positioned at the center of the wheel, on the axle. At times it may be convenient to define this frame at the wheel contact point and it may be useful to have it rotate with the wheel. Each wheel has its own frame.

6.3 The RPY Transform

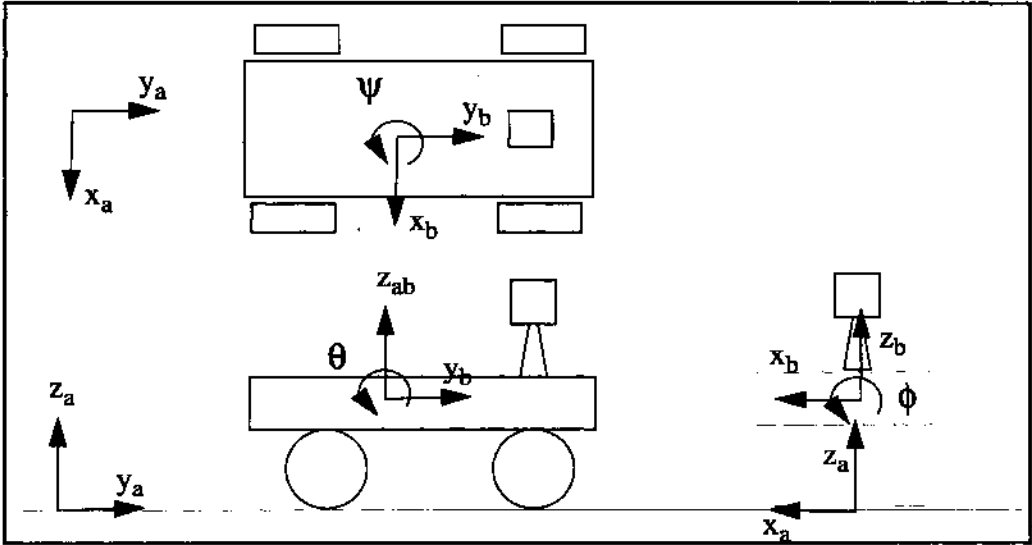
For the purpose of kinematic modelling, any of the former frames can be considered to be embedded in a rigid body. It turns out that the attitude of a rigid body can be expressed in many different ways. It is usually most convenient to express vehicle attitude in terms of three special angles called **roll**, **pitch**, and **yaw**. Luckily, most pan/tilt mechanisms are kinematically formed from a yaw rotation followed by a pitch with no roll, so they are a degenerate form of a more general transform.

These considerations imply that a general homogeneous transform, called the RPY transform, can be formed which is similar in principle to the DH matrix, except that it has three rotations, and which can serve to transform between the body frame and all others. There are six degrees of freedom involved, three translations and three rotations, and each can be either a parameter or a variable.

Let two general frames be defined as 'a' and 'b' and consider the moving axis operations which transform frame 'a' into coincidence with frame 'b'. *In order*, these are:

- translate along the (x,y,z) axes of frame 'a' by (u,v,w) until its origin coincides with that of frame 'b'
- rotate about the new z axis by an angle ψ called **yaw**
- rotate about the new x axis by an angle θ called **pitch**
- rotate about the new y axis by an angle ϕ called **roll**

Angles are measured counterclockwise positive according to the right hand rule. These operations are indicated below for the case of transforming the navigation frame into the body frame.



The forward kinematic transform that represents this sequence of operations is, according to our rules for forward kinematics:

$$T_b = \text{Trans}(u, v, w) \text{Rotz}(\psi) \text{Rotx}(\theta) \text{Roty}(\phi)$$

$$T_b^a = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_b^a = \begin{bmatrix} (c\psi c\theta c\phi - s\psi s\theta s\phi) & -s\psi c\theta c\phi - c\psi s\theta s\phi & (c\psi s\theta c\phi + s\psi c\theta s\phi) & u \\ (s\psi c\theta c\phi + c\psi s\theta s\phi) & c\psi c\theta c\phi - s\psi s\theta s\phi & (s\psi s\theta c\phi - c\psi c\theta s\phi) & v \\ -c\theta s\phi & s\theta & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- This matrix has the following two interpretations⁴:
- its columns represent the axes and origin of frame 'b' expressed in frame 'a' coordinates
 - it converts coordinates from frame 'b' to frame 'a'

The matrix can be considered to be the conversion from a pose vector of the form

$$\begin{bmatrix} x & y & z & \theta & \phi & \psi \end{bmatrix}^T$$

to a coordinate frame.

4. This transform applies to the Litton INS on the HMMWV, the Stagget stable platform, and the ROS pan/tilt head.

6.4 Frame Parameters for Wheeled Vehicles

Using the RPY transform, it is now possible to specify the parameters and variables which permit coordinate transformation from anywhere on a vehicle to anywhere else. This is accomplished by specifying the six degrees of freedom in a table. In the table, var indicates a variable and fixed indicates a fixed parameter:

Table 20: Frame Parameters for Wheeled Vehicles

Transform	u	v	w	ψ	θ	ϕ
T_b^n	var	var	var	var	var	var
T_h^b	fixed	fixed	fixed	var	var	fixed
T_s^h	fixed	fixed	fixed	fixed	fixed	fixed
T_p^b	fixed	fixed	fixed	fixed	fixed	fixed
T_w^b	fixed	fixed	fixed	fixed	var	fixed

6.5 Inverse Kinematics for the RPY Transform

The inverse kinematic solution to the RPY transform has at least two uses:

- it gives the angles to which to drive a sensor head, or a directional antenna given the direction cosines of the goal frame
- it gives the attitude of the vehicle given the body frame axes, which often correspond to the local tangent plane to the terrain over which it moves

This solution can be considered to be the procedure for extracting a pose from a coordinate frame. There are many different ways to get the solution from different elements of the RPY transform. The one used here is useful for modelling terrain following of a vehicle. Proceeding as for a mechanism, the elements of the transform are assumed to be known:

$$T_b^a = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The premultiplication set of equations will be used. The first equation is:

$$T_b^a = \text{Trans}(u, v, w) \text{Rotz}(\psi) \text{Rotx}(\theta) \text{Roty}(\phi)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (c\psi c\phi - s\psi s\theta s\phi) & -s\psi c\theta & (c\psi s\phi + s\psi s\theta c\phi) & u \\ (s\psi c\phi + c\psi s\theta s\phi) & c\psi c\theta & (s\psi s\phi - c\psi s\theta c\phi) & v \\ -c\theta s\phi & s\theta & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The translational elements are trivial. From the (1,2) and (2,2) elements:

$$\psi = \text{atan2}(r_{22}, -r_{12})$$

This implies that yaw can be determined from a vector which is known to be aligned with the body y axis. The second equation is:

$$[\text{Trans}(u, v, w)]^{-1} T_b^a = \text{Rotz}(\psi) \text{Rotx}(\theta) \text{Roty}(\phi)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (c\psi c\phi - s\psi s\theta s\phi) & -s\psi c\theta & (c\psi s\phi + s\psi s\theta c\phi) & 0 \\ (s\psi c\phi + c\psi s\theta s\phi) & c\psi c\theta & (s\psi s\phi - c\psi s\theta c\phi) & 0 \\ -c\theta s\phi & s\theta & c\theta c\phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

which generates nothing new. The next equation is:

$$[\text{Rotz}(\psi)]^{-1} [\text{Trans}(u, v, w)]^{-1} T_b^a = \text{Rotx}(\theta) \text{Roty}(\phi)$$

$$\begin{bmatrix} (r_{11}c\psi + r_{21}s\psi) & (r_{12}c\psi + r_{22}s\psi) & (r_{13}c\psi + r_{23}s\psi) & 0 \\ (-r_{11}s\psi + r_{21}c\psi) & (-r_{12}s\psi + r_{22}c\psi) & (-r_{13}s\psi + r_{23}c\psi) & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ s\theta s\phi & c\theta & -s\theta c\phi & 0 \\ -c\theta s\phi & s\theta & c\theta c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the (3,3) and (3,2) elements:

$$\theta = \text{atan2}(r_{32}, -r_{12}s\psi + r_{22}c\psi)$$

Which implies that pitch can also be determined from a vector known to be aligned with the body y axis. A good solution for ϕ is available from the (1,1) and (1,3) elements. However, for reasons of convenience, the solution will be delayed until the next equation. The next equation is:

$$\begin{aligned}
 &[Rot_x(0)]^{-1} [Rot_z(\phi)]^{-1} [Trans(u, v, w)]^{-1} T_b^* = Rot_y(\phi) \\
 &\begin{bmatrix} (r_{11}c\phi + r_{21}s\phi) & (r_{12}c\phi + r_{22}s\phi) & i(r_{13}c\psi + r_{23}s\psi) & 0 \\ c\phi [-r_{11}s\phi + r_{21}c\psi] + r_{31}s\theta & c\phi [-r_{12}s\phi + r_{22}c\psi] + r_{32}s\theta & c\phi [-r_{13}s\psi + r_{23}c\psi] + r_{33}s\theta & 0 \\ -s\phi [-r_{11}s\phi + r_{21}c\psi] + r_{31}c\theta & -s\phi [-r_{12}s\phi + r_{22}c\psi] + r_{32}c\theta & -s\phi [-r_{13}s\psi + r_{23}c\psi] + r_{33}c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

From the (1,1) and (3,1) elements:

$$\phi = \text{atan2}(s\theta [-r_{11}s\psi + r_{21}c\psi] - r_{31}c\theta, (r_{11}c\psi + r_{21}s\psi))$$

This implies that roll can be derived from a vector known to be aligned with the body x axis.

6.6 Angular Velocity

The roll, pitch, and yaw angles are, as we have defined them, measured about moving axes. Therefore, they are a sequence of **Euler angles**, specifically, the z-x-y sequence⁵. The Euler angle definition of vehicle attitude has the disadvantage that the roll, pitch, and yaw angles are not the quantities that are actually indicated by strapped-down vehicle-mounted sensors such as gyros.

The relationship between the rates of the Euler angles and the angular velocity vector is nonlinear. The angles are measured neither about the body axes nor about the navigation frame axes. It is important to know the exact relationship between the two because it provides the basis for determining vehicle attitude from angular rate measurements.

5. The sequence depends on the convention for assigning the directions of the linear axes.

In order to determine the angular velocity, consider that the total angular velocity is the sum of three components, each measured about one of the intermediate axes in the chain of rotations which bring the navigation frame into coincidence with the body frame. Using the fundamental transforms, each of the three rotation rates are transformed into the body frame by the remaining rotations in the sequence to give the result in the body frame.

$$\begin{aligned} \omega^b &= \begin{bmatrix} 0 \\ \dot{\phi} \\ 0 \end{bmatrix} + \text{rot}(y, \psi) \begin{bmatrix} \dot{\theta} \\ 0 \\ 0 \end{bmatrix} + \text{rot}(y, \psi) \text{rot}(x, \phi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ C^b &= \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} c\phi\dot{\theta} - s\phi\dot{\psi}/c\theta \\ \dot{\phi} + s\phi\dot{\psi}/c\theta \\ s\phi\dot{\theta} + c\phi\dot{\psi}/c\theta \end{bmatrix} = \begin{bmatrix} c\phi & 0 & -s\phi/c\theta \\ 0 & 1 & s\phi/c\theta \\ s\phi & 0 & c\phi/c\theta \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \end{aligned}$$

This result gives the vehicle angular velocity expressed in the body frame in terms of the Euler angle rates. Notice that when the vehicle is level, the x and y components are zero and the z component is just the yaw rate as expected.

This relationship is also very useful in its inverted form. One can verify by substitution that:

$$\begin{aligned} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} c\phi\dot{\theta} - s\phi\dot{\psi}/c\theta \\ \dot{\phi} + s\phi\dot{\psi}/c\theta \\ s\phi\dot{\theta} + c\phi\dot{\psi}/c\theta \end{bmatrix} = \begin{bmatrix} c\phi & 0 & -s\phi/c\theta \\ 0 & 1 & s\phi/c\theta \\ s\phi & 0 & c\phi/c\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \\ &\text{because } c\phi\dot{\theta} - s\phi\dot{\psi}/c\theta = \dot{\phi} \end{aligned}$$

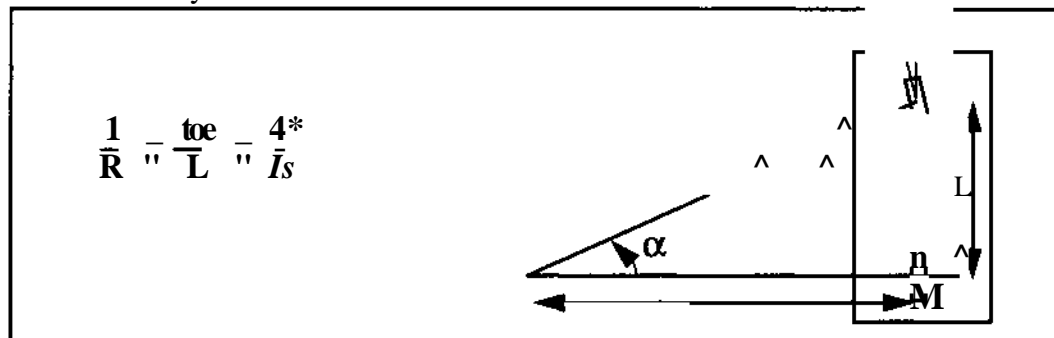
Section 7: Actuator Kinematics

For wheeled vehicles, the transformation from the angles of the steerable wheels and their velocities onto path curvatures, or equivalently angular velocities, can be very complicated. One reason for this is that there can be more degrees of freedom of steer than are necessary. In this case, the equations which relate curvature to steer angle are *overdetermined*. In one particular case, however, the steering mechanism is designed such that this will not be the case. This mechanism is used on most conventional automobiles and is called **Ackerman steering**.

TX The Bicycle Model

It is useful to approximate the kinematics of the Ackerman steering mechanism by assuming that the two front wheels turn slightly differentially so that the instantaneous center of rotation of the vehicle can be determined purely by kinematic means. This amounts to assuming that the steering mechanism is the same as that of a bicycle. Let the angular velocity vector directed along the body z axis be called p .

Using the bicycle model approximation, the path curvature K , radius of curvature R , and steer angle α are related by the wheelbase L .



Where α denotes the tangent of α . Rotation rate is obtained from the speed V as:

$$\dot{\beta} = \frac{d\beta}{ds} \frac{ds}{dt} = \kappa V = \frac{V \tan \alpha}{L}$$

Thus, the steer angle α is an indirect measurement of the ratio of β to velocity through the function:

$$\alpha = \arctan\left(\frac{L \dot{\beta}}{V}\right) = \arctan(KL)$$

Although it is common to think of these equations in kinematic terms, this is only possible when the dependence on time is avoided. In fact, this steering mechanism is modelled by a very complicated nonlinear differential equation thus:

$$\frac{d\beta(t)}{dt} = \frac{1}{L} \tan[\alpha(t)] \frac{ds}{dt} = \kappa(t) \frac{ds}{dt}$$

Chapter 2: Imaging Kinematics

A simple, general kinematic theory is presented which encompasses the kinematics transforms for all existing laser rangefinders and ideal cameras. Many sensors used on robot vehicles are of the imaging variety. For this class of sensors, the process of image formation must be modelled. Image coordinates are identified by the letter i . Typically, these transformations are not linear, and hence cannot be modelled by homogeneous transforms. This section provides the transforms necessary for modelling such sensors. These transforms include projection, reflection, and polar coordinates. These results are used extensively in the RANGER navigator and provide useful relationships for reference purposes.

Section 1: Optical Kinematics

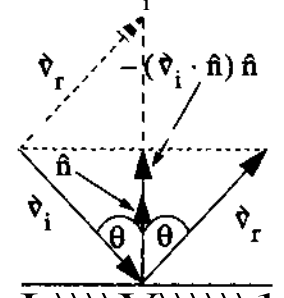
The kinematics of reflecting a laser beam from a mirror are central to the operation of the current generation of laser rangefinders. This section will outline a general technique for modelling such rangefinders.

1.1 The Reflection Operator

From Snell's law for reflection of a ray:

- The incident ray, the normal to the surface, and the reflected ray, all lie in the same plane.
- The angle of incidence equals the angle of reflection.

From these two rules, a very useful matrix operator can be formulated to reflect a vector off of any surface, given the unit normal to the surface. Consider a vector \mathbf{v}_i , not necessarily a unit vector, which impinges on a reflecting surface at a point where the unit normal to the surface is $\mathbf{\hat{n}}$. Unless they are parallel, the incident and normal vector define a plane, which will be called the **reflection plane**. Drawing both vectors in this plane, it is clear that Snell's law can be written in many forms:



$$\mathbf{v}_r = \mathbf{v}_i - 2(\mathbf{v}_i \cdot \mathbf{\hat{n}})\mathbf{\hat{n}}$$

$$\mathbf{v}_r = \mathbf{v}_i - 2\cos\theta\mathbf{\hat{n}}$$

$$\mathbf{v}_r = \mathbf{v}_i - 2(\mathbf{v}_i \otimes \mathbf{\hat{n}})\mathbf{\hat{n}}$$

$$\mathbf{v}_r = \text{Ref}(\mathbf{\hat{n}})\mathbf{v}_i$$

$$\text{Ref}(\mathbf{v}_i) \Rightarrow \mathbf{I} - 2(\mathbf{\hat{n}} \otimes \mathbf{\hat{n}})$$

$$\text{Ref}(\mathbf{v}_i) = \begin{bmatrix} 1 - 2n_x^2 & -2n_x n_y & -2n_x n_z \\ -2n_y n_x & 1 - 2n_y^2 & -2n_y n_z \\ -2n_z n_x & -2n_z n_y & 1 - 2n_z^2 \end{bmatrix}$$

Where the outer product (\otimes) of the normal with itself was used in forming the matrix equivalent.

The result is expressed in the same coordinates in which both the normal and the incident ray were expressed. This can be used to model the "optical" kinematics of laser rangefinders. Notice that a reflection is equivalent to a rotation of twice the angle of incidence about the normal to the reflection plane. A similar matrix refraction operator can be defined.

In order to model rangefinders, the laser beam will be modelled by a unit vector since the length of the beam is immaterial. The unit vector is operated upon by the reflection operator - one reflection for each mirror. The ultimate result of all reflections will be expressed in the original coordinate system.

The basic idea is that, since the mirrors are flat, the reorientation of the beam is independent of exactly where it hits the mirror. This is justified because Snell's law states that the direction of any reflected beam depends only on the direction of the normal to any mirror and the direction of the incident beam.

The results of such an analysis give the orientation of the laser beam as a function of the actuated mirror angles, but it says nothing about where the beam is positioned in space. The precise position of the beam is not difficult to calculate and is important in the sizing of mirrors. From the point of view of computing kinematics, beam position is irrelevant, so it will be ignored here.

1.2 The Vector Projection Operator

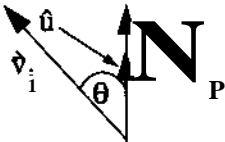
The projection of one vector onto another is trivially given by the dot product. Consider a vector \hat{v}_i , not necessarily a unit vector, which is to be projected onto a unit vector \hat{n} . The projection rule can be written in many forms:

$$\lambda_P = (\hat{V}^n)^u$$

$$\hat{v}_p = \hat{v}_i \cos \theta \hat{n}$$

$$\hat{v}_p = \text{Projv}(\hat{n}) \hat{v}_i$$

$$\text{Projv}(\hat{n}) = \hat{v}_i \otimes \hat{n}$$



$$\text{Projv}(\hat{n}) = \begin{bmatrix} n_x n_x & n_x n_y & n_x n_z \\ n_y n_x & n_y n_y & n_y n_z \\ n_z n_x & n_z n_y & n_z n_z \end{bmatrix}$$

1.3 The Plane Projection Operator

The projection of a vector into a plane is a simple matter. The basic operation is to remove the component of the vector along the normal to the plane. Consider a vector \hat{v} , not necessarily a unit vector, which is to be projected on a surface at a point where the unit normal to the surface is \hat{n} . This plane will be called the **projection plane**. The operator can be written in many ways:

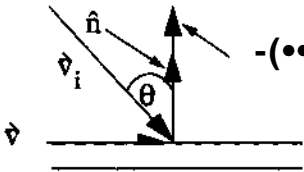
$$\hat{v}_p = \hat{v} - (\hat{v} \cdot \hat{n}) \hat{n}$$

$$\hat{v}_p = \hat{v} - \cos \theta \hat{n}$$

$$\hat{v}_p = \hat{n} \times (\hat{n} \times \hat{v})$$

$$\hat{v}_p = \text{Projp}(\hat{n}) \hat{v}$$

$$\text{Projp}(\hat{n}) = I - (\hat{n} \otimes \hat{n})$$



$$\text{Projp}(\hat{n}) = \begin{bmatrix} 1 - n_x^2 & -n_x n_y & -n_x n_z \\ -n_y n_x & 1 - n_y^2 & -n_y n_z \\ -n_z n_x & -n_z n_y & 1 - n_z^2 \end{bmatrix}$$

The expression involving the vector cross product was generated by noting that \hat{p} is oriented along the vector $\hat{n} \times \hat{v}_i$ and has magnitude $v_i \sin \theta$. The result is expressed in the same coordinates in which both the normal and the incident ray were expressed. It is useful for determining the kinematics of parallel laser beams when they are transformed by the scanning mechanism.

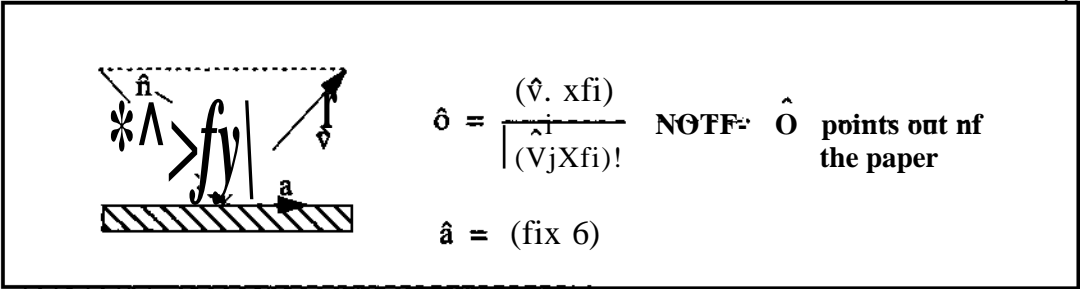
1.4 Rate Relationships

Consider now the problem of predicting the angular velocity of the reflected ray when the incident ray is reflected off a mirror that is rotating. This is, after all, exactly how a scanning mechanism works. Let the mirror rotate about a general axis in space. Its angular velocity is denoted by $\dot{\phi}_n$. This rotation of the mirror gives rise to a rotation $\dot{\phi}_r$ of the reflected beam. In the completely general case, the incident beam may rotate as well with velocity $\dot{\phi}_i$. However, since it is clear that the angular velocity of the normal with respect to the incident beam is the quantity that matters, $\dot{\phi}_i$ can be considered to be fixed with the understanding that its angular velocity must be added to that of the normal when it is not fixed.

This problem can be solved using concepts of moving reference frames. Consider an observer fixed in the reflection plane. The observer is not fixed in the frame of the mirror because the reflection plane may rotate about the mirror normal. Let the angular velocity of the moving observer in the frame fixed to the sensor housing be $\dot{\phi}_5$. Angular velocities are true vectors which add in the usual way. Hence, the angular velocity of the output beam in the fixed frame is just the sum of what the moving observer sees and the motion of the moving observer:

$$\dot{\phi}_r = \dot{\phi}_r|_{\text{moving}} + \dot{\phi}$$

Establish a coordinate system fixed to the projection plane consisting of the mirror normal \hat{n} , the normal to the projection plane \hat{a} , and the cross product of these two \hat{o} . This system is shown in the following figure:



Now, from the point of the moving observer, the motion of the reflected beam is given by the rate of change of θ and, as a vector, it is directed normal to the reflection plane:

$$\dot{\phi}_r|_{\text{moving}} = \dot{\theta} \hat{o}$$

Note there is no factor of two in the moving frame since the mirror normal vector is fixed in the moving frame. Now \hat{o} is the projection of the angular velocity of \hat{n} onto the normal to the reflection plane:

$$\dot{\phi}_r|_{\text{moving}} = [\dot{\phi}_n \cdot \hat{o}] \hat{o}$$

The angular velocity of the reflection plane is equal to that of the mirror minus the component which is oriented along the mirror normal, since rotation around the mirror normal does not rotate the projection plane. This is, of course, the projection into the plane of the mirror itself:

$$\vec{\omega} = \vec{\omega}_n - (\vec{\omega}_n \cdot \hat{n}) \hat{n}$$

Hence, the complete result is:

$$\vec{\omega}_r = [\vec{\omega}_n \cdot \hat{\delta}] \hat{\delta} + \vec{\omega}_n - (\vec{\omega}_n \cdot \hat{n}) \hat{n}$$

This result is intuitively correct because it contains two times the $\hat{\delta}$ component of $\vec{\omega}_n$, one times the $\hat{\delta}$ component of $\vec{\omega}_r$ and zero times the \hat{n} component of $\vec{\omega}_n$. This can be written in terms of the operators given so far as follows:

$$\vec{\omega}_r = [2\text{Proj}_v(\hat{\delta}) + \text{Proj}_v(\hat{n})] \vec{\omega}_n$$

In words, *the angular velocity of the reflected beam is twice the projection of the mirror angular velocity onto the reflection plane normal plus the projection onto the axis normal to both the reflection plane normal and the mirror normal¹.*

1.5 Mirror Gain

Define the **mirror gain** K_M in terms of the angular velocity of the beam $\vec{\omega}_r$ and the angular velocity of the mirror $\vec{\omega}_n$ as follows:

$$K_M = \frac{\omega_r}{\omega_n}$$

The mirror gain can vary from 0 to 2 depending on the orientation of the input beam, the mirror normal, and the mirror angular velocity. The above result explains why the beam enters the polygonal mirror from the side in the ERIM scanner - because there is a gain of 2 in only that direction. In the Hurricane scanner, the beam enters along $\hat{\delta}$ axis, so the gain is 1. Note also, that a two-axis scanner requires two separate mirrors in order to get a gain of 2 in both axes.

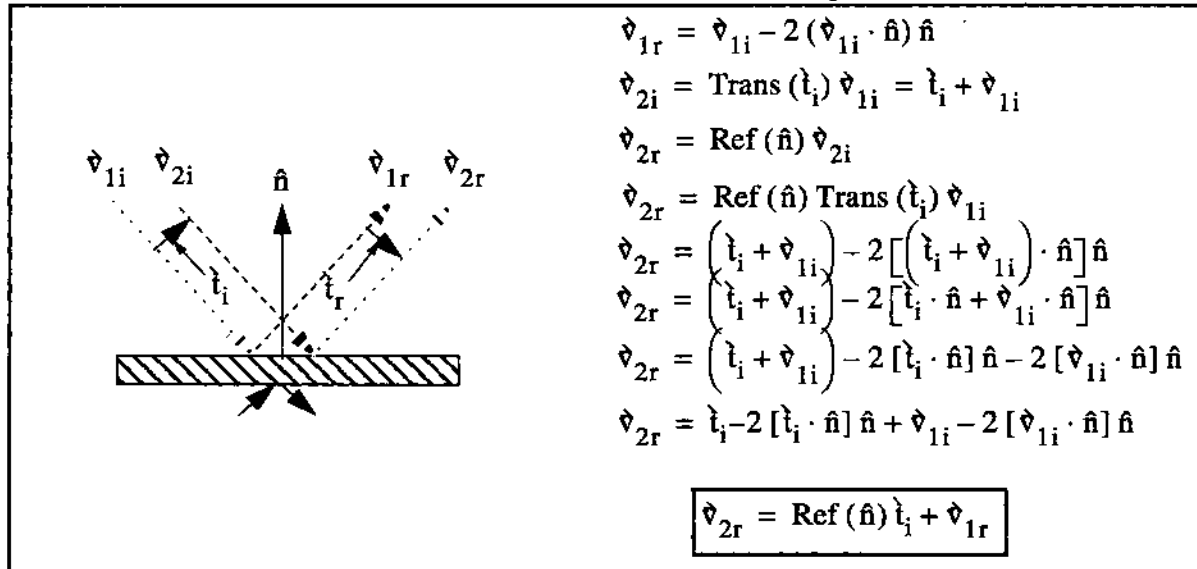
This gain can be a very significant factor in the design of a rangefinder since the angular velocity of mirrors is a major limiting factor. *Throughput can be doubled by appropriate scanning mechanism design.*

1. See Optics by Hecht for a verification of this surprising result.

1.6 Composition of Translation and Reflection

Consider now the issues involved with using two parallel laser transmitters. Peculiar things can happen when two parallel beams enter a system of mirrors. In the ERIM scanner for instance, if two beams enter the mirrors spaced along the z axis, they leave spaced along the x axis, if they enter spaced along the y axis, they leave spaced along the z axis, and finally if they enter spaced along the x axis, they are coincident, and can be considered to leave spaced along the y axis.

This behavior is explained in the following figure. Consider two parallel incident beams $\hat{\mathbf{v}}_{1i}$ and $\hat{\mathbf{v}}_{2i}$ which are separated by a perpendicular translation vector $\hat{\mathbf{t}}_i$. The perpendicular translation is used since beams are themselves lines and otherwise have no unique relative translation.

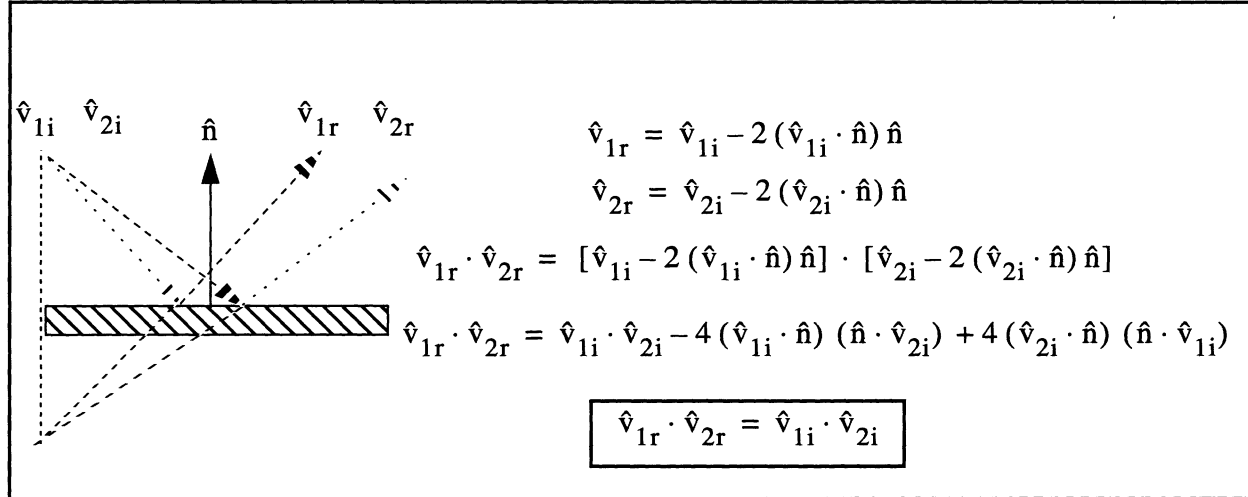


Many conclusions can be drawn from this result:

- The translation vector itself is reflected as if it was a beam entering through the back of the mirror.
- Since the reflection operator preserves the length of the incident vector, the relative perpendicular translation of the two beams is preserved as well. Note however that while the *perpendicular* distance is preserved, the *projection* of the distance between the two beams onto any surface changes.
- If a component of the translation vector exists out of the reflection plane, it is left unchanged, so the above expression is completely general.
- As a convenient mnemonic, whenever the translation vector has a component along the mirror normal, that component is flipped by the reflection operation, and the two beams switch places along the normal direction.

1.7 Composition of Rotation and Reflection

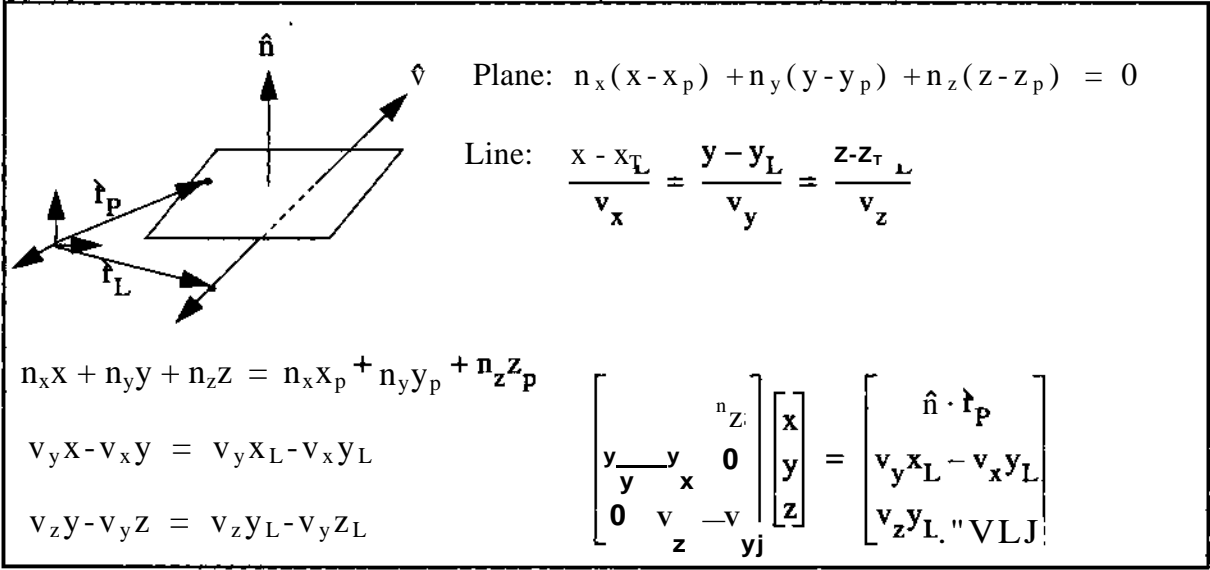
A basic property of plane mirrors is the formation of an image *behind* the mirror. This property arises from the conservation of the relative angles between rays emanating from a point source. This behavior is explained in the following figure. Consider two incident beams \hat{v}_{1i} and \hat{v}_{2i} which are separated by an angle θ . Since both beams are unit vectors, their dot product is a direct measure of the angle between them.



Thus the angle between the reflected beams is equal to the angle between the incident beams. This means that a point source reflected by a mirror forms an image behind the mirror. The laser transmitter of a rangefinder can be modelled as a point source. The implication is that *the projection of the laser spot on the groundplane is just the intersection of a cone surrounding the beam and the groundplane.*

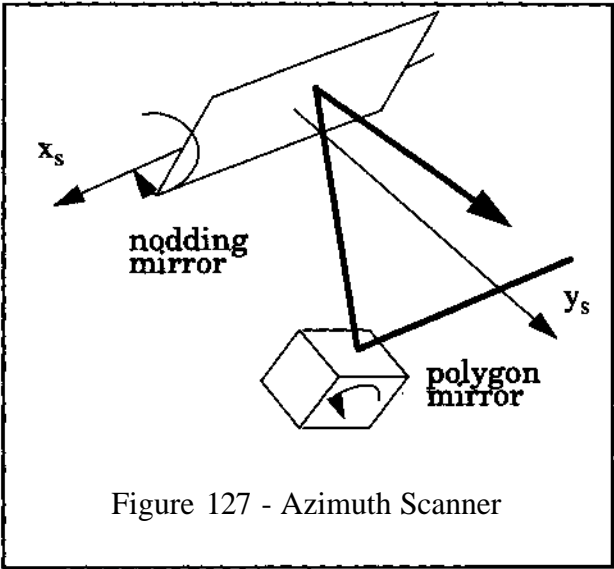
1.8 Intersection of a Line and a Plane

The intersection of a line and a plane is the basis for a complete ray tracing simulation of the internal kinematics of rangefinders. This permits computation of the precise spot pattern of the rangefinder, and the required size of the mirrors. Let a line be defined by a vector \hat{v} to which it is parallel and a point K_L which lies on the line. Let a plane be defined by a vector \hat{n} normal to the plane and a point \mathcal{E}_p which lies on the plane. The general solution to the problem is given in the following figure by a matrix equation.



Section 2: Kinematics of the Azimuth Scanner

The **azimuth scanner** is a generic name for a class of laser rangefinders with equivalent kinematics. In this scanner, the laser beam undergoes the azimuth rotation/reflection first and the elevation rotation/reflection second. Both the ERIM scanner and the Perceptron fall into this category. Both scanners are 2D scanning laser rangefinders employing a "polygonal" azimuth mirror and a flat "nodding" elevation mirror. The mirrors move as shown below:

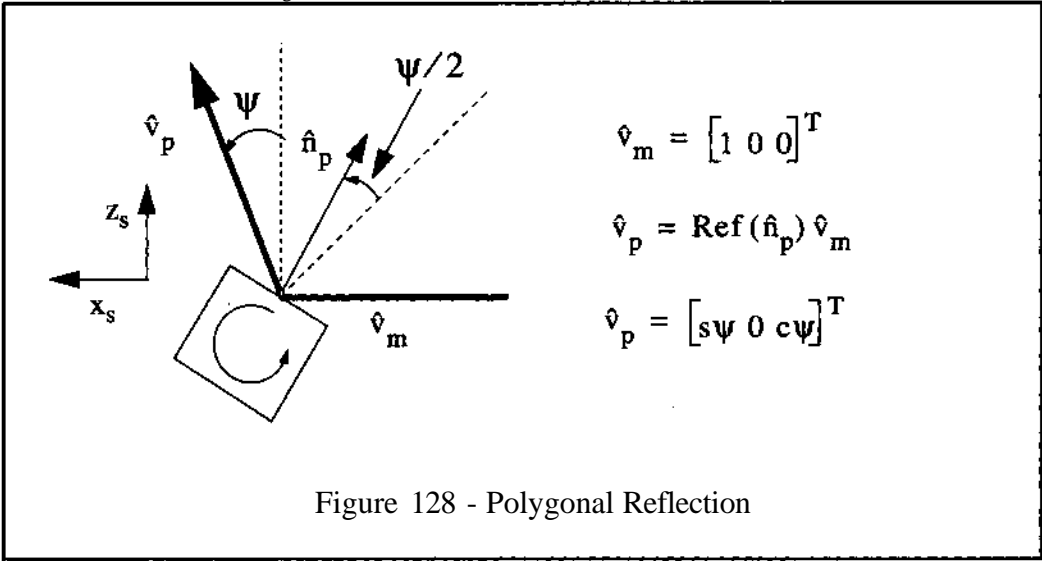


2.1 Forward Kinematics

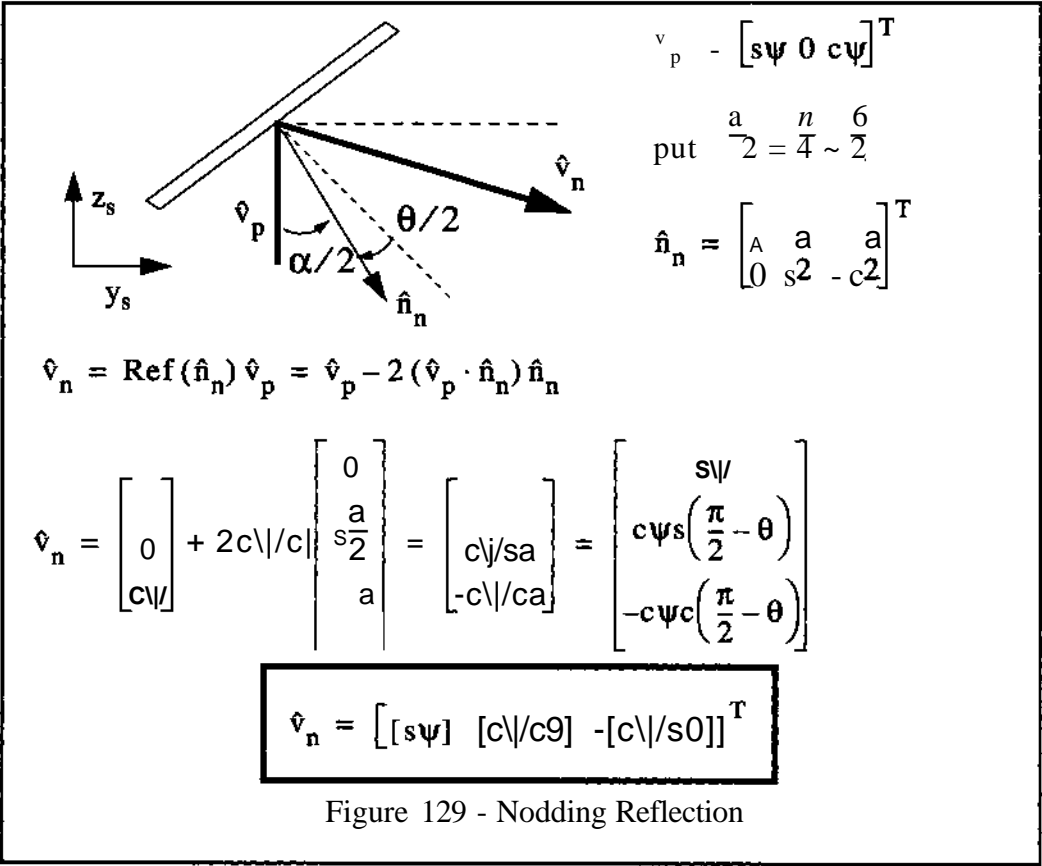
A coordinate system called the "s" system is fixed to the sensor with y pointing out the front of the sensor and x pointing out the right side. The beam enters along the x_s axis. It reflects off the polygonal mirror which rotates about the y_s axis. It then reflects off the nodding mirror, to leave the housing roughly aligned with the y_c axis.

First, the beam is reflected from the laser diode about the normal to the polygonal mirror. Computation of the output of the polygonal mirror can be done by inspection - noting that the beam rotates by twice the angle of the mirror because it is a reflection operation. The z-x plane contains

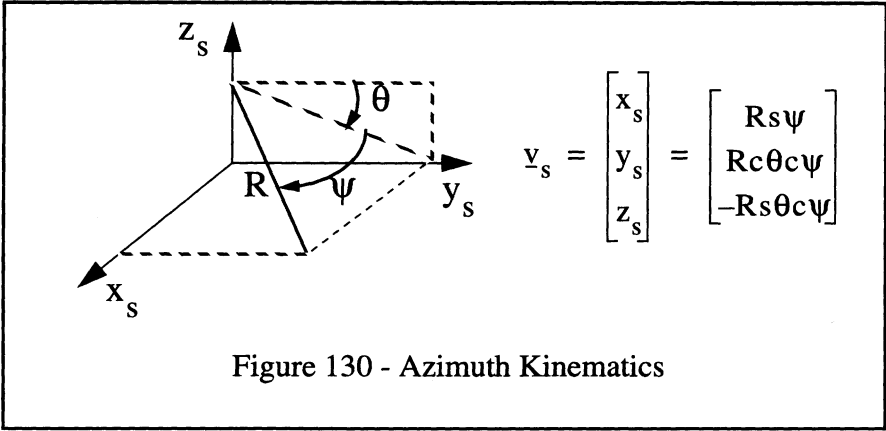
both the incident and normal vectors. The datum position of the mirror should correspond to a perfectly vertical beam, so the datum for the mirror rotation angle is chosen appropriately. Consider an input beam \hat{v}_m along the x_g axis and reflect it about the mirror by inspection:



Notice that this vector is contained within the x_s - z_s plane. Now this result must be reflected about the nodding mirror. Notice that, at this point, \hat{v}_p cannot be simply rotated around the x axis since the axis of rotation which is equivalent to a reflection is normal to both \hat{v}_p and \hat{n}_n . Since \hat{v}_p is not always in the y_s - z_s plane, the x_s axis is not always the axis of rotation. Which is the required result.



This result is summarized in the following figure:



Thus, the kinematics of the azimuth scanner are equivalent to a rotation around the x_s axis followed by a rotation around the *new* z_s axis. By a theorem of 3D rotations, this is also equivalent to two rotations in the opposite order about fixed axes.

2.2 Forward Imaging Jacobian

The imaging Jacobian provides the relationship between the differential quantities in the sensor frame and the associated position change in the image. The Jacobian is:

$$\underline{v}_i = \begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} \quad \underline{v}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} R s \psi \\ R c \theta c \psi \\ -R s \theta c \psi \end{bmatrix}$$

$$J_i^s = \frac{\partial \underline{v}^s}{\partial \underline{v}^i} = \begin{bmatrix} \frac{\partial x_s}{\partial R} & \frac{\partial x_s}{\partial \psi} & \frac{\partial x_s}{\partial \theta} \\ \frac{\partial y_s}{\partial R} & \frac{\partial y_s}{\partial \psi} & \frac{\partial y_s}{\partial \theta} \\ \frac{\partial z_s}{\partial R} & \frac{\partial z_s}{\partial \psi} & \frac{\partial z_s}{\partial \theta} \end{bmatrix} = \begin{bmatrix} s \psi & R x \psi & 0 \\ c \theta c \psi & -R c \theta s \psi & -R s \theta c \psi \\ -s \theta c \psi & R s \theta s \psi & -R c \theta c \psi \end{bmatrix}$$

2.2 Projection Table

The directions of the rays through each pixel of an imaging sensor are fixed with respect to the sensor frame. The trigonometric overhead of computing this information can be severe in some applications, so it is possible and worthwhile to compute it and store it in tables. These tables can be useful both in real-time perception applications and in off-line ray tracing simulation. This section provides the equations necessary to map (row,column) coordinates onto the direction cosines of the ray through those coordinates.

For the azimuth scanner, the unit vector is simply:

$$\hat{u} = [\bar{s} \quad c0c \quad -s0c \quad \bar{j}]$$

In this case, the pixel coordinates are directly proportional to the angles so that:

$$\psi = \frac{(\text{col} - \text{cols}/2)}{\text{cols}} \text{HFOV}$$
$$\theta = \frac{(\text{row} - \text{rows}/2)}{\text{rows}} \text{VFOV}$$

2.4 Inverse Kinematics

The forward transform is easily inverted.

$$\begin{bmatrix} R \\ \psi \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ atan[x/y_s + z_s^*j] \\ atan (-z_s/y_s) \end{bmatrix} = h(x_s, y_s, z_s)$$

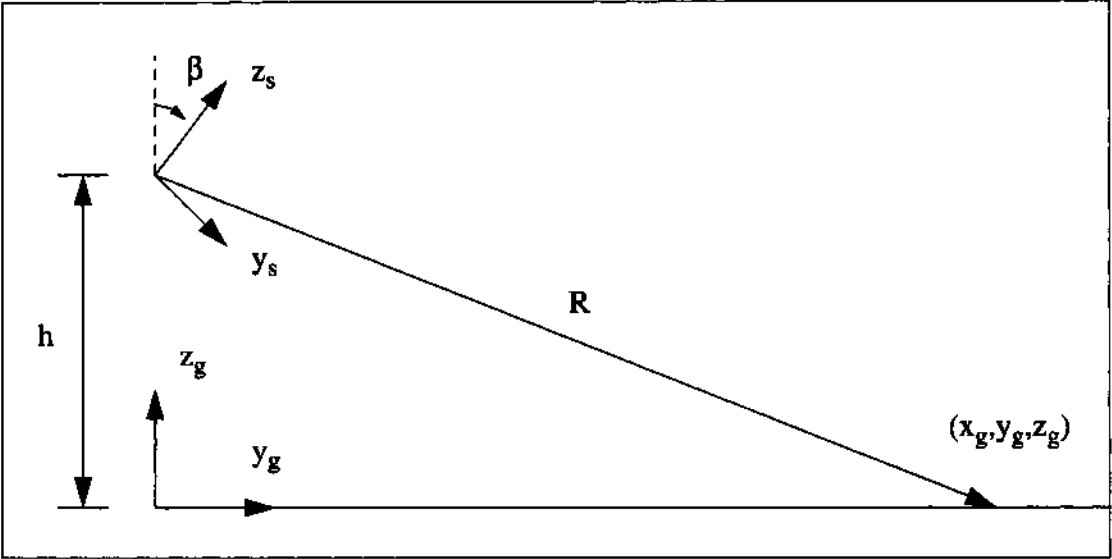
2.5 Inverse Imaging Jacobian

The imaging Jacobian provides the relationship between the differential quantities in the sensor frame and the associated position change in the image. The Jacobian is:

$$\begin{aligned} \mathbf{v}_1 = \begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} &= \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ atan\left(x_s/\left(\sqrt{y_s^2 + z_s^2}\right)\right) \\ atan (-z_s/y_s) \end{bmatrix} \qquad \mathbf{v}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \\ J_s^i = \frac{\partial \mathbf{v}^i}{\partial \mathbf{v}^s} &= \begin{bmatrix} \frac{\partial R}{\partial x_s} & \frac{\partial R}{\partial y_s} & \frac{\partial R}{\partial z_s} \\ \frac{\partial \psi}{\partial x_s} & \frac{\partial \psi}{\partial y_s} & \frac{\partial \psi}{\partial z_s} \\ \frac{\partial \theta}{\partial x_s} & \frac{\partial \theta}{\partial y_s} & \frac{\partial \theta}{\partial z_s} \end{bmatrix} = \begin{bmatrix} \frac{x_s}{R} & \frac{y_s}{R} & \frac{z_s}{R} \\ \frac{\sqrt{y_s^2 + z_s^2}}{R^2} \frac{y_s}{\sqrt{y_s^2 + z_s^2}} \left(\frac{-x_s}{\sqrt{y_s^2 + z_s^2}}\right) & \frac{z_s}{R^2} \left(\frac{-x_s}{\sqrt{y_s^2 + z_s^2}}\right) & \frac{z_s}{R^2} \left(\frac{-y_s}{\sqrt{y_s^2 + z_s^2}}\right) \\ 0 & \frac{z_s}{y_s^2 + z_s^2} & \frac{-y_s}{y_s^2 + z_s^2} \end{bmatrix} \end{aligned}$$

2.6 Analytic Range Image of Flat Terrain

Given the basic kinematic transform, many analyses can be performed. The first is to compute an analytic expression for the range image of a perfectly flat piece of terrain. Let the sensor fixed "s"



coordinate system be mounted at a height h and tilted forward by a tilt angle of β . Then, the transform from sensor coordinates to global coordinates is:

$$\begin{aligned} x_g &= x_s \\ y_g &= y_s \cos \beta + z_s \sin \beta \\ z_g &= -y_s \sin \beta + z_s \cos \beta + h \end{aligned}$$

If the kinematics are substituted into this, the transform from the polar sensor coordinates to global coordinates is obtained:

$$\begin{aligned} x_g &= R \sin \theta \cos \psi \\ y_g &= (R \cos \theta \cos \psi) \cos \beta - (R \sin \theta \sin \psi) \sin \beta = R \cos \theta \cos \psi \cos \beta - R \sin \theta \sin \psi \sin \beta \\ z_g &= (-R \cos \theta \sin \psi) \sin \beta - (R \sin \theta \cos \psi) \cos \beta + h = h - R \sin \theta \cos \psi \cos \beta - R \cos \theta \sin \psi \sin \beta \end{aligned}$$

Now by setting $z_g = 0$ and solving for R , the expression for R as a function of the beam angles θ and ψ for flat terrain is obtained. This is an analytic expression for the range image of flat terrain under the azimuth transform.

$$R = h / (\cos \theta \cos \psi \cos \beta + \sin \theta \sin \psi \sin \beta)$$

Notice that when R is large $s_{Gp} = h/R$. This is the perception ratio defined earlier. As a check on the range image formula, the resulting range image is shown below for $h = 2.5$, $\beta = 16.5^\circ$, a HFOV of 140° , a VFOV of 30° , and an IFOV of 5 mrad. It has 490 columns and 105 rows. The edges correspond to contours of constant range of 20 meters, 40 meters, 60 meters, etc.

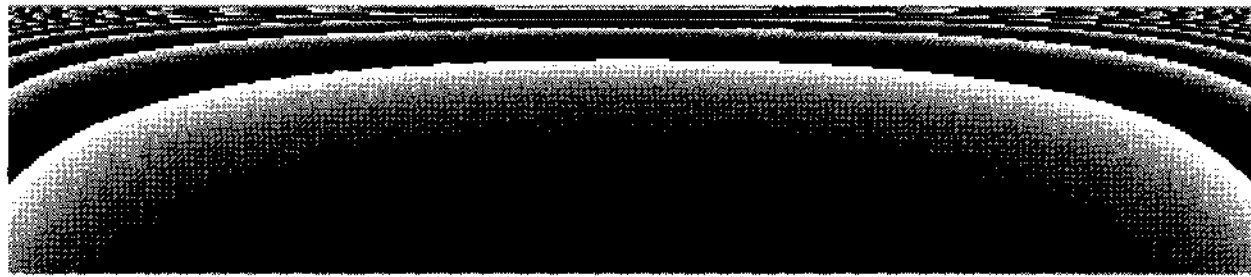


Figure 131 - Azimuth Range Image

The curvature of the contours of range is intrinsic to the sensor kinematics and is independent of the sensor tilt. Substituting this back into the coordinate transform, the coordinates where each ray intersects the groundplane are:

$$\begin{aligned}x_g &= ht\psi/s\theta\beta \\y_g &= h/t\theta\beta \\z_g &= 0\end{aligned}$$

Notice that the y coordinate is independent of ψ and hence, lines of constant elevation in the image are *straight lines along the y -axis on flat terrain*.

2.7 Field of View

Of basic interest is the region on the groundplane illuminated by the sensor. This can be computed in closed form as follows. From the previous result, it can be verified by substitution and some algebra that:

$$\left[\frac{x_g}{t\psi}\right]^2 - y_g^2 = h^2$$

Thus lines of constant azimuth are *hyperbolas* on the groundplane.

Let the two extreme columns of the image be given by the polar azimuth angles of $\pm F$. Then, for the two extreme columns of the image given by $\psi = \pm F$, the asymptotes of the hyperbola are of the form:

$$\frac{x_g}{t\psi} = \pm y_g$$

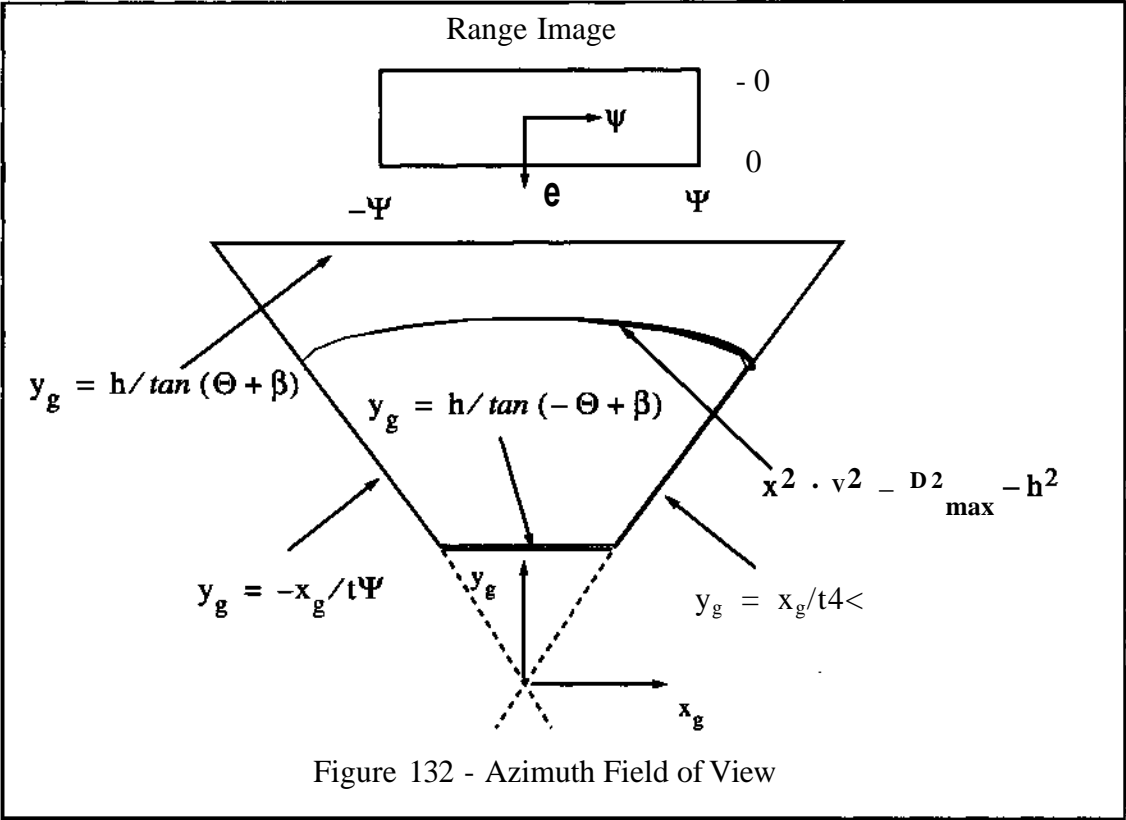
Let the two extreme rows of the image be given by the polar elevation angles of $\pm \theta$. Then the proximal and distal lines of the field of view are given by:

$$y_g = h/\tan(\pm \theta + \beta)$$

However, for most sensor tilt angles, the distal bound is far outside the sensor maximum range. Hence, the true limit of data is given by the intersection of a sphere centered at the sensor and the groundplane. By inspection, this sphere is given by:

$$x^2 + y^2 + h^2 = R_{\max}^2$$

The shape of the field of view of a single image is given below:



2.8 Resolution

The Jacobian of the groundplane transform has many uses. Most important of all, it provides a measure of sensor resolution on the ground plane. Differentiating the previous result:

$$\begin{bmatrix} dx_g \\ dy_g \end{bmatrix} = \begin{bmatrix} \frac{h (\sec \psi)^2}{s \theta \beta} & \frac{-h \tan \psi \csc \theta \beta}{(s \theta \beta)^2} \\ 0 & \frac{h}{(s \theta \beta)^2} \end{bmatrix} \times \begin{bmatrix} d\psi \\ d\theta \end{bmatrix}$$

The determinant of the Jacobian relates differential areas:

$$dx_g dy_g = \left[\frac{(h \sec \psi)^2}{(s \theta \beta)^3} \right] d\psi d\theta$$

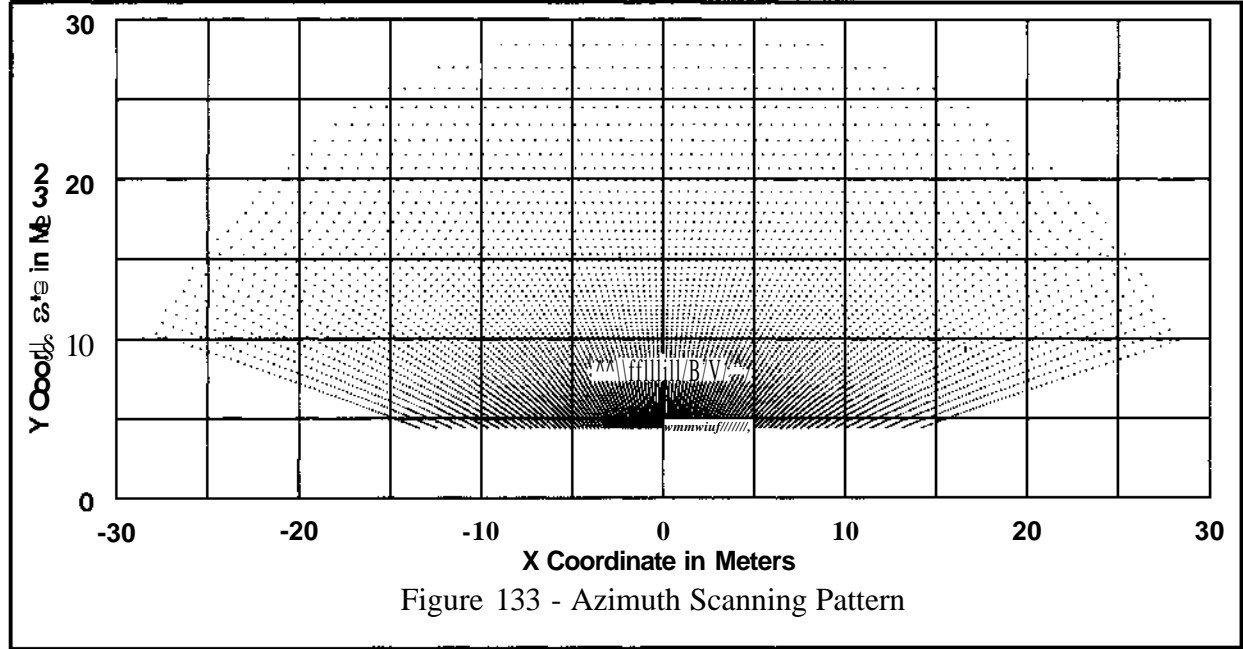
Notice that when R is large and $\psi = 0$, the Jacobian norm can be approximated by:

$$\left[\frac{(h \sec \psi)^2}{(s\theta\beta)^3} \right] \approx h^2 / \left(\frac{h}{R} \right)^3 = R^2 / \left(\frac{h}{R} \right)$$

which is the same result given earlier. The scanning density is proportional to the cube of the range.

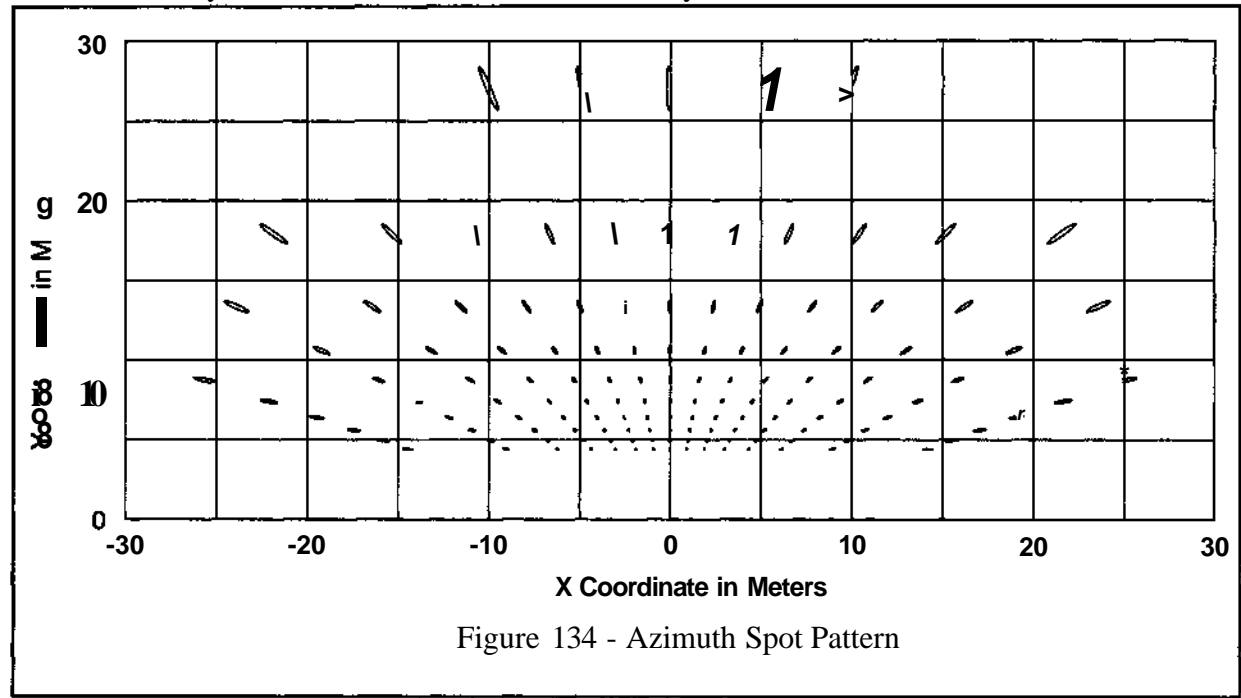
2.9 Azimuth Scanning Pattern

The scanning pattern is shown in the following figure with a 10 m grid superimposed for reference purposes. Only every fifth pixel is shown in azimuth to avoid clutter.



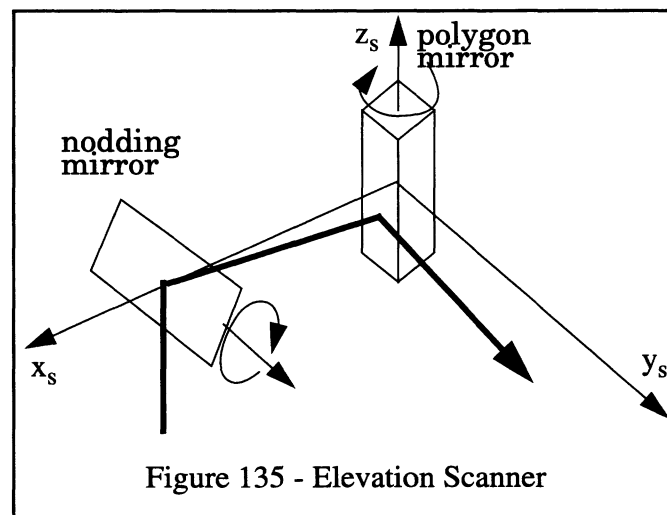
2.10 Azimuth Spot Pattern

The spot pattern is obtained by intersecting the ground plane with the beam. Spots are shown at one-tenth density in elevation and one-seventh density in azimuth:



Section 3: Kinematics of the Elevation Scanner

The **elevation scanner** is a generic name for a class of laser rangefinders with equivalent kinematics. In this scanner, the laser beam undergoes the elevation rotation/reflection first and the azimuth rotation/reflection second. It is the *kinematic dual* of the azimuth scanner. The Hurricane scanner falls into this category. While the Hurricane uses a single mirror actuated in two directions, equivalent kinematics can be generated in a dual mirror system in such a manner as to reduce by half the rotational speed of the mirrors. Consider a representative elevation scanner which is constructed by reconfiguring the ERIM scanner. Like the ERIM, this scanner is a 2D scanning laser rangefinder employing a “polygonal” mirror and a flat “nodding” mirror. Compared to the ERIM, the roles of the mirrors are reversed, and the order in which the beam encounters them is reversed. The mirrors move as shown below:

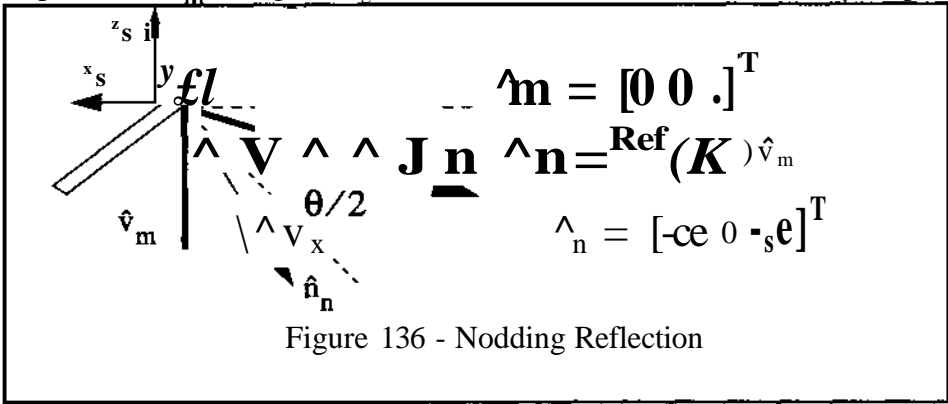


3.1 Forward Kinematics

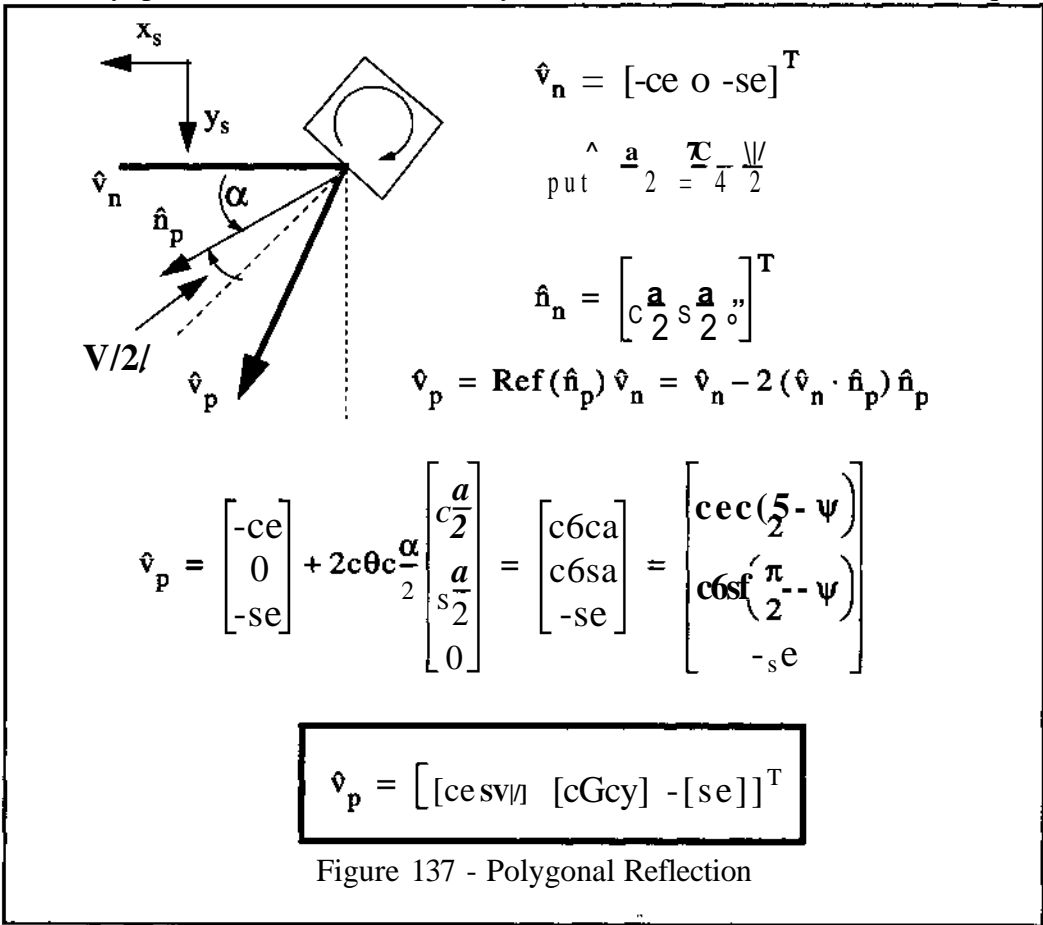
A coordinate system called the “s” system is fixed to the sensor with y pointing out the front of the sensor and x pointing out the right side. The beam enters along the z_s axis. It reflects off the nodding mirror which rotates about the y_s axis. It then reflects off the polygonal mirror, which rotates about the z_s axis, to leave the housing roughly aligned with the y_s axis.

First, the beam is reflected from the laser diode about the normal to the nodding mirror. Computation of the output of the nodding mirror can be done by inspection - noting that the beam rotates by twice the angle of the mirror because it is a reflection operation. The z - x plane contains

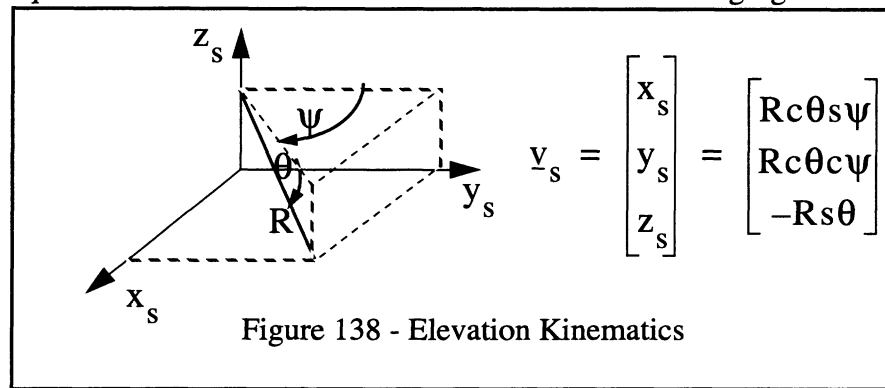
both the incident and normal vectors. The datum position of the mirror must correspond to a perfectly horizontal output beam, so the datum for the mirror rotation angle is chosen appropriately. Consider an input beam \hat{v}_m along the z_g axis and reflect it about the mirror by inspection:



Notice that this vector is contained within the x_s - z_s plane. Now this result must be reflected about the polygonal mirror. Notice that, at this point, \hat{v}_n cannot simply be rotated around the z axis since the axis of rotation which is equivalent to a reflection is normal to both \hat{v}_n and \hat{n}_p . Since \hat{v}_n is not always in the x_s - y_s plane, the z_s axis is not always the axis of rotation. Which is the required result.



Which is the required result. This result is summarized in the following figure:



In comparison to the azimuth scanner, the kinematics of the elevation scanner are equivalent to the same rotations taken in the opposite order. First, a rotation around the z_s axis followed by a rotation around the *new* x_s axis. By a theorem of 3D rotations, this is also equivalent to two rotations in the opposite order about fixed axes².

3.2 Forward Imaging Jacobian

The imaging Jacobian provides the relationship between the differential quantities in the sensor frame and the associated position change in the image. The Jacobian is:

$$\underline{v}_i = \begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} \quad \underline{v}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} Rc\theta s\psi \\ Rc\theta c\psi \\ -Rs\theta \end{bmatrix}$$

$$J_i^s = \frac{\partial \underline{v}^s}{\partial \underline{v}^i} = \begin{bmatrix} \frac{\partial x_s}{\partial R} & \frac{\partial x_s}{\partial \psi} & \frac{\partial x_s}{\partial \theta} \\ \frac{\partial y_s}{\partial R} & \frac{\partial y_s}{\partial \psi} & \frac{\partial y_s}{\partial \theta} \\ \frac{\partial z_s}{\partial R} & \frac{\partial z_s}{\partial \psi} & \frac{\partial z_s}{\partial \theta} \end{bmatrix} = \begin{bmatrix} c\theta s\psi & Rc\theta c\psi & -Rs\theta s\psi \\ c\theta c\psi & -Rc\theta s\psi & -Rs\theta c\psi \\ s\theta & 0 & -Rc\theta \end{bmatrix}$$

2. It is clear that there are only two possibilities for the order in which two rotations can take place about two orthogonal axes. Hence, there are **only two canonical forms** that the 2D laser rangefinder can take.

3.3 Projection Table

The directions of the rays through each pixel of an imaging sensor are fixed with respect to the sensor frame. The trigonometric overhead of computing this information can be severe in some applications, so it is possible and worthwhile to compute it and store it in tables. These tables can be useful both in real-time perception applications and in off-line ray tracing simulation. This section provides the equations necessary to map (row,column) coordinates onto the direction cosines of the ray through those coordinates.

For the elevation scanner, the unit vector is simply:

$$\mathbf{f}^i = [c6s\backslash f\ c9c\backslash / \ -so]$$

In this case, the pixel coordinates are directly proportional to the angles so that:

$$\psi = \frac{(col - cols/2)}{cols} HFOV$$
$$\theta = \frac{(row - rows/2)}{rows} VFOV$$

3.4 Inverse Kinematics

This forward transform is easily inverted.

$$\begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ \text{atan}(x/y_s) \\ \text{atan}\left(\frac{z_s}{\sqrt{x_s^2 + y_s^2}}\right) \end{bmatrix} = {}^h({}^x_s, {}^y_s, {}^z_s)$$

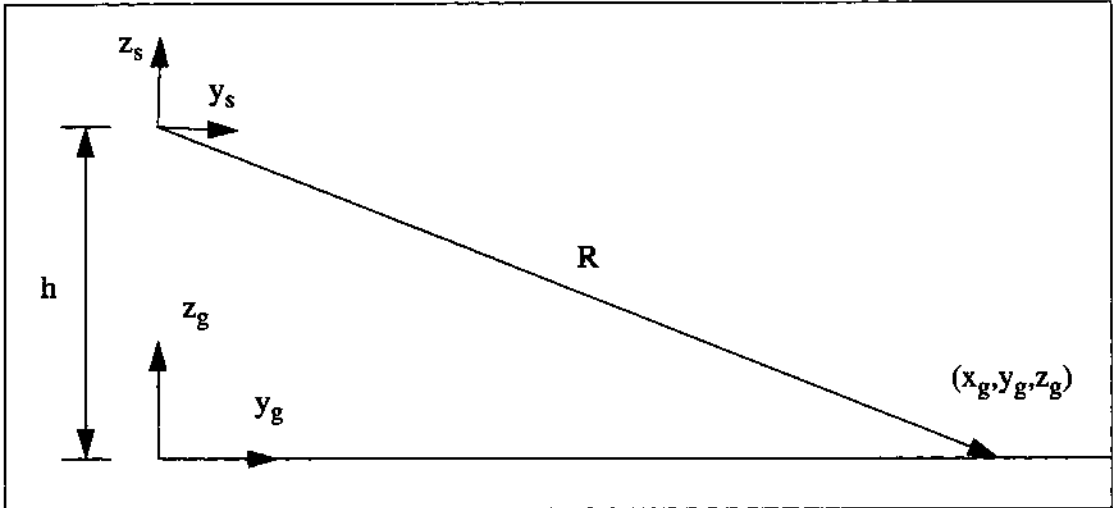
3.5 Inverse Imaging Jacobian

The imaging Jacobian provides the relationship between the differential quantities in the sensor frame and the associated position change in the image. The Jacobian is:

$$\begin{aligned} \mathbf{v}_1 = \begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} &= \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ \text{atan}(x/y_s) \\ \text{atan}\left(-z_s/\sqrt{x_s^2 + y_s^2}\right) \end{bmatrix} & \mathbf{v}_{\sim s} &= \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \\ J_s^i = \frac{\partial \bar{\mathbf{v}}^i}{\partial \bar{\mathbf{v}}^s} &= \begin{bmatrix} \frac{\partial R}{\partial x_s} & \frac{\partial R}{\partial y_s} & \frac{\partial R}{\partial z_s} \\ \frac{\partial \psi}{\partial x_s} & \frac{\partial \psi}{\partial y_s} & \frac{\partial \psi}{\partial z_s} \\ \frac{\partial \theta}{\partial x_s} & \frac{\partial \theta}{\partial y_s} & \frac{\partial \theta}{\partial z_s} \end{bmatrix} = \begin{bmatrix} \frac{x_s}{R} & \frac{y_s}{R} & \frac{z_s}{R} \\ \frac{y_s}{x_s^2 + y_s^2} & \frac{-x_s}{x_s^2 + y_s^2} & 0 \\ \frac{x_s}{R^2} \left(\frac{z_s}{\sqrt{x_s^2 + y_s^2}} \right) & \frac{y_s}{R^2} \left(\frac{z_s}{\sqrt{x_s^2 + y_s^2}} \right) & \frac{-\sqrt{x_s^2 + y_s^2}}{R^2} \end{bmatrix} \end{aligned}$$

3.6 Analytic Range Image of Flat Terrain

Given the basic kinematic transform, many analyses can be performed. The first is to compute an analytic expression for the range image of a perfectly flat piece of terrain. Let the sensor fixed "s"



coordinate system be mounted at a height h . In the azimuth scanner, the sensor was tilted forward by an angle β . This was done to increase generality and to cover the ERIM scanner as actually used on the HMMWV. It was easy to do because the elevation rotation was last in the azimuth scanner, so tilting the whole sensor was equivalent to tilting the nodding mirror a little more. In this case, the situation is reversed. While the mathematics of rolling the sensor are easy, they are of little use and the mathematics of pitching the sensor obscure the issues. Hence, for this scanner, a trivial transform from sensor coordinates to global coordinates is considered:

$$\begin{aligned} x_g &= x_s \\ y_g &= y_s \\ z_g &= z_s + h \end{aligned}$$

However, a bias angle of ϕ will be introduced into the nodding mirror. This will be the actual elevation angle of the center scanline of the image. Substituting the kinematics into this, the transform from the polar sensor coordinates to global coordinates is obtained:

$$\begin{aligned} x_g &= R \cos \phi \sin \beta \sin \psi \\ y_g &= R \cos \phi \sin \beta \cos \psi \\ z_g &= h - R \sin \phi \end{aligned}$$

Now by setting $z_g = 0$ and solving for R , the expression for R as a function of the beam angles ψ and ϕ for flat terrain is obtained. This is an analytic expression for the range image of flat terrain under the elevation transform.

$$R = h / \sin \phi$$

Notice that in this case, $\sin(\beta) = h/R$. This is the perception ratio defined earlier. As a check on the range image formula, the resulting range image is shown below for $h = 2.5$, $\beta = 16.5^\circ$, a HFOV of 140° , a VFOV of 30° , and an IFOV of 5 mrad. It has 490 columns and 105 rows.

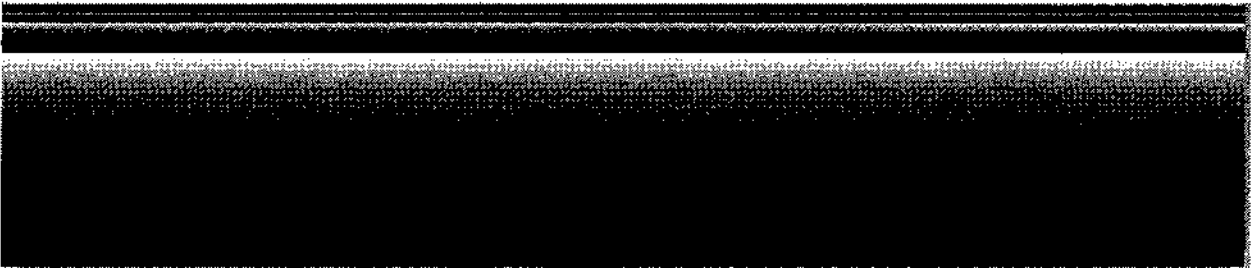


Figure 139 - Elevation Range Image

The edges correspond to contours of constant range of 20 meters, 40 meters, 60 meters, etc. Notice that the contours do not approach the lower corners as they did in the azimuth scanner. This is partly because the tilt β was introduced directly into the elevation scanning mirror as bias. If the sensor z axis was physically tilted, this would not be the case. Substituting this back into the coordinate transform gives the coordinates where each ray intersects the groundplane:

$$\begin{aligned}x_g &= h \sin(\beta) / \tan(\theta) \\y_g &= h \cos(\beta) / \tan(\theta) \\z_g &= 0\end{aligned}$$

Notice that the ratio x/y is $\tan(\beta)$, which is independent of θ . Hence, lines of constant azimuth in the image are *straight radial lines* on flat terrain.

3.7 Field of View

Of basic interest is the region on the groundplane illuminated by the sensor. This can be computed in closed form as follows. From the previous result, it can be verified by substitution and some algebra that:

$$[x_g]^2 + [y_g]^2 = \left(\frac{h}{\tan(\beta)}\right)^2$$

Thus lines of constant azimuth are *circles* on the groundplane.

Let the two extreme columns of the image be given by the polar azimuth angles of $\pm \Psi$. Then the equations of the two lines which bound the field of view are:

$$y_g = \pm \frac{x_g}{\tan(\Psi)}$$

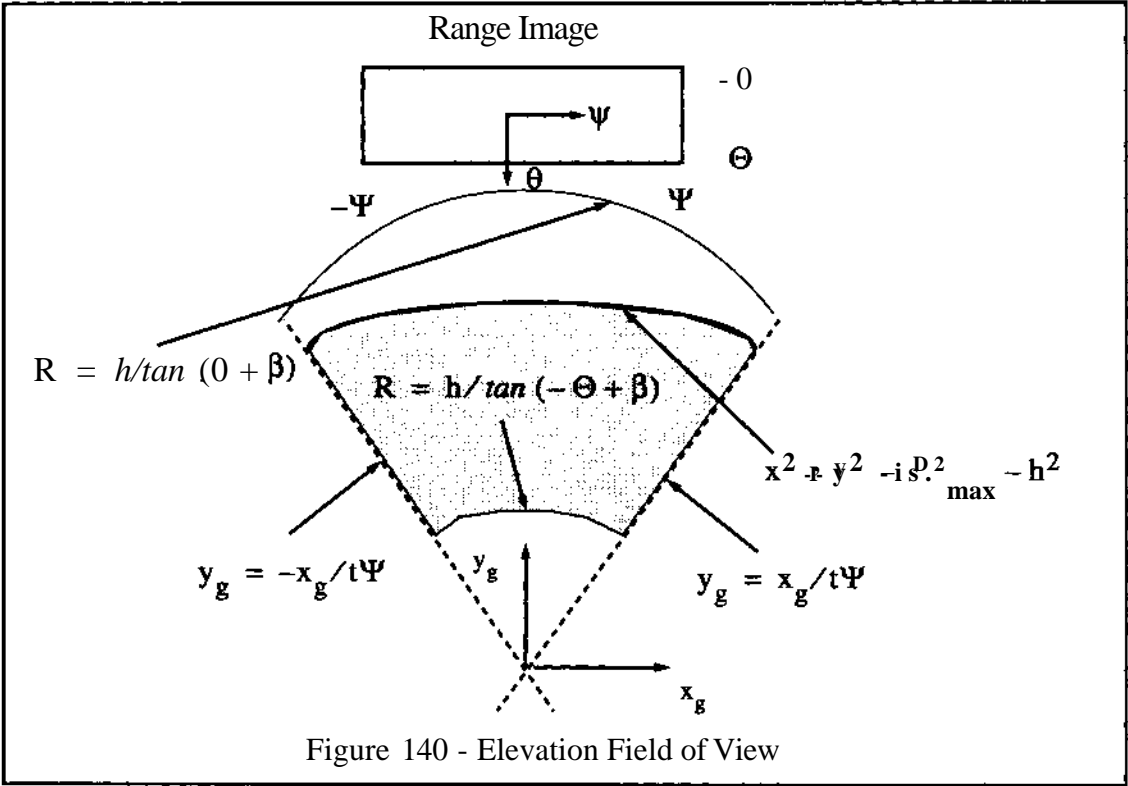
Let the two extreme rows of the image be given by the polar elevation angles of $\pm\theta + \beta$. Then the radii to the proximal and distal circles of the field of view are given by:

$$R = h / \tan (\pm\theta + \beta)$$

If this distal bound is far outside the sensor maximum range, then the maximum range is the radius of the far circle. In this case, the true limit of data is given by the intersection of a sphere centered at the sensor and the groundplane. By inspection, this sphere is given by:

$$x^2 + y^2 + h^2 = R_{\max}^2$$

The shape of the field of view of a single image is given below:



3.8 Resolution

The Jacobian of the groundplane transform has many uses. Most important of all, it provides a measure of sensor resolution on the ground plane. Differentiating the previous result:

$$\begin{bmatrix} dx \\ dy_g \end{bmatrix} = \begin{bmatrix} \frac{h \tan \theta}{t \theta \beta} & \frac{h [s \psi \sec^2 \theta - c \psi \tan \theta]}{t^2 \theta \beta} \\ -\frac{h s \psi}{t \theta \beta} & \frac{h [c \psi \sec^2 \theta + s \psi \tan \theta]}{t^2 \theta \beta} \end{bmatrix} \times \begin{bmatrix} d\psi \\ d\theta \end{bmatrix}$$

The determinant of the Jacobian relates differential areas:

$$dx_g dy_g = \left[\frac{(h \sec \psi)^2}{(t \theta \beta)^3} \right] d\psi d\theta$$

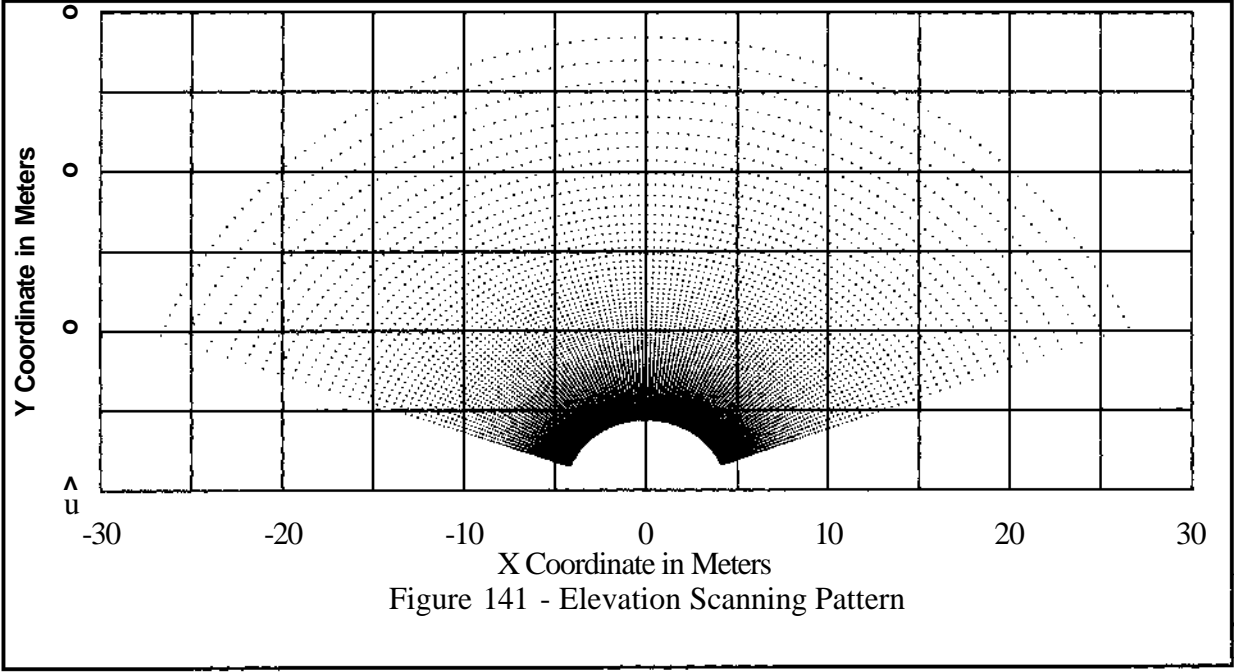
which is identical to the azimuth Jacobian norm except that the denominator is a tangent instead of a sine. Notice that when R is large or small and $y = 0$, the Jacobian norm is:

$$\left[\frac{(h \sec \psi)^2}{(t \theta \beta)^3} \right] = h^2 / \left(\frac{h}{R} \right)^3 = R^2 / \left(\frac{h}{R} \right)$$

which is the same result given earlier. Scanning density is cubic in range.

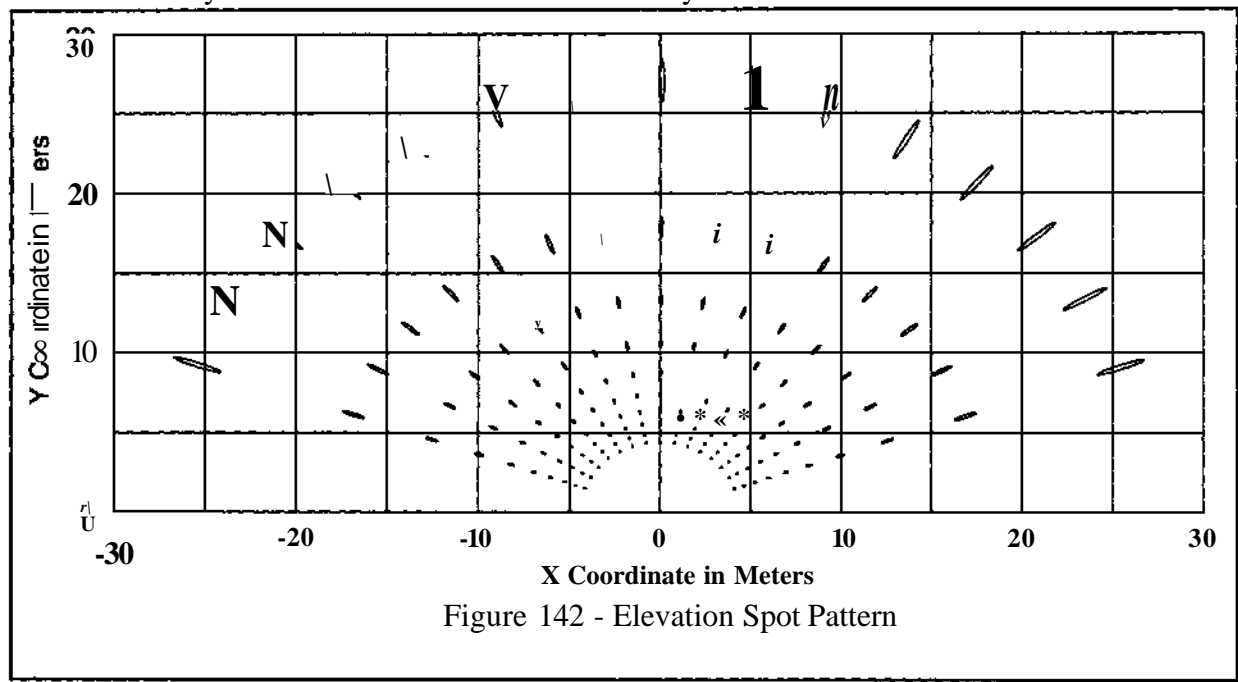
3.9 Elevation Scanning Pattern

The scanning pattern is shown in the following figure with a 10 m grid superimposed for reference purposes. Only every fifth pixel is shown in azimuth to avoid clutter.



3.10 Elevation Spot Pattern

The spot pattern is obtained by intersecting the ground plane with the beam. Spots are shown at one-tenth density in elevation and one-seventh density in azimuth:



Section 4: Kinematics of the Stereo Range Image

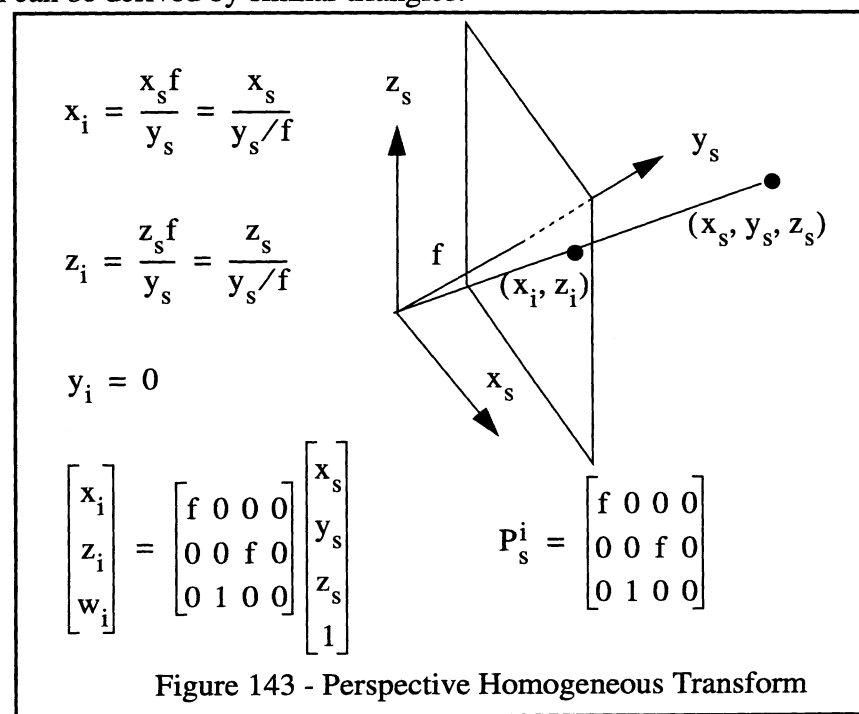
In a manner analogous the laser rangefinders discussed thus far in this appendix, it is possible to describe the stereo range image in similar terms. Stereo range images are represented in the coordinates of one of the cameras used - hence, stereo range images have **perspective** kinematics. Stereo images are generated from the principle of triangulation - noting that the relative shift of an object viewed from two different directions is inversely proportional to its distance measured normal to the image plane. In the simplest case, the two cameras exhibit **epipolar geometry**, that is, the translation vector from one camera to the other is aligned with either the x or z axis of the camera.

4.1 Perspective Projection

In the case of passive imaging systems, a system of lenses forms an image on an array of sensitive elements called a CCD. These systems include traditional video cameras and infrared cameras. The transformation from the sensor frame to the image plane row and column coordinates is the standard perspective projection. This type of transform is unique in two ways:

- it reduces the dimension of the input vector by 1 and hence it discards information
- it requires a post normalization step where the output is divided by the scale factor in order to re-establish a unity scale factor

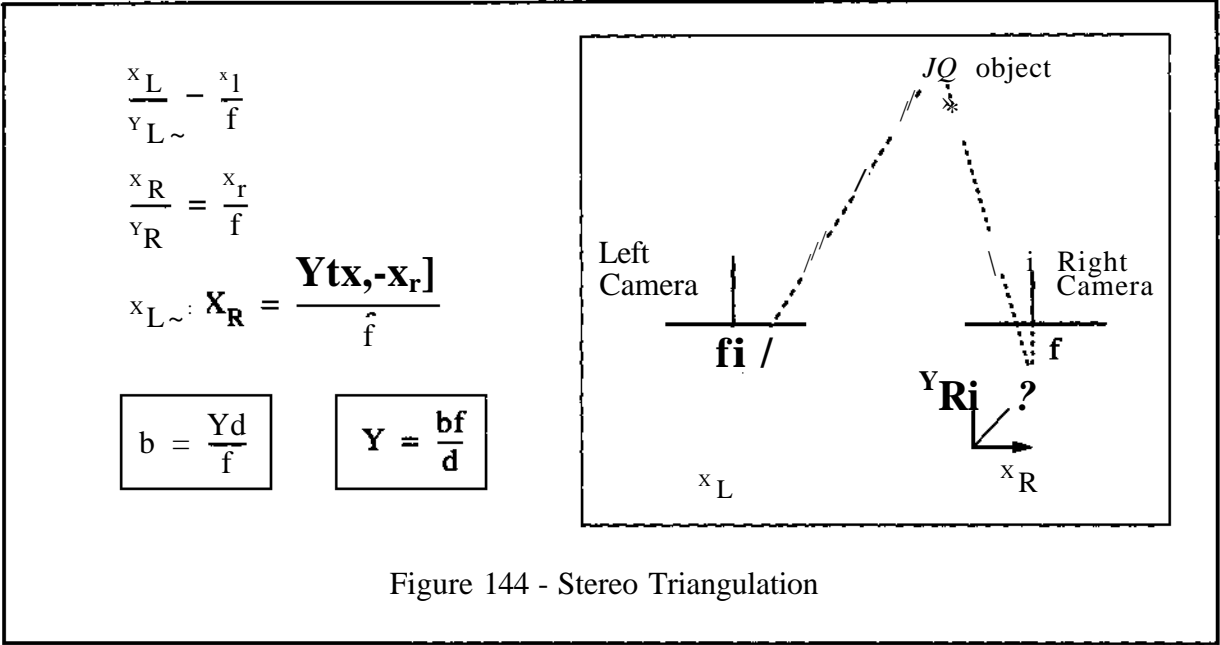
It is simplest to place the origin of the sensor coordinate system at the lens focal point. This transformation can be derived by similar triangles.



Note that all projections are not invertible because they are basically nonsquare matrices. This, of course, is the ultimate source of the difficulty of measuring scene geometry with a single camera. Therefore, the transform is identified by the special capital letter P.

4.2 Stereo Triangulation

The basic stereo triangulation formula for perfectly aligned cameras of epipolar geometry is quoted below for reference purposes. It can be derived simply from the principle of similar triangles.



Once stereo matching is performed, the y coordinate can be determined from the disparity of each pixel. Then the x and z coordinates come from the known unit vector through each pixel which is given by the camera kinematic model. In order to remain consistent with the treatment of rangefinders, azimuth and elevation angles will be defined analogously but note that the *(row,col) coordinates of a pixel are proportional to the tangents of these angles for perspective geometry*. For rangefinders, the (row,col) coordinates are directly proportional to these angles.

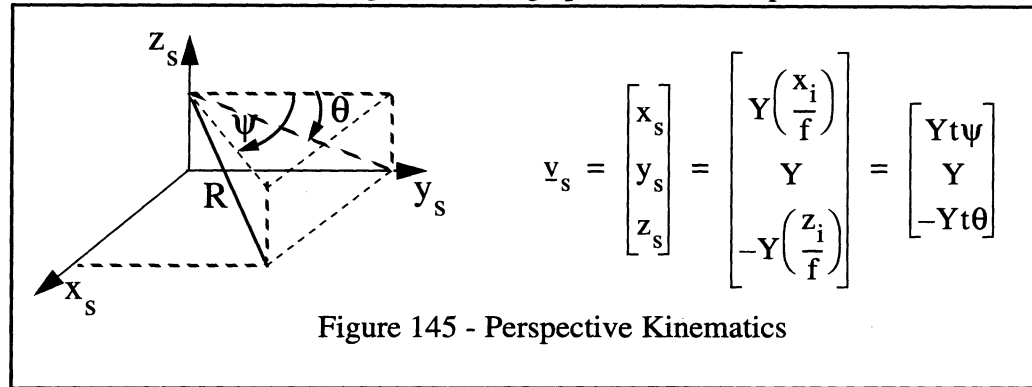
It will be useful at times to define disparity as the tangent of an angle thus:

$$\tilde{s} = \frac{d}{f} = \frac{b}{Y}$$

because this hides the dependence of disparity on focal length of the lens.

4.3 Forward Kinematics

Unlike the case for rangefinders, it is natural to consider the y coordinate known instead of the true 3D range to an object. This should be clear from the following figure - the vector R is not adjacent to either the azimuth or elevation angle. The image plane is the x-z plane:



In comparison to the azimuth and elevation scanners, the kinematics of the perspective image is not based on rotations - rather, pixel coordinates are the tangents of the direction cosines of the ray measured from the y axis.

4.4 Forward Imaging Jacobian

The imaging Jacobian provides the relationship between the differential quantities in the sensor frame and the associated position change in the image. The Jacobian is:

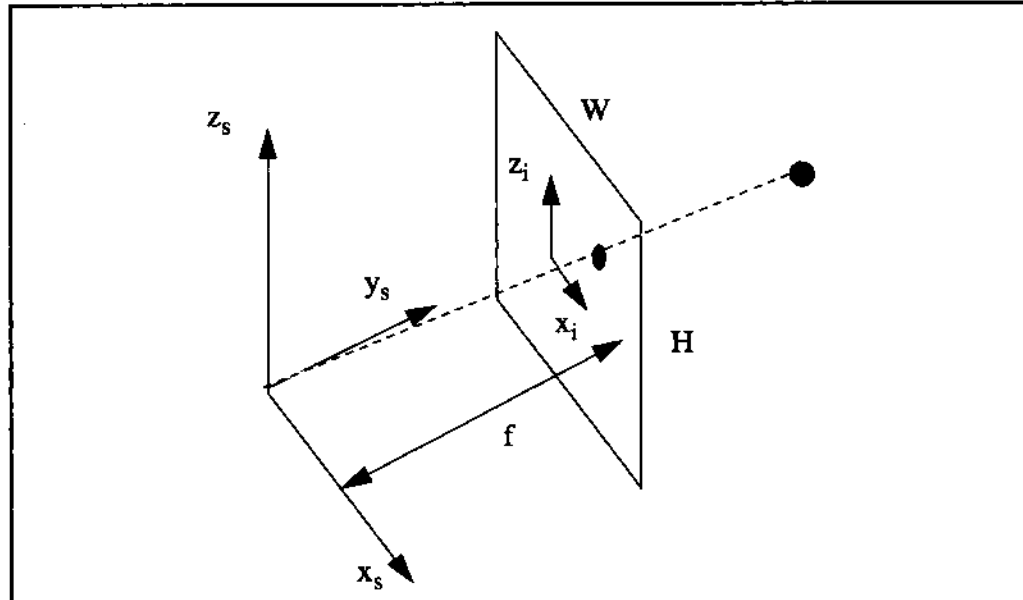
$$\mathbf{v}_i = \begin{bmatrix} Y \\ \psi \\ \theta \end{bmatrix} \quad \mathbf{v}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} Yt\psi \\ Y \\ -Yt\theta \end{bmatrix}$$

$$\mathbf{J}_i^s = \frac{\partial \mathbf{v}^s}{\partial \mathbf{v}^i} = \begin{bmatrix} \frac{\partial x_s}{\partial Y} & \frac{\partial x_s}{\partial \psi} & \frac{\partial x_s}{\partial \theta} \\ \frac{\partial y_s}{\partial Y} & \frac{\partial y_s}{\partial \psi} & \frac{\partial y_s}{\partial \theta} \\ \frac{\partial z_s}{\partial Y} & \frac{\partial z_s}{\partial \psi} & \frac{\partial z_s}{\partial \theta} \end{bmatrix} = \begin{bmatrix} t\psi & Y \sec^2 \psi & 0 \\ 1 & 0 & 0 \\ -t\theta & 0 & -Y \sec^2 \theta \end{bmatrix}$$

4.5 Projection Table

The directions of the rays through each pixel of an imaging sensor are fixed with respect to the sensor frame. The trigonometric overhead of computing this information can be severe in some applications, so it is possible and worthwhile to compute it and store it in tables. These tables can be useful both in real-time perception applications and in off-line ray tracing simulation. This section provides the equations necessary to map (row,column) coordinates onto the direction cosines of the ray through those coordinates.

Let the width of the image plane be W and the height be H as shown in the figure. Let the horizontal and vertical field of view be $HFOV$ and $VFOV$.



A vector from the sensor frame origin to the image plane at the pixel is:

$$\mathbf{d} = [x_i \ f \ z_i]^T$$

A unit vector in this direction is:

$$\hat{\mathbf{u}} = \frac{1}{\sqrt{1 + (x_i/f)^2 + (z_i/f)^2}} \begin{bmatrix} x_i \\ f \\ z_i \end{bmatrix}$$

Which can be written as:

$$\hat{\mathbf{u}} = \frac{1}{\sqrt{1 + (t\psi)^2 + (t\theta)^2}} \begin{bmatrix} t\psi \\ 1 \\ -t\theta \end{bmatrix}$$

The tangents scale linearly with distance along the image plane. Therefore, these are functions of the pixel coordinates as follows:

$$\begin{aligned} \vec{x}_1 &= \frac{(col-cols/2)}{cols} \vec{w} \\ z_i &= \frac{(rows/2-row)}{rows} H \end{aligned}$$

Dividing by the focus distance.:

$$\begin{aligned} t\psi &= \frac{(col-cols/2)}{cols} \frac{W}{f} \\ \tilde{tw} &= \frac{(row - rows/2)}{rows} \frac{H}{f} \end{aligned}$$

This can be written in terms of the field of view as follows:

$$\begin{aligned} t\psi &= \frac{(col-cols/2)}{cols} \tan (HFOV) \\ te &= \frac{(row - rows/2)}{rows} \tan (VFOV) \end{aligned}$$

When the field of view is small enough to satisfy a small angle assumption, then the focus distance and image plane size can be ignored and their ratios can be approximated as follows:

$$\tan (HFOV) \approx HFOV \qquad \tan (VFOV) \approx VFOV$$

4.6 Inverse Kinematics

The forward transform is easily inverted.

$$\begin{bmatrix} Y \\ V \\ e \end{bmatrix} = \begin{bmatrix} y_s \\ atan(x_g/y_s) \\ atan(-z_s/y_s) \end{bmatrix} = h(x_s,y_s,z_s)$$

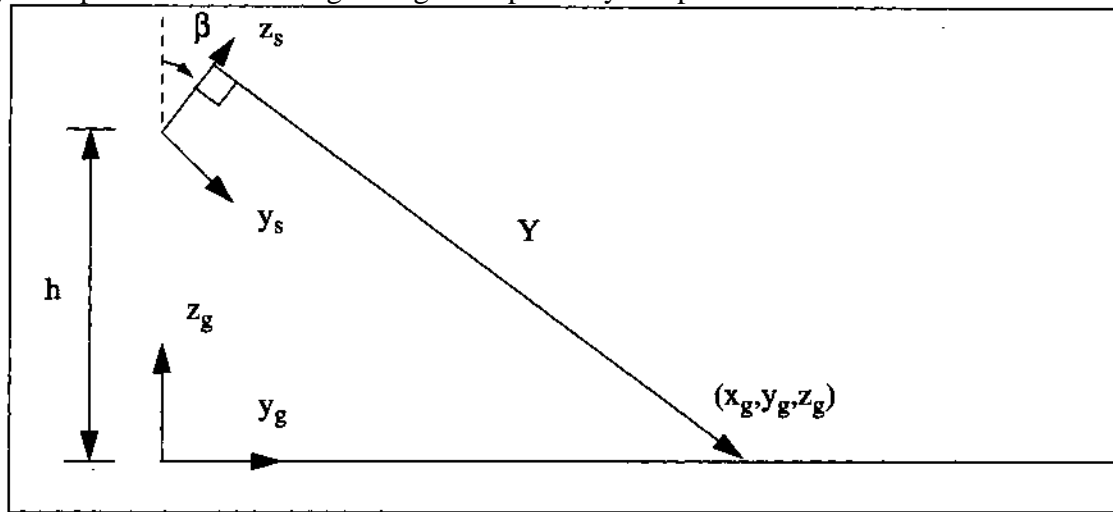
4.7 Inverse Imaging Jacobian

The imaging Jacobian provides the relationship between the differential quantities in the sensor frame and the associated position change in the image. The Jacobian is:

$$\begin{aligned} \mathbf{v}_1 = \begin{bmatrix} Y \\ \psi \\ e \end{bmatrix} &= \begin{bmatrix} y_s \\ \text{atan}(x_s/y_s) \\ \text{atan}(-z_s/y_s) \end{bmatrix} & \mathbf{v}_s &= \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \\ \mathbf{j}_s = \frac{\partial \mathbf{v}_1}{\partial \mathbf{v}_s} &= \begin{bmatrix} \frac{\partial Y}{\partial x_s} & \frac{\partial Y}{\partial y_s} & \frac{\partial Y}{\partial z_s} \\ \frac{\partial \psi}{\partial x_s} & \frac{\partial \psi}{\partial y_s} & \frac{\partial \psi}{\partial z_s} \\ \frac{\partial e}{\partial x_s} & \frac{\partial e}{\partial y_s} & \frac{\partial e}{\partial z_s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{y_s}{x_s^2 + y_s^2} & \frac{-x_s}{x_s^2 + y_s^2} & 0 \\ 0 & \frac{z_s}{y_s^2 + z_s^2} & \frac{-y_s}{y_s^2 + z_s^2} \end{bmatrix} \end{aligned}$$

4.8 Analytic Range Image of Flat Terrain

Given the basic kinematic transform, many analyses can be performed. The first is to compute an analytic expression for the range image of a perfectly flat piece of terrain. Let the sensor fixed "s"



coordinate system be mounted at a height h and tilted forward by a tilt angle of p . Then, the transform from sensor coordinates to global coordinates is:

$$\begin{aligned} x_g &= x_s \\ y_g &= y_s \cos p + z_s \sin p \\ z_g &= -y_s \sin p + z_s \cos p + h \end{aligned}$$

If the kinematics are substituted into this, the transform from the polar sensor coordinates to global coordinates is obtained:

$$\begin{aligned} x_g &= Y \sin \theta \cos \phi \\ y_g &= (Y \cos \theta - Y \tan \theta \sin \phi) \sin p = Y (\cos p - \tan p \sin \phi) = Y \cos p / \cos \phi \\ z_g &= (-Y \sin \theta \sin \phi - Y \tan \theta \cos \phi) \cos p + h = h - Y (\sin p + \tan p \cos \phi) = h - Y \sin p / \cos \phi \end{aligned}$$

Now by setting $z_g = 0$ and solving for Y , the expression for Y as a function of the beam angles θ and ϕ for flat terrain is obtained. This is an analytic expression for the range image of flat terrain under the perspective transform.

$$Y = h \cos \phi / (\sin p)$$

Notice that when Y is large $s(3 = h/R$. This is the perception ratio defined earlier. As a check on the range image formula, the resulting range image is shown below for $h = 2.5$, $p = 16.5^\circ$, a HFOV of 140° , a VFOV of 30° , and an IFOV of 5 mrad. It has 490 columns and 105 rows.

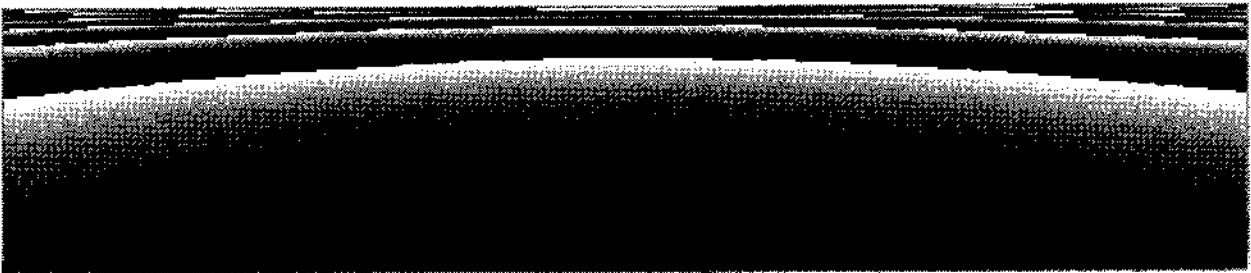


Figure 146 - Perspective Range Image

The edges correspond to contours of constant range of 20 meters, 40 meters, 60 meters, etc. The true 3D range is shown, not the y coordinate. The y coordinate image is similar to the elevation scanner image - with contours of constant range appearing as horizontal lines. Substituting this back into the coordinate transform gives the coordinates where each ray intersects the groundplane:

$$\begin{aligned} x_g &= htv/c9/sep \\ y_g &= h/tep \\ z_g &= 0 \end{aligned}$$

Notice that the y coordinate is independent of θ and hence, lines of constant elevation in the image **are straight lines along the y-axis on flat terrain**.

4.9 Disparity Image of Flat Terrain

The disparity image of flat terrain is immediate from the range image³:

$$\delta = \frac{d}{f} = \frac{b}{Y} = \frac{bs\theta\beta}{hc\theta} = \left(\frac{b}{h}\right)[c\beta t\theta + s\beta]$$

The disparity image corresponding to the above range image is shown below. A baseline of 0.25 meters was chosen to give 11 integer disparity levels so that the bands of disparity would be obvious.

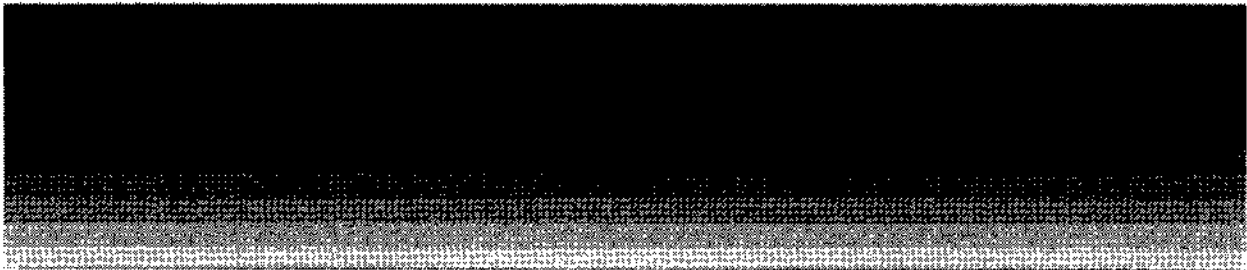


Figure 147 - Disparity Image

3. This expression is undefined above the horizon because it is based on substituting flat terrain into the sensor kinematics.

The disparity curve for every column is identical in this example. It is plotted below:

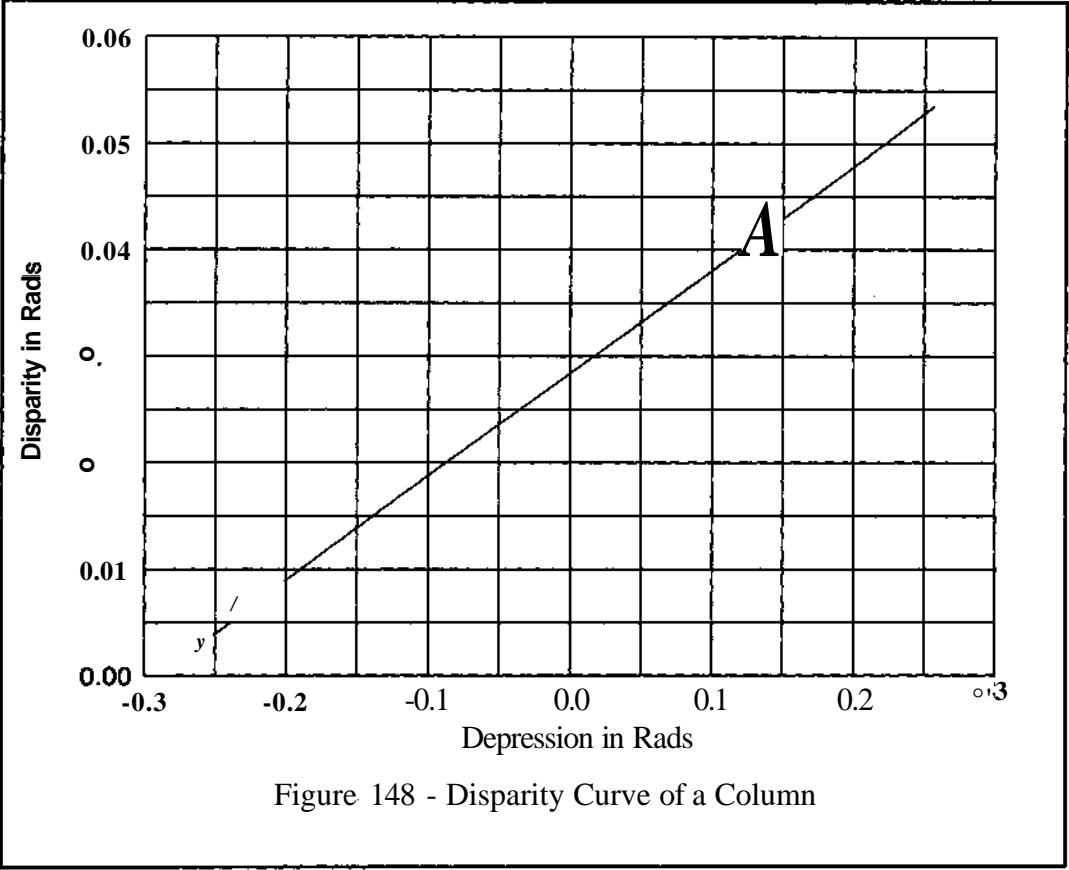


Figure 148 - Disparity Curve of a Column

4.10 Disparity Gradient of Flat Terrain

The behavior of the disparity gradient is a central concern in stereo correlation. Differentiating the above expression:

$$\frac{\partial \delta}{\partial \theta} = \left(\frac{b}{h}\right)c\beta \sec^2 \theta$$

Noting that the VFOV is usually small, and that sensor tilt is normally a small angle, this expression is basically the constant b/h . This is the **normalized baseline** defined earlier. The disparity gradient can be interpreted in units of radians of disparity per radian of depression angle of a pixel, or in the case of square pixels, it can be interpreted in units of pixels per row. The disparity gradient places limits on the height of the stereo correlation window. When it exceeds one pixel over the height of the window, the two regions imaged by each camera are misaligned by one pixel - and they will not match unless the local scene spatial frequencies are low enough.

4.11 Field of View

Of basic interest is the region on the groundplane illuminated by the sensor. This can be computed in closed form as follows. From the previous result, it can be verified by substitution and some algebra that:

$$\left[\frac{x_g}{t\Psi}\right]^2 - y_g^2 = h^2 \frac{[c^2\theta - c^2\theta\beta]}{s^2\theta\beta} \approx h^2$$

Thus (when the VFOV is small, $c^2\theta \ll 1$) lines of constant azimuth are *hyperbolas* on the groundplane.

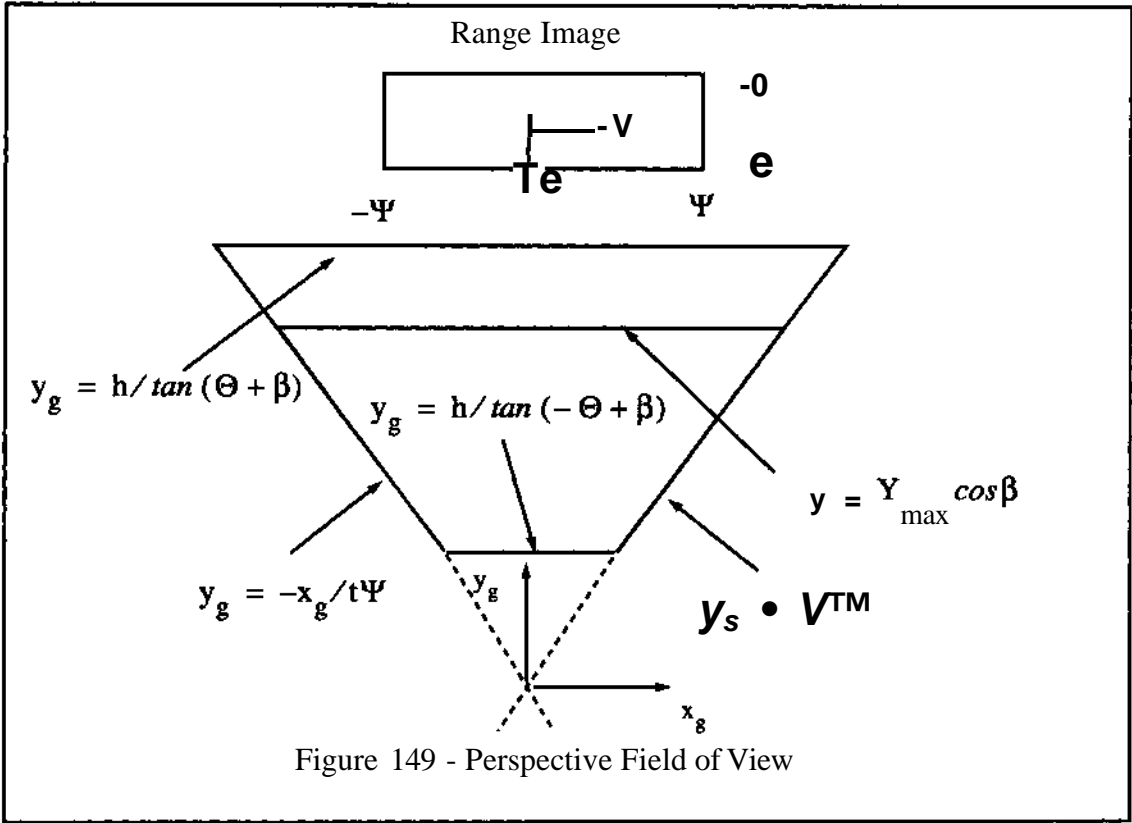
Let the two extreme columns of the image be given by the polar azimuth angles of $\pm^*\text{F}$. Then, for the two extreme columns of the image given by $\Psi = \pm^*\text{F}$, the asymptotes of the hyperbola are of the form:

$$\frac{x_g}{t\Psi} = \pm y$$

Let the two extreme rows of the image be given by the polar elevation angles of ± 0 . Then the proximal and distal lines of the field of view are given by:

$$y_{\sigma} = h/\text{fon}(\pm 0+\text{fi})$$

However, for most sensor tilt angles, the distal bound is far outside the sensor maximum range. Hence, the true limit of data is given by the intersection of a plane, parallel to the image plane, and a fixed maximum distance Y_{max} away, and the groundplane. The shape of the field of view of a single image is given below:



4.12 Resolution

The Jacobian of the groundplane transform has many uses. Most important of all, it provides a measure of sensor resolution on the ground plane. Differentiating the previous result:

$$\begin{bmatrix} \frac{dx_g}{dy_g} \\ \mathbf{K} \end{bmatrix} = \begin{bmatrix} \frac{hce(sec\psi)^2}{sep} & \frac{-h cet\psi/c\psi p}{(sep)^2} & \frac{-h setJ}{s6p} \\ 0 & \frac{h}{(sep)^2} \end{bmatrix} \times \begin{bmatrix} d\psi \\ d\theta \end{bmatrix}$$

The determinant of the Jacobian relates differential areas:

$$dx_g dy_g = \frac{WcB(sec\psi)^*J}{L (sep)^3} d\psi d\theta$$

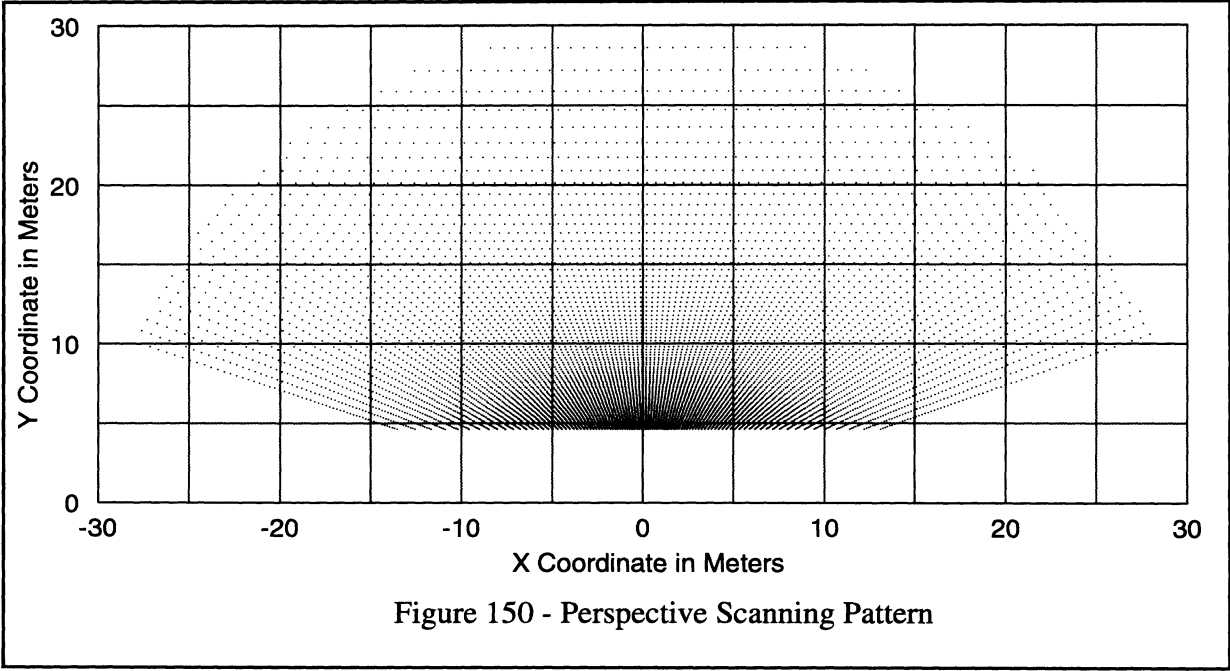
Notice that when R is large, θ is small, and $\psi = 0$, the Jacobian norm can be approximated by:

$$\left[\frac{(h \sec \psi)^2}{(s \theta \beta)^3} \right] \approx h^2 / \left(\frac{h}{R} \right)^3 = R^2 / \left(\frac{h}{R} \right)$$

which is the same result given earlier. The scanning density is proportional to the cube of the range.

4.13 Perspective Scanning Pattern

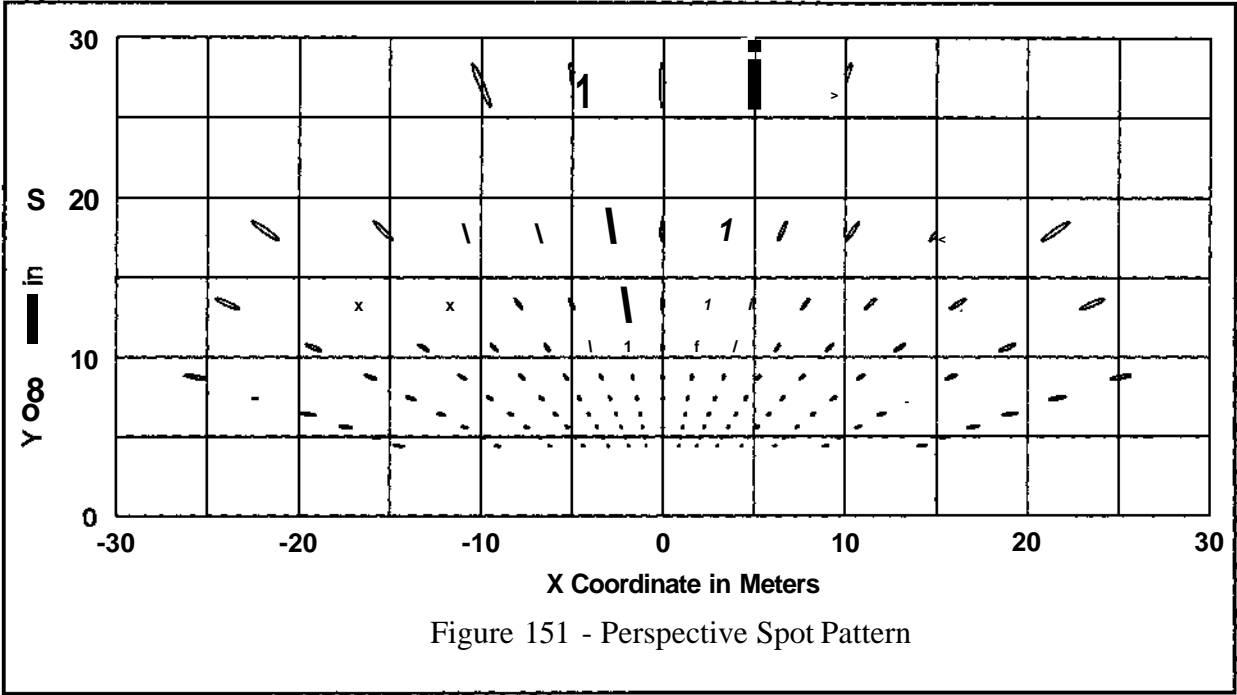
The scanning pattern is shown in the following figure with a 10 m grid superimposed for reference purposes. Only every fifth pixel is shown in azimuth to avoid clutter. The result is very similar to



the azimuth scanner - as would be expected when the vertical field of view is small.

4.14 Perspective Spot Pattern

The spot pattern is obtained by intersecting the ground plane with the "beam" emanating from the image plane for each pixel. Spots are shown at one-tenth density in elevation and one-seventh density in azimuth:



Chapter 3: The State Space Kalman Filter

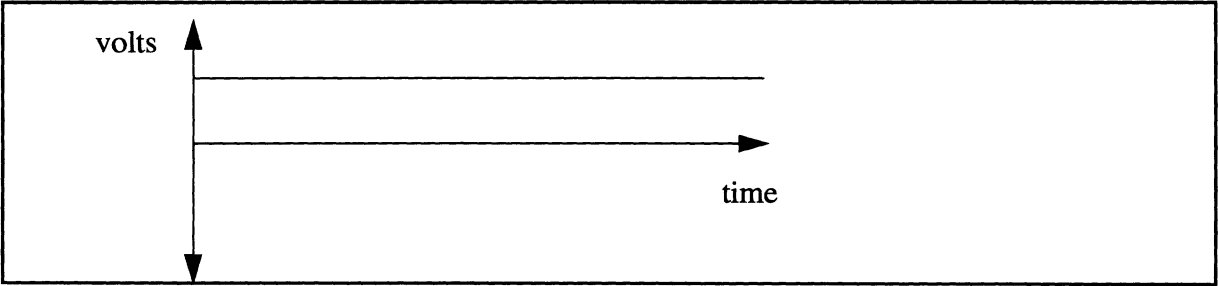
Section 1: Random Processes

Before looking into the Kalman filter itself, it is necessary to understand the error models upon which it is based. This section presents a very abbreviated discussion of the aspects of the theory of random signals which are applicable to the appendix.

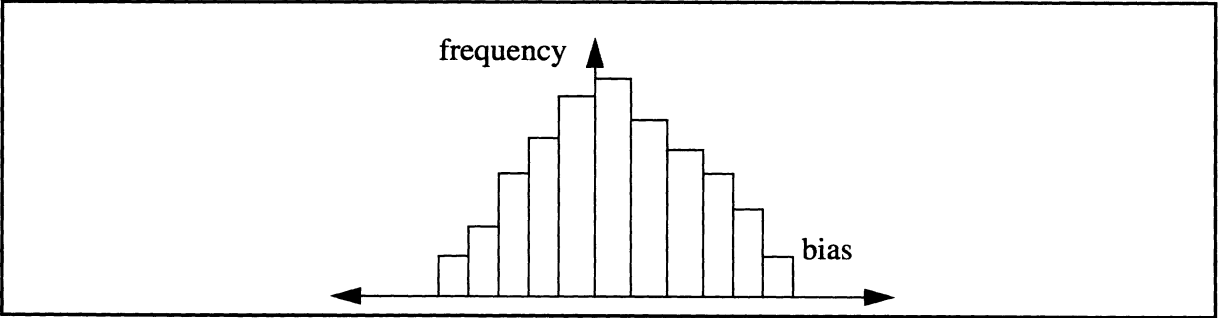
Noise sources in the Kalman filter are modelled as random processes. The **random process** can be considered to be a collection of functions of time called an **ensemble**, any one of which may be observed in a particular experiment. Exactly which one occurs is a random variable, and therefore the value of the chosen function at any time *across all experiments* is a random variable. Usually, the statistical variation of the ensemble of functions at any time is known.

1.1 Random Constant

Consider for example a random constant process. How can a “process” be constant and random at the same time? Well, consider a bin which contains alot of gyroscopes, and suppose that the manufacturer has said that the bias torques of all gyroscopes manufactured follow some sort of distribution. To be sure, the bias torque for any particular gyroscope could be plotted on, say, an oscilloscope and it would produce the following time signal:



Nothing random about that. Suppose, however, that alot of people reach into the bin at the same time and pick out a gyroscope, wait ten seconds, and record the bias value. The values of the bias torques at $t = 10$ seconds could be plotted in a histogram to produce something like.



This distribution is the probability distribution of the random constant process at $t = 10$ seconds. Thus, while any member of the ensemble of functions is **deterministic in time**, the **choice** of the function is random. In general, there may be a family of related functions of time where some important parameter varies randomly - a family of sinusoids of random amplitude, for instance. Even though there is no way to predict which function will be chosen, it may be known that all functions are related by a single random parameter and from this knowledge, it is possible to compute the distribution for the process as a function of time.

1.2 Bias, Stationarity, Ergodicity and Whiteness

A random process is **unbiased** if its expected (i.e. average) value is zero for all time. A random process is said to be **stationary** if the distribution of values of the functions in the ensemble is not varying with time. Conceptually, a movie of the histograms above for the gyroscopes for each second of time would look like a still picture. An **ergodic** random process is one where time averaging is equivalent to ensemble averaging - which is to say that everything about the process can be discovered by watching a single function for all time, or by watching all signals at a single instant. A **white** signal is one which contains all frequencies.

It is clear that fluency with these concepts requires the ability to think about a random process in three different ways:

- in terms of its probability distribution
- in terms of its evolution over time
- in terms of its frequency content.

These different views of the same process will be discussed in the next sections, as well as methods for converting back and forth between them.

1.3 Correlation Functions

Correlation is a way of thinking about both the probability distributions of a random process and its time evolution. The **autocorrelation function** for a random process $x(t)$ is defined as:

$$R_{xx}(t_1, t_2) = E[x(t_1)x(t_2)]$$

so it's just the expected value of the product of two random numbers - each of which can be considered to be functions of time. The result is a function of both times. Let:

$$x_1 = x(t_1) \qquad x_2 = x(t_2)$$

then the autocorrelation function is, by definition of expectation:

$$R_{xx}(t_1, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2 f(x_1, x_2) dx_1 dx_2$$

where $f(x_1, x_2)$ is the joint probability distribution. The autocorrelation function gives the "tendency" of a function to have the same sign and magnitude (i.e. to be **correlated**) at two different times.

For smooth functions it is expected that the autocorrelation function would be highest when the two times are close because the smooth function has little time to change. This idea can be expressed formally in terms of the frequency content of the signal, and conversely, the autocorrelation function says a lot about how smooth a function is. Equivalently, the autocorrelation function specifies how fast a function can change, which is equivalent to saying something about the magnitude of the coefficients in its Taylor series, or its Fourier series, or its Fourier transform. All of these things are linked.

The crosscorrelation function relates two different random processes in an analogous way:

$$R_{xy}(t_1, t_2) = E[x(t_1)y(t_2)]$$

For a **stationary process** the correlation function is dependent only on the difference $x = t_x - t_y$. When the processes involved are unbiased, the correlation functions give the variance and covariance of the indicated random variables. This is easy to see by considering the general formula for variance and setting the mean to zero.

Thus, for stationary unbiased processes, the correlation functions *are* the variances and covariances expressed as a function of the time difference:

$$R_{xx}(\tau) = \sigma_{xx}^2(\tau) \qquad R_{xy}(\tau) = \sigma_{xy}^2(\tau)$$

Also, for stationary unbiased processes:

$$\sigma_{xx}^2 = R_{xx}(0) \qquad \sigma_{xy}^2 = R_{xy}(0)$$

1.4 Power Spectral Density

The **power spectral density** is just the Fourier transform of the autocorrelation function, thus:

$$S_{xx}(j\omega) = \mathcal{F}[R_{xx}(t)] = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j\omega\tau} d\tau$$

The power spectral density is a direct measure of the frequency content of a signal, and hence, of its power content. Of course, the inverse Fourier transform yields the autocorrelation back again.

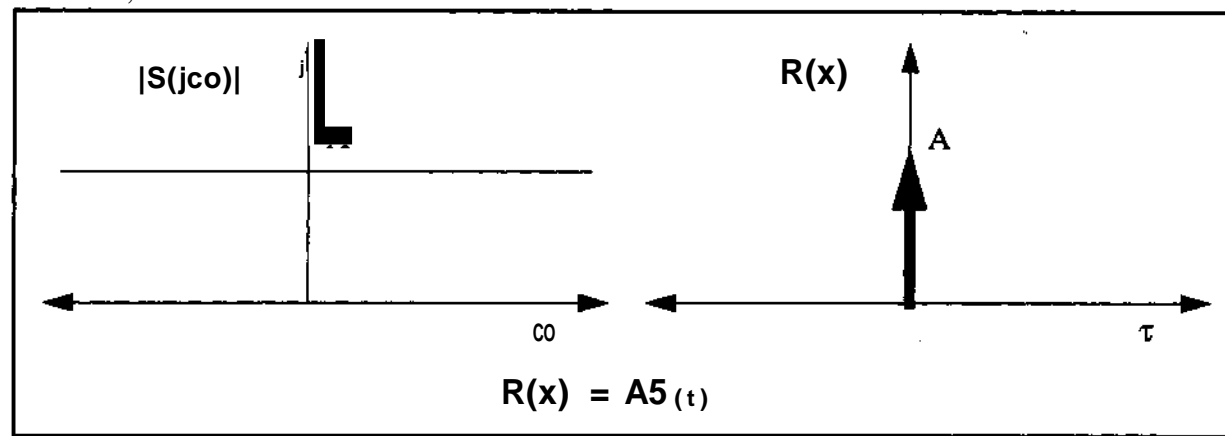
$$R_{xx}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xx}(j\omega) e^{j\omega\tau} d\omega$$

Similarly, the **cross power spectral density** function is:

$$S_{xy}(j\omega) = \mathcal{F}[R_{xy}(x)] = \int_{-\infty}^{\infty} R_{xy}(\tau) e^{-j\omega\tau} d\tau$$

1.5 White Noise

White noise is defined as a stationary random process whose power spectral density function is constant. That is, it contains all frequencies of equal amplitude. If the constant **spectral amplitude** is A , then the corresponding autocorrelation function is given by the inverse Fourier transform of a constant, which is the Dirac delta.



Thus, knowing the value of a white noise signal at some instant of time says absolutely nothing about its value at any other time, and this is because it is possible for it to jump around at infinite frequency.

1.6 Kalman Filter Noise Model

With these preliminaries, it is possible to define and understand the noise model of the Kalman filter. The noises modelled in a Kalman filter must be:

- unbiased (have zero mean for all time)
- Gaussian (have a Gaussian distribution for all time)
- white (contain all frequencies)

This model is a mathematical idealization since white noise cannot occur in nature because it requires infinite energy. The formulas of the previous section permit mapping these white noise processes onto variances and covariances which are easier to think about.

1.6.1 White Noise Process Covariance

Specifically, let the white unbiased Gaussian random process $x(t)$ have power spectral density S . Then the variance is:

$$\sigma_{xx}^2 = S_P$$

Thus *the variance of a white noise process is its spectral amplitude*. This one is easy.

1.6.2 Random Walk Process Covariance

Now suppose that the white noise process represents the time derivative of the real variable of interest. Thus:

$$\sigma_{xx}^2 = S_P$$

The question to be answered is what is σ_{xx}^2 , σ_{xx}^2 and σ_{xx}^2 . These can be derived from first principles. Consider that the value of x at any time must be, by definition, the integral of \dot{x} as follows:

$$\sigma_{xx}^2(x^2) = E \left[\int_{L_0}^{rt} \dot{x}(u) du \int_0^t \dot{x}(v) dv \right]$$

this can be written easily as a double integral since the variables of integration are independent thus:

$$G_{xx}^2 = E \left[\int_{L_0}^{rt} \int_0^t \dot{x}(u) \dot{x}(v) dudv \right]$$

interchanging the order of expectation and integration:

$$\sigma_{xx}^2 = \int_0^t \int_0^t E[\dot{x}(u)\dot{x}(v)]dudv$$

but now the integrand is just the autocorrelation function, which, for white noise is the Dirac delta, so:

$$\sigma_{xx}^2 = \int_0^t \int_0^t S_p \delta(u-v) dudv = \int_0^t S_p dv = S_p t$$

Thus the variance of the integral of white noise grows linearly with time. Also, the standard deviation grows with the square root of time. The process $x(t)$ is called a **random walk**.

1.6.3 Integrated Random Walk Process Covariance

If the second derivative of $x(t)$ with respect to time is a white process, then the variance in $x(t)$ is:

$$\sigma_{xx}^2 = \int_0^t \int_0^t \int_0^t \int_0^t S_p \delta(u-v) \delta(s-t) dudvdsdt = \frac{S_p t^2}{2}$$

Thus the variance grows with the square of time and the standard deviation with time. This is called an **integrated random walk**. This technique can be used in general to compute the elements of a covariance matrix given the spectral amplitudes of the noises.

1.7 **Scalar Uncertainty Propagation**

It is useful to know the following rough rule for the propagation of error in a sum of scalar random variables with identical statistics:

$$y = \sum_{i = 1}^n x_i$$
$$\sigma_y^2 = \sum_{i = 1}^n \sigma_{x_i}^2 = n\sigma_x^2$$

Thus, the variance in the sum is the sum of the variances of the individual terms of the sum. When they are equal, the variance grows linearly with the number of elements in the sum. In practical use, the expression gives the development of the standard deviation versus time because:

$$n = \frac{t}{\Delta t}$$

in a discrete time system. In this simple model, uncertainty expressed as a standard deviation **grows with the square root of time**. The net result of this is that uncertainty apparently grows rapidly and then levels off as time evolves. This simply arises from the fact that truly random errors tend to cancel each other if enough of them are added.

1.8 **Combined Observations of a Random Constant**

Suppose several redundant measurements of a constant are obtained and that they all have identical statistics. Then the true value can be approximated as the mean of the observations. Under these circumstances, the uncertainty in this mean is:

$$y = \frac{1}{n} \sum_{i = 1}^n x_i$$
$$I_{a_y}^2 = \sum_{i = 1}^n \frac{1}{\sigma_{x_i}^2} = \frac{n}{\sigma_{x_i}^2} \wedge a_y^2 = \frac{1}{n} \sigma_x^2$$

Thus, the variance in the mean decreases with increasing numbers of measurements. Under the assumptions, the standard deviation **decreases with the square root of time**. Experimenters use this every time they take multiple observations and average. This idea that taking and merging multiple observations **reduces the uncertainty of the combined result** is the basic idea of the Kalman filter.

Section 2: Fundamentals of the Discrete Kalman Filter

The Kalman filter was invented by R.E. Kalman and first published in about 1960 [37]. It is a method of estimating the state of a system based on *recursive* measurement of *noisy* data. The filter comes in many forms, including continuous and discrete time variants and linear and nonlinear variants. The Kalman filter was recognized immediately by engineers in the navigation industry as a solution to many formerly intractable problems, and it continues to be used throughout the navigation industry today.

The practical utility of the filter stems from its ability to estimate, say vehicle position, based on a number of measurements which are [19]:

- incomplete: related to some but not all of the variables of interest
- indirect: related indirectly to the quantities of interest
- intermittent: available at irregularly-spaced instants of time
- inexact: corrupted by many forms of error

A central idea in the Kalman filter is to model the system of interest as a linear dynamic system which is excited by noise and whose sensors are also excited by noise. By knowing something about the nature of the noise (its first order statistics), it is possible to construct an optimal estimate of the system state even though the sensors are inexact. This is the fundamental idea of estimation theory. Without knowing the errors themselves, knowledge of their statistics allows construction of useful estimators based solely on that information.

2.1 The State Model of a Random Process

2.1.1 Continuous Model

In the linear systems model, or **state model** of a random process, if time is considered to be continuous, the process is described in the following form:

$$\begin{aligned}\dot{\bar{\mathbf{x}}} &= \mathbf{F}\bar{\mathbf{x}} + \mathbf{G}\bar{\mathbf{w}} && \text{"state" or "process" model} \\ \bar{\mathbf{z}} &= \mathbf{H}\bar{\mathbf{x}} + \bar{\mathbf{v}} && \text{"measurement" or "observation" model}\end{aligned}$$

2.1.2 Discrete Model

If time is considered to be discrete, the process is described in the following form:

$$\begin{aligned}\bar{\mathbf{x}}_{k+1} &= \phi_k \bar{\mathbf{x}}_k + \Gamma_k \bar{\mathbf{w}}_k && \text{"state" or "process" model} \\ \bar{\mathbf{z}}_k &= \mathbf{H}_k \bar{\mathbf{x}}_k + \bar{\mathbf{v}}_k && \text{"measurement" or "observation" model}\end{aligned}$$

There is, of course, a way to transfer from one form to the other, which will be given shortly. In the discrete model, the names and sizes of the vectors and matrices are:

$\bar{\mathbf{X}}_k$ is the $(n \times 1)$ system **state vector** at time t_k
 ϕ_k is the $(n \times n)$ **transition matrix** which relates $\bar{\mathbf{X}}_k$ to $\bar{\mathbf{X}}_{k+1}$ in the absence of a forcing function
 \mathbf{F}_k is the $(n \times n)$ **process noise distribution matrix** which transforms the $\bar{\mathbf{w}}_k$ vector into the coordinates of $\bar{\mathbf{x}}_k$
 $\bar{\mathbf{w}}_k$ is a $(n \times 1)$ white **disturbance sequence** or **process noise sequence** with known covariance structure.
 $\bar{\mathbf{z}}_k$ is a $(m \times 1)$ **measurement** at time t_k
 \mathbf{H}_k is a $(m \times n)$ **measurement matrix** or **observation matrix** relating $\bar{\mathbf{X}}_k$ to $\bar{\mathbf{z}}_k$ in the absence of measurement noise

The **state vector** for a dynamic system is any set of quantities sufficient to completely describe the unforced motion of the system. Given the state at any point in time, the state at any future time can be determined from the control inputs and the system model. Intuitively, a state vector contains values for all variables in the system up to one order less than the highest order derivative represented in the model. This is, of course, the exact number of initial conditions required to solve a differential equation.

The system model is basically a matrix linear differential equation. Such a model considers the process to be the result of passing white noise through a system with linear dynamics. The covariance matrices for the white sequences are:

$$\mathbf{E}(\bar{\mathbf{w}}_k \bar{\mathbf{w}}_i^T) = \delta_{ik} \mathbf{Q}_k \quad \mathbf{E}(\bar{\mathbf{v}}_k \bar{\mathbf{v}}_i^T) = \delta_{ik} \mathbf{R}_k \quad \mathbf{E}(\bar{\mathbf{w}}_k \bar{\mathbf{v}}_i^T) = 0, \forall (i, k)$$

where δ_{ik} is the Kronecker delta.

2.2 A Word on the Transition Matrix

Often, the differential equations of a system in their time continuous form are known. However, a Kalman filter is generally implemented in discrete time in a computer cycling at a finite rate. In the theory of linear systems, this matter is discussed in some detail, and the transition matrix Φ figures prominently. In order to implement the Kalman filter, much of what is known about the transition matrix is unnecessary. It suffices to know that the time continuous matrix differential equation:

$$\dot{\bar{x}} = F\bar{x}$$

can always be transformed into:

$$\bar{x}_{k+1} = \Phi_k \bar{x}_k$$

The only question is how hard it is to do. When the F matrix is constant in time and the equation is linear (no elements of x occur inside F), then the transition matrix is a function only of the time step Δt and it is given by the matrix exponential:

$$\Phi_k = e^{F\Delta t} = I + F\Delta t + \frac{(F\Delta t)^2}{2!} + \dots$$

In practice, the transition matrix can often be written by inspection. When this is not possible, writing a few terms of the above series often generates recognizable series in each element of the matrix partial sum, and the general form for each term can be generated by inspection. Other times, higher powers of F conveniently vanish anyway. When Δt is much smaller than the dominant time constants in the system, just the two-term approximation:

$$\Phi_k = e^{F\Delta t} = I + F\Delta t$$

is sufficient.

2.3 Low Dynamics Assumption

When F depends on time, so does Φ and it satisfies the matrix version of the same differential equation as the state vector thus:

$$\frac{d\Phi}{dt} = F(t) \Phi$$

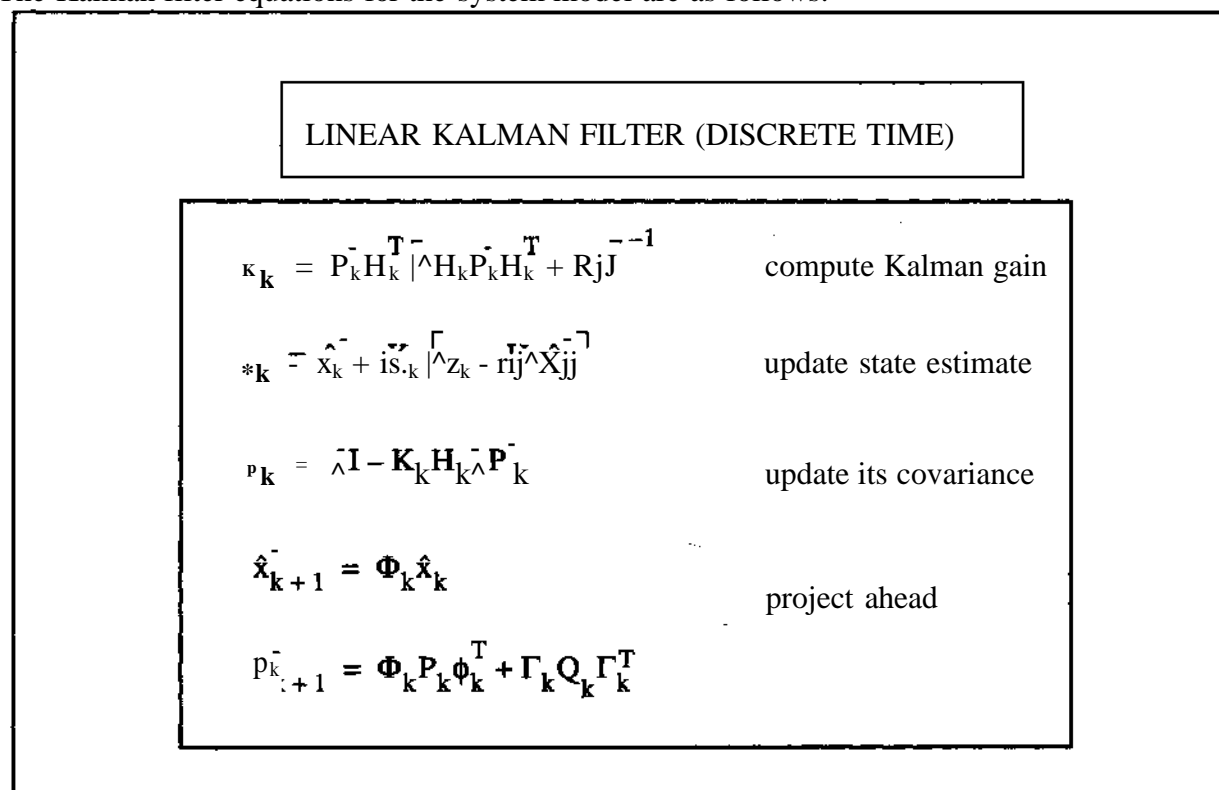
If F is assumed to be slowly varying relative to Δt , then the matrix exponential can still be used. This will be called the **low dynamics assumption**. It is a big assumption as the time step gets larger¹.

1. The matrix exponential arises simply because the system model is a differential equation. When F is constant, it is the exact solution and it is possible to *approximate the known solution* by a Taylor series. When F is time-dependent, the matrix exponential is no longer the right answer. Then the approach is to *approximate the differential equation* itself by assuming that F is constant to get an *exact solution to an approximate differential equation*.

2.4 Discrete Filter Equations

The Kalman filter propagates both the state and its covariance forward in time, given an initial estimate of the state. At any point in time, the state estimate prior to incorporation of any new measurements will be denoted by $\hat{\mathbf{x}}^-$, where the hat denotes an estimate and the super minus denotes the estimate prior to incorporation of the measurements (running one iteration of the filter equations).

The Kalman filter equations for the system model are as follows:



The proof that these equations constitute an optimal filter is remarkably straightforward for an algorithm that is only thirty years old. It is provided in [9]. **The equations are not run all at once.** The last two run at high frequency and the first three are run when measurements are available. For the first three, the process is started by entering the prior estimate $\hat{\mathbf{x}}_k^-$ and its covariance \mathbf{P}_k^- . For each cycle of the system model, the state transition matrix Φ_k , and the disturbance covariance \mathbf{Q}_k must be known. For each cycle of the Kalman filter equations, the measurement matrix \mathbf{H}_k , and the sequence covariance \mathbf{R}_k^* must be known *a priori* or computed based on the measurements and partial prior knowledge.

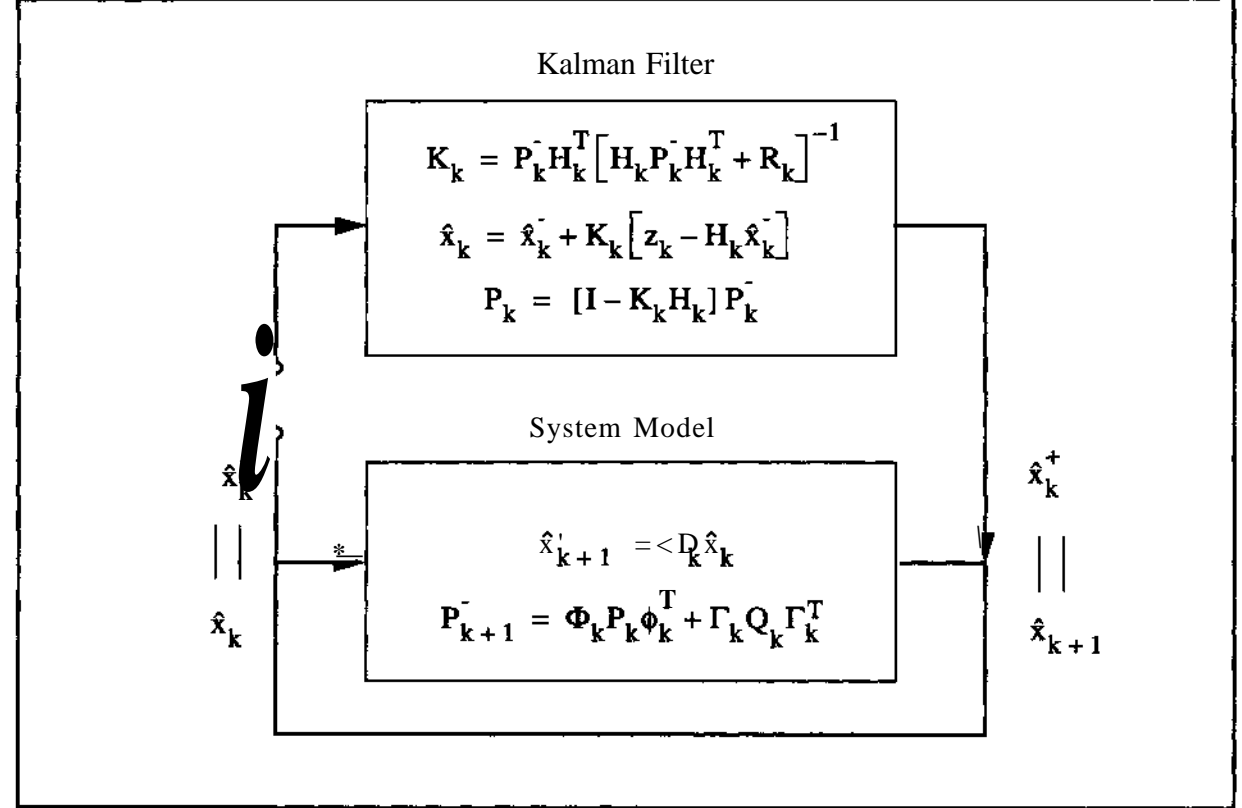
While there are several roughly equivalent forms of the Kalman filter equations and several different forms of system model, the one adopted here is in common use in the navigation industry since it avoids certain numerical problems with the matrix \mathbf{R}_k . Another advantage of this formulation is that it requires inversions of matrices of order m (number of measurements) which is usually less than n (the number of states). Indeed, it is possible to, under assumptions necessary for other reasons, set m to 1 and **avoid matrix inversion completely**.

2.5 Time and Updates

In thinking about the operation of the filter, it is important to distinguish the *time element* from the *arrival of measurements*. Conceptually, the projection of the system state forward in time proceeds *solely on a measurement of time*. That is, by the definition of state, it is possible to propagate the state indefinitely forward in time based only on the initial conditions, and the forcing function, if there is one. The formulation used here assumes no forcing function.

Measurements are conceptualized as indirect measurements of state, and they arrive intermittently. When they arrive, they are incorporated into the state estimate through the Kalman gain but they are not strictly necessary. The number of measurements m , may be greater or less than n , and they may be redundant measurements of the same quantity.

This conceptualization of the filter is represented in the figure below. The state equations cycle forever, and whenever a measurement is available, the switch closes *after the state has been predicted for that cycle* by the system model, and the filter proper is executed.



This situation has a perfect analog in control theory. If the system model is perfect in a feedback loop, then there is no need of feedback, and the control law can run "open loop". Sensors are incorporated to provide feedback as a pragmatic answer to the problem that the model is never perfect. Similarly, the Kalman filter is the feedback element here, which is necessary in practice, but it is important to recognize that it is completely independent of the system dynamics.

2.6 Measurement Model

Notice that the model presents how the measurements are derived from the state, that is, the operation of the sensor itself. Often, in other applications, the process which converts a measurement into a state estimate is considered. That is, the problem of *perception*. However, this model is the simpler reverse process of *sensing* itself.

This is important to keep in mind because the filter is able to use underdetermined measurements of state for this reason. For example, if a single range measurement is available, the filter can use it to attempt to estimate two position coordinates or even more. This situation cannot persist for too long a period of time but single underdetermined measurements of multidimensional state vectors are quite legal.

2.7 Observability

In general, situations may arise, however, where there are not enough measurements in the entire sensor suite to predict the system state over the long term. These are called **observability** problems and they can be detected when diagonal elements of P_k are diverging with time.

Observability problems can be fixed by reducing the number of state variables (i.e by incorporating the assumption that some are not too relevant) or by adding additional sensors. Observability is a property of the entire model including both the system and the measurement model, so it changes every time the sensors change.

Formally, *a system is observable if the initial state can be determined by observing the output for some finite period of time*. Generalizing from Gelb [27], consider the discrete, nth order, constant coefficient linear system:

$$x_{k+1} = \Phi x_k$$

for which there are m noise free measurements:

$$z_k = Hx_k \quad k = 0, m-1$$

where each H is an m X n matrix. The sequence of the first n measurements can be written as:

$$\begin{aligned} z_0 &= Hx_0 \\ z_1 &= Hx_1 = H\Phi x_0 \\ z_2 &= Hx_2 = H(\Phi)^2 x_0 \\ &\vdots \\ z_{n-1} &= Hx_{n-1} = H(\Phi)^{n-1} x_0 \end{aligned}$$

This can be rewritten as the augmented set of equations:

$$Z = \Xi^T x_0 \quad \text{or} \quad Z^T = x_0^T \Xi$$

If the initial state x_0 is to be determined from this sequence of measurements then, the matrix:

$$H = [H^T O^T H^T \dots (\Phi^T)^{n-1} H^T]$$

must have rank² n .

2.8 Uncertainty Transformation

If z is an arbitrary measurement of the vector \bar{x} through some nonlinear relationship:

$$Z = f(X)$$

Then it can be easily shown using a Taylor series approximation and the definition of the covariance matrix that:

$$\text{Cov}(Az) = \text{Exp}[AzAz^T] = H \text{Exp}[AxAx^T] H^T = H \text{Cov}(Ax) H^T$$

where H is the Jacobian of f . This relationship is responsible for the terms involving the measurement matrix H , the transition matrix O , and the T matrix in the Kalman filter equations.

2.9 Sequential Measurement Processing

In situations where the errors in individual measurements are uncorrelated, it can be shown that processing them one at a time gives the same result as processing them as one large block³. That is, the measurement matrix can be reduced to individual rows or any logical group of submatrices and presented to the filter as such.

This has extreme advantages in real-time systems with intermittent asynchronous sensor suites. It allows fairly modular software implementations which adapt in real time to the presence or absence of measurements at any particular time step.

Thus, the software complication involved in restructuring the matrices to accommodate presence or absence of measurements can be completely avoided. The technique has computational advantages as well since inverting two matrices of order $n/2$ is much cheaper than inverting one of order n .

2.10 The Uncertainty Matrices

It is important to distinguish the different roles of the three covariance matrices in the equations. The Q_k matrix models the uncertainty which corrupts the system model. The R_k matrix models the uncertainty associated with the measurement, and its elements are expressed in the coordinate systems and units of the measurements, not the states they measure. For example, the element to

2. Recall that the rank of a matrix is the size of the largest nonzero determinant that can be formed from it. The rank of an $m \times n$ matrix can be no larger than the smaller of m and n . A square $n \times n$ matrix of rank n is called **nonsingular**. The rank of the product of matrices is never larger than the smallest rank of the matrices forming the product.
3. See [7] pp. 264-265.

be entered into R_k for a potentiometer is related to the number of counts of noise on the pot output. Finally, the P_k matrix is largely managed by the filter itself, and it gives the total uncertainty of the state estimate as a function of time.

2.11 Connection to Navigation Theory

Notice that the fourth and fifth filter equations can be used to estimate the state between measurements by integrating over themselves only until a new measurement is obtained. This is the basic mechanism of augmenting **dead reckoning** by a position fix. The system model, whose solution is the second last equation of the filter, can be identified with the process of dead reckoning in navigation theory.

The measurement model can be identified with both the process of measuring velocity and attitude for dead reckoning purposes, or with any mechanism for generating a position fix. The distinction rests on whether the measurements project directly onto the position through the H matrix (**triangulation**) or indirectly through the O matrix (**dead reckoning**).

In particular, when the measurements are the positions of landmarks, the H matrix can be identified with the process which transforms landmark position measurements into vehicle position measurements. Indeed, the "measurement matrix" FL_k is also the **Jacobian** in nonlinear problems and its norm is the well-known **GDOP** or **geometric dilution of precision** from triangulation theory.

2.12 Connection to Smith and Cheeseman

The Kalman filter equations are generalized versions of the "merging" operation discussed in the robotics literature [75] and the state propagation equations are the "compounding" operations of dead reckoning.

Section 3: Linearization of Nonlinear Problems

The filter formulation presented earlier is based on a linear systems model and it is therefore not applicable in situations when either the system model or the measurement relationships are nonlinear. Consider an exact nonlinear model of a system as follows:

$$\begin{aligned}\dot{\bar{\mathbf{x}}} &= \mathbf{f}(\bar{\mathbf{x}},t) + \mathbf{g}(t)\mathbf{W}(t) \\ \bar{\mathbf{z}} &= \mathbf{h}(\bar{\mathbf{x}},t) + \bar{\mathbf{v}}(t)\end{aligned}$$

Where \mathbf{f} and \mathbf{h} are vector nonlinear functions and $\bar{\mathbf{w}}$ and $\bar{\mathbf{v}}$ are white noise processes with zero crosscorrelation. Let the actual trajectory of the system be written in terms of an approximate trajectory $\bar{\mathbf{x}}^*(t)$ and an error trajectory $\mathbf{A}\bar{\mathbf{x}}(t)$ as follows:

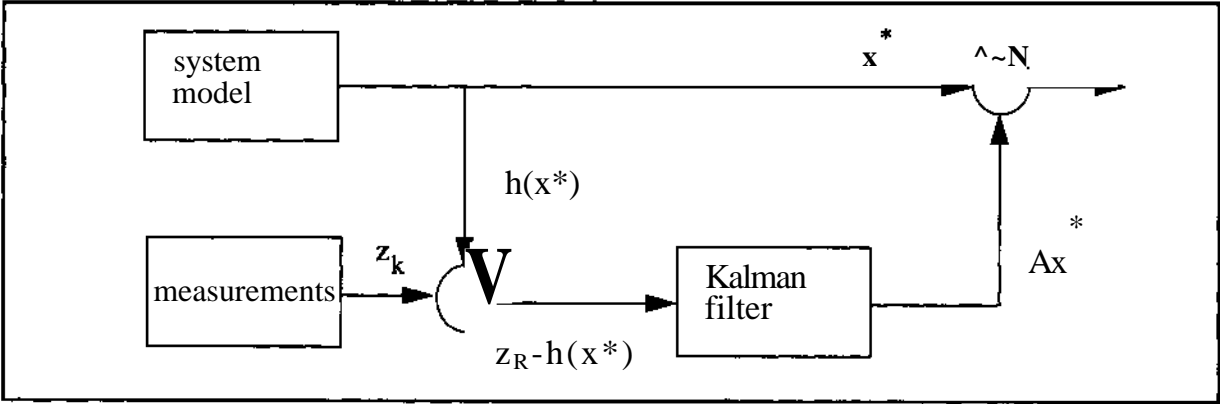
$$\bar{\mathbf{x}}(t) = \bar{\mathbf{x}}^*(t) + \mathbf{A}\bar{\mathbf{x}}(t)$$

By substituting this back into the model and approximating \mathbf{f} and \mathbf{h} by their Jacobians evaluated at the reference trajectory:

$$\begin{aligned}\mathbf{A}\dot{\bar{\mathbf{x}}} &= \frac{d\mathbf{f}}{d\bar{\mathbf{x}}}(\bar{\mathbf{x}}^*,t) \mathbf{A}\bar{\mathbf{x}} + \mathbf{g}(t) \bar{\mathbf{w}}(t) \\ \bar{\mathbf{z}} - \mathbf{h}(\bar{\mathbf{x}}^*,t) &= \frac{d\mathbf{h}}{d\bar{\mathbf{x}}}(\bar{\mathbf{x}}^*,t) \mathbf{A}\bar{\mathbf{x}} + \bar{\mathbf{v}}(t)\end{aligned}$$

3.1 Linearized Kalman Filter

It is clear that this **linearized system model** can be used to implement a **linearized Kalman filter** because the error dynamics and error measurement relationships are linear. The *deviation from the reference trajectory is the state vector* and the measurements are the true measurements less that predicted by the nominal trajectory in the absence of noise. The linearized filter is used in a **feedforward** configuration as shown below:



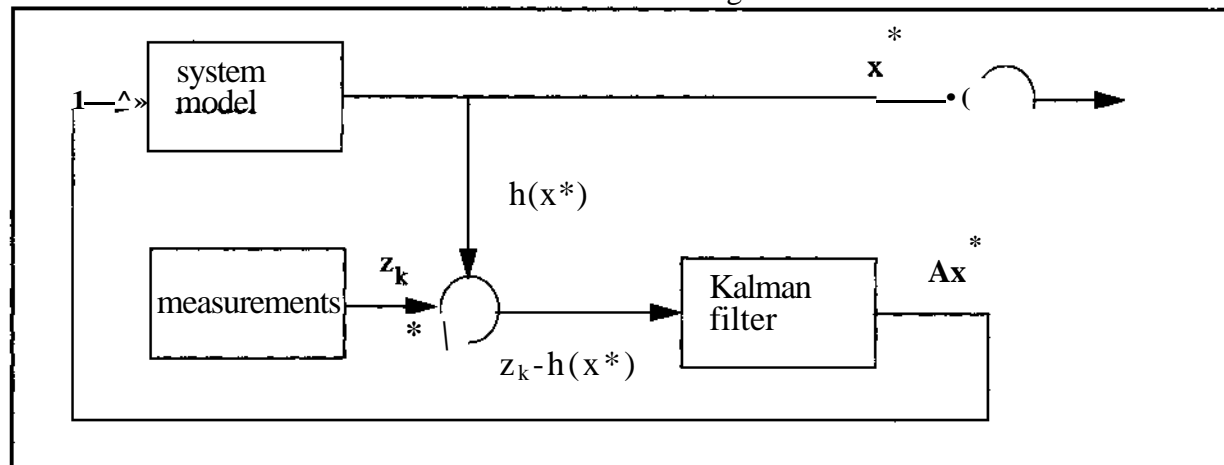
In this form, the nominal trajectory is *not* updated to reflect the error estimates computed by the filter. One of the primary advantages of the linearized filter is that, because it operates exclusively on errors, the unfiltered system model output provides high-fidelity response in the presence of high dynamics. Such a filter is difficult to use for extended missions because, after a time, the reference trajectory may diverge to the point where the linear assumption is no longer valid across the variation in the state vector.

The feedforward model can be used to integrate an INS, or **inertial navigation system**, with a number of navigation aids. The INS is considered to be the system model and its outputs are regarded as the reference trajectory. Measurement aids are used to compute errors and they are applied to the reference to generate the combined output.

3.2 Extended Kalman Filter

In the **extended Kalman filter**, the trajectory error estimates are used to update the reference trajectory as time evolves. This has the advantage that it is more applicable to extended missions. The precise distinction between the two forms of filter is based on the measurement function $h[x]$, that is, whether it is updated based on the *corrected* (extended filter) or the *nominal* (linearized filter) trajectory.

The extended filter can be visualized in a **feedback** configuration as shown below:



In the case of the extended Kalman filter, it is possible to formulate the filter in terms of the state variables themselves rather than the error states. This can be seen as follows. The linearized measurement relationships are:

$$\bar{z} - h(\bar{x}^*, t) = \frac{\partial h}{\partial \bar{x}}(\bar{x}^*, t) \Delta \bar{x}$$

so that the left-hand side is the "measurement" presented to the filter. The discrete time state update equation for this measurement is:

$$\Delta \hat{x}_k = \Delta \hat{x}_k^- + K_k [\bar{z}_k - h(\bar{x}_k^*) - H_k \Delta \hat{x}_k^-]$$

Now, associating the last two terms in the brackets rather than the first two, define the **predictive measurement** to be the sum of the ideal measurement of the reference state and the deviation given by the Jacobian of $h(\bar{x}, t)$.

$$\hat{z}_k^- = h\left(\bar{x}_k^-\right) + H_k A \hat{x}_k^- = h\left(\hat{x}_k^-\right)$$

Then the **measurement residual** is the difference between the predictive measurement and the actual measurement:

$$\Delta z_k = z_k - \hat{z}_k^-$$

Under this substitution, the state update equation can be written as:

$$\hat{x}_k = \hat{x}_k^- + K_k \left[z_k - \hat{z}_k^- \right]$$

The discrete time extended Kalman filter equations for the system model are now as follows:

EXTENDED KALMAN FILTER (DISCRETE TIME)

$\dot{X} = f(\bar{x}, t) + g(t)\bar{w}(t)$ system model
 $E\left[\bar{w}_k \bar{w}_k^T\right] = \Lambda_k$

$\bar{z} = h(\bar{x}, t) + \bar{v}(t)$ measurement model
 $E\left(\bar{v}_k \bar{v}_k^T\right) = R_k$

$J_k^T = \frac{dh(\hat{x}_k^-)}{d\bar{x}_k}$

compute measurement Jacobian

$F_k = \frac{df(\hat{x}_k^-)}{d\bar{x}_k}$

compute system Jacobian

$K_k = P_k^T H_k^T \left(L_k^T \Lambda_k^{-1} L_k + R_k^{-1} \right)^{-1}$

compute Kalman gain

$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - h(\hat{x}_k^-) \right)$

update state estimate

$P_k = \Lambda \sim \Lambda_k \Lambda_k^{-1} \Lambda_k^-$

update its covariance

$\hat{x}_{k+1}^- = (f)_k \hat{x}_k$ see section 3.4.3

project ahead

$P_{k+1}^- = \Phi_F P_k \Phi_F^T + \Gamma_k Q_k \Gamma_k^T$

where the usual conversion to the discrete time model has been performed.

3.3 Taylor Remainder Theorem

One of the most useful approximation tools in all of calculus is the **Taylor remainder theorem**. It quantifies the cost of truncating a power series in terms of lost accuracy of the approximation. For reference, the theorem is:

$$f(x + \Delta x) = f(x) + f'(x) \Delta x + \frac{f''(x) \Delta x^2}{2!} + \sum_{i=3}^n \frac{f^{(i)}(x) \Delta x^i}{i!} + R_{n+1}(x)$$

$$R_{n+1}(x) = \frac{f^{(n+1)}(x^*) (\Delta x)^{n+1}}{(n+1)!}$$

Notice that, although the series is not infinite, the equality is strict if the remainder is evaluated at some point x^* between x and $x + \Delta x$. The practical significance of this extraordinarily useful result is that $|x - x^*|$ is bounded, and the derivative is also bounded, so even though x is not known precisely, it is possible to compute the maximum value of the truncation remainder. This is the formal basis of all linearization.

This means that it is perfectly legitimate to invoke the low dynamics assumption, and an estimate of the highest value of the first neglected derivative estimates the modelling error incurred by the assumption. The low dynamics assumption will be used to estimate the system model of a nonlinear plant and the remainder theorem will be used to estimate the error involved in doing this.

3.4 Forms of Linearization

It is very important to distinguish linearization across time and linearization across the states in any form of linearized Kalman filter. One leads to the transition matrix and the other to the system Jacobian. Four totally different cases of the linear assumption have been encountered thus far.

3.4.1 First Order Statistics

The Kalman filter itself is derived based on the assumption that the noises are Gaussian white sequences/The Gaussian assumption amounts to assuming that higher moments of the probability distributions are all zero. For this reason, the algorithm uses a covariance matrix model of uncertainty. In reality this amounts to *approximating the probability distributions*.

3.4.2 Low Dynamics Assumption

The low dynamics assumption amounts to assuming that the dynamics matrix $F(t)$ is constant. This is a trivial kind of linearization across time where a zeroth order Taylor series approximation of the true answer is used. The true answer is:

$$F(t + \Delta t) = F(t) + \dot{F}(t) \Delta t + \frac{\ddot{F}(t) (\Delta t)^2}{2!} + \dots$$

Assuming the **dynamics matrix** F to be constant is *linearizing in time*. It is often the case in practice that the mere conversion from time-continuous to time-discrete form involves this approximation, although there are ways to avoid it.

3.4.3 Quasilinearization of a Differential Equation

Notice that the transition matrix is, strictly speaking, not defined at all when the system model is nonlinear in the states. In this context, the transition matrix can be thought of as an expression of whatever technique is used to solve the original nonlinear system differential equation, which is a subject in itself, outside of estimation theory. See the linear systems literature, for example, [7] for what little is known about this matter. For a nonlinear plant, this amounts to *approximating the system model*.

3.4.4 Linearized Filters

A separate matter dealt with in estimation theory is the propagation of uncertainty when the plant is nonlinear. Thus the linearized and extended Kalman filters incorporate an assumption of *linearization across states* and, ultimately, the transition matrix and the system Jacobian both appear in the formulation.

The linearized and extended Kalman filters are **first order filters**, thus:

$$F_k = \frac{\partial f}{\partial \bar{x}} \left(\hat{x}_k^- \right)$$

The **higher order filter** is a generalization of the EKF where a higher order Taylor series is used to approximate the nonlinearities. All of these filters are approximations, so it is pointless to argue their merits without getting quantitative about neglected terms.

When the whole list of approximations used in a particular real model are tallied, it is clear that the Kalman filter is a mathematical idealization, which happens to be useful in practice, but which can only be used in practice after some strong assumptions are made.

3.5 State Transition for Nonlinear Problems

When the system differential equation is nonlinear:

$$\dot{x} = f(\bar{x}(t), t)$$

the state propagation equation is written as:

$$\hat{x}_{k+1}^- = \Phi_k \hat{x}_k$$

the transition matrix comes from time linearization because:

$$\bar{x}_{k+1} = \bar{x}_k + \dot{\bar{x}}_k \Delta t$$

$$\bar{x}_{k+1} = \bar{x}_k + f(\bar{x}_k, t_k) \Delta t$$

Therefore, the transition matrix can be written by inspection as an identity matrix with linear and angular velocity cross terms added which are multiplied by Δt .

3.6 System Jacobian for Nonlinear Problems

The system Jacobian for the EKF comes from state space linearization:

$$F = \frac{\partial}{\partial \bar{x}} f(\bar{x}, t)$$

3.7 Uncertainty Propagation for Nonlinear Problems

In the nonlinear case, the uncertainty propagation equation is:

$$P_{k+1}^- = \Phi_F P_k \Phi_F^T + \Gamma_k Q_k \Gamma_k^T$$

The matrix Φ_F is the transition matrix associated with the system Jacobian. It can be computed from time linearization of the continuous time uncertainty propagation equation for a nonlinear problem:

$$\dot{P} = FP + PF^T + GQG^T$$

The last term can be ignored for now. Then, in discrete time, the first two terms can be written as:

$$P_{k+1} = P_k + F P_k \Delta t + P_k F^T \Delta t$$

Define:

$$\Phi_F = (I + F) \Delta t$$

Then, notice that the following expression is correct to first order:

$$P_{k+1} = \Phi_F P_k \Phi_F^T = (I + F \Delta t) P_k (I + F \Delta t)^T$$

$$P_{k+1} = (P_k + P_k F \Delta t) \left(I + F^T \Delta t \right) = P_k + F P_k \Delta t + P_k F^T \Delta t + F P_k F^T (\Delta t)^2$$

So the following equations are correct to first order and may be used in implementation:

$$P_{k+1}^- = \Phi_F P_k \Phi_F^T + \Gamma_k Q_k \Gamma_k^T$$

$$\Phi_F = (I + F) \Delta t$$

3.8 Other Kalman Filter Algorithms

It is worth noting briefly that there are other forms of the Kalman filter. Two such forms may be applicable in special cases. The **correlated measurement and process noise** filter incorporates a model of the correlation between disturbances and measurement noise. It is also possible to readily model **deterministic inputs** in a Kalman filter. These often correspond to the command signals in automatic control applications.

3.9 The Measurement Conceptualization in the EKF

It is important to recognize that the *measurement process itself* is used in the *extended* Kalman filter, and the computation of the deviation from predicted to actual measurement is automatic in the formalism. More specifically, the state update equation is:

$$\hat{x}_k = \hat{x}_k^- + K_k [z_k - h(\hat{x}_k^-)] \quad \text{update state estimate}$$

and this contains the computation of the predicted measurement $h(\hat{x}_k^-)$ already. The predicted measurement is computed inside the filter itself.

Section 4: State Vector Augmentation

One of the secrets to high performance navigation systems is the mechanism by which they utilize the Kalman filter to **identify**⁴ themselves in real time. In the language of estimation theory, the mechanism is known as **state vector augmentation**. It is intimately related to the idea that, although the Kalman filter requires white noise processes, unknown system parameters and nonwhite noise sources can be modelled as the result of passing a white noise process through a system with linear dynamics.

The Kalman filter is very sensitive to the exact form of the model chosen. In particular, it requires that the errors and disturbances be precisely zero mean, Gaussian, white sequences. More often than not, the driving functions are not white and they must be modelled by differential equations which relate them to fictitious white noise processes. This process of creating additional state variables allows the filter to account for measurement noise and disturbances that are not white.

While vendors often like to quote the number of states in their filter algorithms, it is often the **choice** of states that matters, and not their number. Indeed, filter performance often degrades in practice when too many states are used. For example, very much is known about the propagation of errors in inertial navigation systems, and intimate knowledge of the system dynamics, and the likely sources of error, and their dynamics is necessary for optimal filter performance.

In situations when disturbances and measurement errors are changing very slowly relative to the system state or measurements, the error sources themselves can be estimated and added to the system state vector. In this way, the filter estimates the system parameters as well as state. This is known as **state vector augmentation**. Additional states are assumed to follow a **correlation model** which is based upon knowledge of the underlying physics of the system.

4.1 Principle

Suppose the measurement noise \bar{v} in the continuous time model:

$$\begin{aligned}\dot{\bar{x}} &= F\bar{x} + G\bar{w} \\ \bar{z} &= H\bar{x} + \bar{v}\end{aligned}$$

is correlated. Oftentimes it is possible to consider that the correlated measurement noise arises through passing uncorrelated white noise \bar{w}_1 through a system with linear dynamics (i.e by filtering it⁵) thus:

$$\dot{\bar{v}} = E\bar{v} + \bar{w}_1$$

4. In linear system theory, “identification” is the term used for calibration of system parameters.
5. A first order differential equation is also the defining equation of a first order filter. There is a direct equivalence between the “memory” of a dynamic system which arises from its expression as a differential equation, and its ability to operate as a filter. They are the same animal under two different names.

Using this model, the correlated component of noise can be considered to constitute a random element in the state vector. This element is added to the existing states to form the augmented state vector:

$$\frac{d}{dt} \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{v}} \end{bmatrix} + \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{w}} \\ \bar{\mathbf{w}}_1 \end{bmatrix}$$

Then the measurement equation becomes:

$$\bar{\mathbf{z}} = [\mathbf{H} \mathbf{i}] \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{v}} \end{bmatrix}$$

With this reformulation, the measurement noise has disappeared completely because it has been absorbed into the system model. Under these circumstances, the Kalman filter equations used in this document are *necessary* to address the singularity of the R_k matrix.

It is also just as easy to model deterministic new state variables with the technique and these can be used to calibrate a system. For example, the random constant model described in the next section can be used to define an unknown constant, provided the measurement matrix H is updated to reflect the new state. The basic operation of the filter is to project the measurement residual onto the states, so it will determine a value for this constant and even allow it to vary slowly with time.

4.2 Correlation Models

Some important **correlation models** are as follows:

4.2.1 Random Walk

The **random walk** is also known as the **Weiner process** or **Brownian motion**. Essentially, this is the integral of a random Gaussian noise signal. That is to say, the position of a drunk when each new step is of random size either forward or backward. The state differential equation is:

$$\dot{\mathbf{x}}_1 = \mathbf{w}$$

where w is a random signal.

4.2.2 Random Constant

Also called **random bias**. It is a constant value which varies randomly from one experiment to the next, but which is both constant over time and unknown for any particular experiment. It is important for modelling system biases. The state differential equation is:

$$\dot{\mathbf{x}}_1 = \mathbf{0}$$

4.2.3 Random Harmonic

This is a sinusoid of random amplitude and phase. The state differential equation is:

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{x}_{i+1} + \mathbf{w} \\ \mathbf{x}_{i+1} &= -\omega^2 \mathbf{x}_i - 2\beta \mathbf{x}_{i+1} + (\mathbf{b} - 2\mathbf{a}\beta) \mathbf{w}\end{aligned}$$

4.2.4 Random Ramp

This is a process which grows linearly with time but has a random initial value and slope. It is important for modelling system scale errors. The state differential equation is:

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{x}_{i+1} \\ \mathbf{x}_{i+1} &= \mathbf{0}\end{aligned}$$

4.2.5 Gauss Markov Process

The **Gauss Markov process**, or first order Markov process, is one which has an exponentially decreasing autocorrelation function. In practical terms, this amounts to an assumption that the process exhibits little correlation between values which are sufficiently well separated in time. It is often used to represent errors about which little is known other than that they are bandlimited. The state differential equation is:

$$\dot{\mathbf{x}}_1 = -\mathbf{a} \mathbf{x} + \mathbf{W}$$

When the estimation period is much smaller than the time constant of this process, it is well approximated by the random walk.

The random walk, constant, and ramp can be modelled together by only two state variables. The number of states for any correlation model depends on the order of the underlying differential equation and not on the number of parameters.

PART X: Bibliography

A

[I] O. Amidi, "Integrated Mobile Robot Control", Robotics Institute Technical Report CMU-RI-TR-90-17, Carnegie Mellon University, 1990.

B

[2] M. G. Bekker, "Off-the-Road Locomotion", The University of Michigan Press, 1960.
[3] M. G. Bekker, "The Theory of Land Locomotion", The University of Michigan Press, 1956.
[4] R. Bhatt, L. Venetsky, D. Gaw, D. Lowing, A. Meystel, "A Real-Time Pilot for an Autonomous Robot", Proceedings of IEEE Conference on Intelligent Control, 1987.
[5] B.E. Bona and R. J. Smay, "Optimum reset of SINS", IEEE Trans on Aerospace and Electronic Systems, AES-2, No 4, 409-414 (July 1966).
[6] J. E. Bresenham, "Algorithm for Computer Control of a Digital Plotter", IBM Systems Journal, 4(1), 1965,25-30.
[7] W. L. Brogan, "Modern Control Theory", Quantum Publishers, 1974
[8] R. A. Brooks, "A Hardware Retargetable Distributed Layered Architecture for Mobile Robot Control", Proceedings of IEEE International Conference on Robotics and Automation, 1987.
[9] R. G. Brown and P. Y. C. Hwang, "Introduction to Random Signals and Applied Kalman Filtering", Wiley, 1992.
[10] B. Brumitt, R. C. Coulter, A. Stent, "Dynamic Trajectory Planning for a Cross-Country Navigator", Proceedings of the SPIE Conference on Mobile Robots, 1992.

C

II1] T. S. Chang, K. Qui, and J. J. Nitao, "An Obstacle Avoidance Algorithm for an Autonomous Land Vehicle", Proceedings of the 1986 SPIE Conference on Mobile Robots, pp. 117-123.
[12] J. Craig, "Introduction to Robotics, Mechanics and Control", Addison Wesley, 1986.

D

[13] M. Daily, et al. "Autonomous Cross-Country Navigation with the ALV", Proceedings of the 1988 IEEE International Conference on Robotics and Automation, pp. 718-726.
[14] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," J. Applied Mechanics, pp 215-221, June 1955.
[15] E. D. Dickmanns, A. Zapp, "A Curvature-Based Scheme for Improving Road Vehicle Guidance by Computer Vision", Proceedings of the SPIE Conference on Mobile Robots, 1986.
[16] E. D. Dickmanns, "Dynamic Computer Vision for Mobile Robot Control", Proceedings of the 19th International Symposium and Exposition on Robots, pp. 314-27.

[17] J. C. Dixon, "Linear and Non-Linear Steady State Vehicle Handling", Proceedings of Instn. Mechanical Engineers, Vol. 202, No. D3,1988.

[18] R. T. Dunlay, D. G. Morgenthaler, "Obstacle Avoidance on Roadways Using Range Data", Proceedings of SPIE Conference on Mobile Robots, 1986.

E

F

[19] J. A. Farrel, "Integrated Aircraft Navigation", Academic Press, 1976.

[20] W. Feiten and R. Bauer, "Robust Obstacle Avoidance in Unknown and Cramped Environments", Proc. IEEE Intl. Conference on Robotics and Automation, May 1994, San Diego.

[21] D. Feng, "Satisficing Feedback Strategies for Local Navigation of Autonomous Mobile Robots", Ph.D. Dissertation at CMU, May, 1989.

[22] D. Feng, S. Singh, B. Krogh, "Implementation of Dynamic Obstacle Avoidance on the CMU Navlab", Proceedings of IEEE Conference on Systems Engineering, August, 1990.

[23] M. Fernandez & G. R. MacComber, "Inertial Guidance Engineering", Prentice Hall, Englewood Cliffs, NJ, 1962.

[24] R. E. Fikes & N. J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving", Artificial Intelligence, Vol 2, pp. 189-208 (1971).

[25] U. Franke, H. Fritz, S. Mehring, "Long Distance Driving with the Daimler-Benz Autonomous Vehicle VITA", PROMETHEUS Workshop, Grenoble, December, 1991.

G

[26] K. G. Gauss, "Theory of Motion of the Heavenly Bodies", New York, Dover, 1963.

[27] A. Gelb editor, "Applied Optimal Estimation", MIT Press.

[28] A. Gelb, "Synthesis of a Very Accurate Inertial Navigation System", IEEE Trans on Aerospace and Navigational Electronics, Vol. 12, June 1965, pp 119-128.

[29] J. Gowdy, A. Stentz, and M. Hebert, "Hierarchical Terrain Representation for Off-Road Navigation", In Proc SPIE Mobile Robots, 1990.

H

[30] M. Hebert and E. Krotkov, "Imaging Laser Radars: How Good Are They", IROS 91, November 91.

[31] M. Hebert, T. Kanade, and I. Kweon. "3-D Vision Techniques for Autonomous Vehicles", Technical Report CMU-RI-TR-88-12, The Robotics Institute, Carnegie Mellon University, 1988.

[32] M. Hebert, "Building and Navigating Maps of Road Scenes Using an Active Sensor", In Proceedings IEEE conference on Robotics & Automation, 1989; pp. 36-42.

[33] R. Hoffman, E. Krotkov, "Terrain Mapping for Outdoor Robots: Robust Perception for Walking in the Grass", Submitted to IEEE International Conference on Robotics and Automation, 1993.

[34] B. K. Horn and J. G. Harris, "Rigid Body Motion from Range Image Sequences", Image Understanding, Vol. 53, No. 1, January 1991, pp 1-13.

[35] B. Hotz, Z. Zhang, P. Fua, "Incremental Construction of Local DEM for an Autonomous Planetary Rover", Workshop on Computer Vision for Space Applications - Antibes, Sept. 22-24, 1993.

I

J

K

[36] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory", Trans of the ASME - J of Basic Engr., 95-108, March 1961.

[37] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Trans, of the ASME - J of Basic Engr., 35-45, March 1960.

[38] D. Keirsey, D. Payton, J. Rosenblatt, "Autonomous Navigation in Cross-Country Terrain", proceedings of Image Understanding Workshop, 1988.

[39] A. J. Kelly, "A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-19.

[40] A. J. Kelly, "A Feedforward Control Approach to the Local Navigation Problem for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-17.

[41] A. J. Kelly, "A Partial Analysis of the High-Speed Autonomous Navigation Problem", CMU Robotics Institute Technical Report CMU-RI-TR-94-16.

[42] A. J. Kelly, "Adaptive Perception for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-18.

[43] A. J. Kelly, "An Intelligent Predictive Controller for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-20.

[44] A. J. Kelly, "Concept Design of A Scanning Laser Rangefinder for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-21.

[45] A. J. Kelly, "Essential Kinematics for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-14.

[46] A. J. Kelly, "Modern Inertial and Satellite Navigation Systems", CMU Robotics Institute Technical Report CMU-RI-TR-94-15.

[47] A. Kelly, A. Stentz, M. Hebert, "Terrain Map Building for Feist Navigation on Rough Terrain", Proceedings of the SPIE Conference on Mobile Robots, 1992.

[48] E. Krotkov, and M. Hebert, "Mapping and Positioning for a Prototype Lunar Rover", proceedings of the International Conference on Robotics and Automation, May 1995.

[49] A. M. Keuthe and C. Y. Chow, Foundations of Aerodynamics, Wiley 1950.

[50] P. R. Klarer, J. W. Purvis, "A Highly Agile Mobility Chassis Design for a Robotic All-Terrain Lunar Exploration Rover."

[51] R. E. Korf, "Real Time Heuristic Search: First Results", Proc. Sixth National Conference on AI, July 1987.

[52] In So Kweon, "Modelling Rugged Terrain by Mobile Robots with Multiple Sensors", CMU Ph.D. Thesis, 1990.

L

[53] D. Langer, J. K. Rosenblatt, M. Hebert, "A Reactive System For Off-Road Navigation", CMU Tech Report

[54] J. C. Latombe, "Robot Motion Planning", Kluwer Academic Publishers, 1991.

[55] T. Lozano-Perez, and M. A. Wesley, "An Algorithm for Planning Collision Free Paths Among Polyhedral Obstacles". Communications of the ACM, Vol. 22, Num. 10, October 1979, pp. 560-570.

M

- [56] M. Marra, R. T. Dunlay, D. Mathis, "Terrain Classification Using Texture for the ALV", Proceedings of SPIE Conference on Mobile Robots, 1988.
- [57] L. Matthies, "Stereo Vision for Planetary Rovers", International Journal of Computer Vision, 8:1, 71-91, 1992.
- [58] L. Matthies, S. Shafer, "Error Modelling in Stereo Navigation", IEE Journal of Robotics and Automation, Vol. RA-3, No 3, June 1987.
- [59] L. Matthies, A. Kelly, T. Litwin, "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report," to appear in the Proceedings of the IEEE International Symposium on Intelligent Vehicles.
- [60] P. S. Maybeck, "Stochastic Models, Estimation, and Control", Academic Press, 1982.
- [61] L. S. McTamane, "Mobile Robots: Real-Time Intelligent Control", IEEE Expert, Vol. 2, No. 4, Winter, 1987.
- [62] H. P. Moravec, "The Stanford Cart and the CMU Rover", Proceedings of the IEEE, Vol. 71, No. 7, July 1983, pp. 872-884.
- [63] Military Handbook: Transformation of Datums, Projections, Grids, and Common Corodinate Systems, MIL-HDBK-600008.

N

O

- [64] K. Olin, and D. Tseng, "Autonomous Cross-Country Navigation", IEEE Expert, August 1991, pp. 16-30.
- [65] M. Okutomi, T. Kanade, "A Locally Adaptive Window for Signal Matching", in Proc ICCV, Dec 1990.
- [66] M. Okutomi, T. Kanade, "A Multiple Baseline Stereo", IEEE PAMI, Vol. 15, No. 4, April 1993.

P

- [67] D. Pieper and B. Roth, "The Kinematics of Manipulators Under Computer Control", Ph.D. Thesis, Stanford University, 1968.
- [68] D. A. Pomerleau, "Efficient Training of Artificial Neural Networks for Autonomous Navigation", Neural Computation, Vol. 3, No. 1, 1991, pp. 88-97.
- [69] D. A. Pomerleau, "Neural Network Perception for Mobile Robot Guidance", CMU tect report CMU-CS-92-115.

Q

R

- [70] J. Rosenblatt, D. Payton, "A Fine-Grained Alternative to the Subsumption Architecture", Proceedings of the AAAI Symposium on Robot Navigation, March, 1989.
- [71] B. Ross, "A Practical Stereo Vision System", in proc CVPR 93.

S

- [72] D. H. Shin, S. Singh and Wenfan Shi. "A Partitioned Control Scheme for Mobile Robot Path Planning", Proceedings IEEE Conference on Systems Engineering, Dayton, Ohio, August 1991.

[73] S. Shafer, A. Stentz, C. Thorpe, “An Architecture for Sensor Fusion in a Mobile Robot”, Proceedings of IEEE International Conference on Robotics and Automation”, April, 1986.

[74] S. Singh, et. al, “FastNav: A System for Fast Navigation”, Robotics Institute Technical Report CMU-RI-TR-91-20, Carnegie Mellon University, 1991.

[75] R. C. Smith and P. Cheeseman, “On the Representation and Estimation of Spatial Uncertainty”, Int’l J of Robotics Research, Vol. 5, No. 4, Winter 1986.

[76] A. Stentz, “The NAVLAB System for Mobile Robot Navigation”, CMU Ph. D. thesis, CMU-CS-90-123.

[77] A. Stentz, “Optimal and Efficient Path Planning for Unknown and Dynamic Environments”, CMU Tech report CMU-RI-TR-93-20.

[78] H. C. Stone, “Design and Control of the Mesur/Pathfinder Microrover”, Jet Propulsion Laboratory

[79] A. A. Sullivan & J. A. Gelb, “Application of the Kalman Filter to Aided Inertial Systems”, Analytical Sciences Corporation.

T

[80] A. Thompson, “The Navigation System of the JPL Robot”, Proceedings of the International Joint Conference for Artificial Intelligence, 1977, pp. 749-757.

U

V

W

[81] P. J. Ward and S. J. Mellor, “Structured Development for Real-Time Systems”, Yourdon Press.

[82] B. Wilcox, et al. “A Vision System for a Mars Rover. SPIE Mobile Robots II”, November 1987, Cambridge Mass., pp. 172-179.

Index

Numerics

2-1/2 D world assumption..... 342

A

absolute observations.....303
accuracy.....16
Ackerman steering.....384
actuation space.....31, 112, 152, 226, 230, 269, 270
actuator contention problem.....226
acuity.....17
acuity adaptive planning.....23
acuity adaptive scan.....23
acuity problem.....22, 69, 141
acuity ratio.....148
acuity ratios.....22, 67
adaptive lookahead.....47, 60, 124,182, 183, 186, 217, 218
adaptive perception.....182, 186, 217, 218
adaptive regard.....46,47, 121, 144, 146,224,268
adaptive scan.....102, 124, 131, 149, 153, 182, 185, 186, 217, 218
adaptive sweep.....60, 102, 124, 129, 147, 149, 153, 182, 186, 191, 217, 218, 219
AHRS.....278,329
Arbiter.....260
attitude acuity rule.....116
autocorrelation function.....426
azimuth.....69
azimuth scanner.....392

B

bandwidth-conceptualization gap.....159, 163, 178
beam dispersion.....69
benign terrain assumption.....25, 63
bias.....318
bias stability.....318
braking angle.....36,43
braking coefficient.....34, 144
braking distance.....31, 33, 34
braking reaction time.....30
Brownian motion.....447

C

cell aging feature.....206
characteristic time.....106
clothoid.....107, 112
clothoid generation.....230
clothoid generation problem.....229, 270
coefficient of lateral acceleration.....36

command following problem.....	119, 152, 160, 178
Command Generator.....	257
command vector.....	226, 227, 240
communication latency.....	103
communications load.....	65
complete subsampling.....	195
completeness.....	157
computational spiral effect.....	137, 153, 340
computational stabilization.....	129
configuration space.....	112, 228, 230, 269, 270, 361
constant flux.....	64
constant scan.....	64
continuity assumption.....	342
control laws.....	256,273
control layer.....	157, 177
control time.....	161
controllability.....	157
Controller.....	166, 180
coordinate frame.....	357
correlated.....	426
correlated disparity assumption.....	80, 89
correlated measurement and process noise.....	445
correlation.....	303
correlation curve.....	79, 190
correlation model.....	446
correlation models.....	447
correlation tensor.....	79, 190
correspondence problem.....	77
cross power spectral density.....	427
crossrange.....	69
crosstrack vector.....	300
C-space.....	228,269
curvature.....	285
D	
data flow diagram.....	165
datalogger.....	169
dead reckoning.....	438
dead zone.....	45, 145
deduced reckoning.....	112
degrees of freedom.....	362
deliberation.....	229
deliberative.....	177
depression.....	69
detection zone.....	45, 145, 182, 217, 224, 249, 268, 272
detection zone tracker.....	253
deterministic inputs.....	445

deviation	286, 296
differential observations	303
disparity	77
disparity / correlation coupling	79
disparity gradient	149
disparity image	190
disparity window	185, 218
disturbance sequence	432
downrange	69
dynamic braking coefficient	34
dynamic braking regime	35
dynamic steering limit	37
dynamic turning coefficient	42
dynamic turning regime	42
dynamically feasible	119, 152
dynamics constraint	119, 152
dynamics matrix	442
dynamics reaction time	28

E

elevation	69
elevation angle window	184
elevation scanner	402
ensemble	425
epipolar geometry	412
epipolar geometry assumption	85
epipolar geometry constraint	77
ergodic	426
Euler angles	382
execution monitoring	159
extended Kalman filter	440

F

feasible set	228, 229, 269
feature based	303
feedback	440
feedback control	119
feedforward	119, 120, 152, 229, 270, 439
fidelity	17, 103
fidelity adaptive planning	25
fidelity problem	24, 41, 117, 119, 141
fidelity ratio	150
fidelity ratios	24
filter	105
first order filters	443
fix	438
flat world assumption	25, 61, 67, 75, 188
focal length	77

focus of attention.....	182, 217
forward model.....	227, 269
forward modelling.....	152
frame.....	357
free zones.....	45
Fresnel Integrals.....	112
fundamental tradeoff.....	136
G	
Gauss Markov process.....	448
Gaussian filtering.....	243, 260
GDOP.....	438
generate and test.....	156
geometric decorrelation.....	80, 86, 196
geometric dilution of precision.....	438
geometric efficiency.....	101, 102, 149
goal vector.....	300
guaranteed detection.....	22, 126
guaranteed localization.....	24
guaranteed response.....	18, 126, 139
guaranteed safety.....	17, 63, 340
guaranteed throughput.....	20
H	
halo effect.....	87
hardware reaction time.....	28
hazard space.....	226,231
hazard vector.....	226,227
Heeling error.....	296
high throughput assumption.....	21
higher order filter.....	443
hill occlusion rule.....	61
hole occlusion rule.....	62
homogeneous coordinates.....	353
I	
iconic.....	303
identify.....	446
image registration problem.....	25, 117, 141
imaging density.....	60, 100, 102, 130, 149
impulse turn.....	31, 144
impulse turn maneuver.....	39
impulse turning distance.....	40
incremental lookahead angle.....	47
incremental lookahead distance.....	46
inertial navigation system.....	440
instantaneous field of view.....	69
integrated random walk.....	429
intelligent autonomous mobility.....	334

intensity crosstalk.....	87
interpolation.....	69
inverse model.....	227, 269
J	
Jacobian.....	438
Jacobians.....	372
K	
Kalmangain.....	315, 332
kinematic braking coefficient.....	34
kinematic braking regime.....	35
kinematic steering limit.....	37
kinematic turning coefficient.....	42
kinematic turning regime.....	42
L	
landmark Jacobian.....	307
latency problem.....	19,30, 111, 119, 140, 151
lateral occlusion problem.....	63
linearized Kalman filter.....	439
linearized system model.....	439
local correlation maximum problem.....	192
local minimum problem.....	225, 269
local planning problem.....	163, 179
localization.....	118
locally-planar terrain assumption.....	236
logistic layer.....	157
longitudinal aspect ratio.....	26
lookahead.....	161
lookahead angle.....	46,47
lookahead distance.....	47, 259, 299
lookahead ratio.....	47, 128
lookahead vector.....	300
low dynamics assumption.....	25, 106, 240, 328, 433
low latency assumption.....	19, 106
M	
main event loop.....	171
Map Manager.....	166, 180
map matching.....	309
map resolution.....	24
mapping.....	309
maximal visual similarity assumption.....	79, 89
maximum acuity.....	141
maximum acuity rule.....	23
maximum fidelity.....	104
maximum fidelity ratio.....	104
maximum fidelity rule.....	25

maximum sensor acuity rule.....	74
measurement.....	432
measurement gradient.....	315, 331
measurement matrix.....	432
measurement noise sequence.....	432
measurement residual.....	315, 332, 441
mechanism model.....	361
minimum acuity.....	141, 149
minimum acuity rule.....	22
minimum fidelity rule.....	24
minimum potential energy principle.....	156
minimum sensor acuity rule.....	73
minimum significant delay.....	104, 150
mission analysis.....	156
modelling.....	275,328
modelling problem.....	119,275
monotone arc length assumption.....	247
monotone range assumption.....	184, 188, 343
motion distortion problem.....	23, 93, 141
multivariable state space system.....	227
myopia problem.....	19, 75, 140, 146
N	
negative obstacle.....	62
nonholonomic.....	112
nonholonomy problem.....	111, 151
normalized baseline.....	82, 85, 420
normalized correlation.....	79
normalized disparity.....	77, 191
normalized incremental lookahead distance.....	47
normalized map resolution.....	99
normalized range difference.....	102
normalized time constant.....	106, 109, 151
normalized undercarriage clearance.....	26
normalized wheel radius.....	26
normalized wheelbase.....	26
Nyquist rate.....	161
O	
objective function.....	226
observability.....	157,436
observability problem.....	25, 141
observation matrix.....	432
obstacle sparsity assumption.....	25, 342
occlusion constraint.....	80
occlusion problem.....	61, 242
on demand interpolation.....	262
onto.....	200

operator.....	352
optimality.....	157
output vector.....	227
oversampling.....	69
oversampling factor.....	23, 67
P	
panic stop.....	31, 144, 186, 219
panic stop maneuver.....	33
partial subsampling.....	194
path features.....	297
path replanning.....	159
Perception.....	166, 180
perception ratio.....	26,61,68, 102, 142, 148, 194
perceptual software efficiency.....	65
perspective.....	412
piloting mission.....	1
pitch.....	378
pitch regulator.....	254
pixel processing load.....	65
planning angle.....	36, 43
planning window.....	186, 249, 272
plant dynamics.....	103
policy layer.....	156
pose.....	373
Position Estimator.....	166, 180
positive obstacle.....	61
potential field.....	159
power spectral density.....	427
predictive measurement.....	315, 331, 441
principle motion assumption.....	282, 292, 328, 329
principle of characteristic times.....	156, 159
principle of optimality.....	157
process noise distribution matrix.....	432
process noise sequence.....	432
processing latency.....	103
processing reaction	time.....27
processor load.....	65
projection plane.....	386
pure pursuit.....	233,259
R	
random bias.....	447
random process.....	425
random walk.....	429,447
range gating.....	185
range image.....	190
range rate.....	285

range ratio.....	47, 71, 102
range window.....	183, 218, 249, 272
range window tracker.....	254
reaction distance.....	30, 33, 34, 40, 144, 152
reaction time.....	16
reactive.....	177
reactive ratio.....	28, 138, 144
real time kernel.....	165
real-time.....	144
real-time approach.....	153
redundancy.....	366
reference points.....	236, 271
reflection plane.....	385
region of interest.....	182,217
relative accuracy.....	24
relative crosstrack error.....	299
repetitive texture problem.....	79
resolution.....	16, 67
response adapted lookahead rule.....	19
response adapted speed.....	19
response adaptive lookahead.....	19
response adaptive speed.....	19
response problem.....	18, 140
response ratio.....	18, 27, 45, 111, 143
reverse turn.....	31, 41, 53, 144, 186, 219
rigid suspension assumption.....	116, 236
rigid terrain assumption.....	116, 236
roll.....	378

S

safe terrain assumption.....	17"
sampling problem.....	23, 69, 141, 149, 242
sampling property.....	88
sampling theorem.....	67, 158
scale factor.....	318
scanning density.....	98, 102, 149
scatter matrix.....	202,221
selection problem.....	183
sensing reaction time.....	27
sensitivity problem.....	25, 141
sensor dwell.....	103
sensor flux.....	64
sensor throughput.....	64
sign correlation.....	79
singularity.....	373
small incidence angle assumption.....	57, 76, 148, 187, 191, 219, 342
small pitch assumption.....	241

soundness.....	157
spectral amplitude.....	428
speed.....	16
stability.....	157
stability problem.....	25, 41, 141
stabilization problem.....	21, 59, 141, 146
stacking.....	311, 313
standard model.....	155, 177, 277
state model.....	432
state space.....	226, 231
state space model.....	227, 269
state space simulator.....	229
state vector.....	226, 227, 240, 432
state vector augmentation.....	446
static stability margin.....	245
stationary.....	426
stationary process.....	427
steady-state regime.....	105
steering regulator.....	253
stereoscopy.....	77
stopping angle.....	36, 43
stopping distance.....	33
stopping region.....	38
Strategic Controller.....	259
strategic goal.....	226, 234, 270
strategic layer.....	157, 177
strategic planning problem.....	1
strategic time.....	161
subpixel disparity estimation.....	81, 149
sum of absolute differences.....	79
sum of squared differences.....	79
survey point.....	297
sweep rate.....	58, 60, 64, 128, 147
sweep rate rule.....	58
system reaction time.....	27, 143

T

tactical control.....	178
Tactical Controller.....	257
tactical goal.....	226, 234, 270
tactical layer.....	158
tactical time.....	161
task space.....	361
Taylor remainder theorem.....	328, 442
temporal planning horizon.....	106
terrain smoothness assumption.....	23, 68, 69, 342
test environment.....	168, 180

thrashing	160
throughput	16
throughput adapted speed	21
throughput adapted sweep	21
throughput adaptive speed	21
throughput adaptive sweep	21
throughput performance limit	126
throughput problem	20, 59, 123, 137, 141, 147, 153, 182, 217
throughput ratio	20, 47, 49, 60, 100, 139, 146
time constant	105
timing	16
transient regime	105
transient turning coefficient	109, 151
transition matrix	432
triangulation	374, 438
tunnel vision problem	21, 141, 146
turning coefficient	42
turning distance	36, 40
turning fidelity ratio	110, 121
turning reaction time	30
turning stop	31, 144, 186, 219
turning stop maneuver	38
U	
unbiased	426
uncorrelated disparity assumption	80
uncorrelated measurement error assumption	290, 330
undercarriage tangent	26, 142
undersampling	69
unforced Kalman Filter	278
uniform scan assumption	23
unknown hazard assumption	63
update rate	161
V	
variation	286, 296
Vehicle	166, 180, 261, 273
vertical	69
W	
wander azimuth	318
Weiner process	447
wheel fraction	26, 74
wheelbase	26
white	426
wide depth of field assumption	19
workspace	366
world tree	171

	Y	
yaw.....		378
yaw rate.....		36
	Z	
zup.....		319

Glossary

The author makes no representation that these definitions are accepted broadly. These are, however, the meanings of the terms as they are used in this document.

Numerics

2-1/2 D world assumption - the assumption that a vertical line aligned with gravity passes through the terrain only once. Trees and overhangs can violate this assumption.

A

- absolute observations** - any sensor observations which are inherently expressed relative to the navigation coordinate system.
- accuracy** - a measure of how close a measurement is to the true value of the quantity measured.
- Ackerman steering** - a steering mechanism, typical of automobiles, where the two front wheels turn together.
- actuation space** - an abstract space consisting of all independent control inputs to a system. For an automobile, this space is spanned by steering, brake, and throttle.
- actuator contention problem** - a specific problem of the behavioral approach to autonomy where independent agents contend for control of actuators. More generally, the contention of software modules for the control of actuators.
- acuity** - a fancy word for resolution. Often used in the context of visual angular resolution.
- acuity adaptive planning** - any planning mechanism which adapts to measured or predicted acuity of available measurements.
- acuity adaptive scan** - a mechanism which adjusts the resolution of image data in order to control groundplane resolution.
- acuity problem** - the problem of maintaining adequate resolution in computations and sensing.
- acuity ratios** - nondimensional ratios of world model resolution and an appropriate dimension of the vehicle.
- adaptive lookahead** - any mechanism which adapts sensory lookahead to the speed and response of the vehicle.
- adaptive perception** - an algorithm based on a focus of attention and a resolution transform which adapts to vehicle speed and response to provide minimum throughput and uniform minimum resolution of perceptual processing.
- adaptive regard** - an algorithm based on a focus of attention and a resolution transform which adapts to vehicle speed and response to provide minimum throughput and uniform minimum resolution of planning processing.
- adaptive scan** - an algorithm which attempts to make perceptual groundplane resolution homogeneous and isotropic.
- adaptive sweep** - an algorithm which projects a focus of attention on the ground-

plane into image space.
AHRS - a device which measures vehicle attitude and heading.
Arbiter - a software module which arbitrates the actuator commands of several agents.
attitude acuity rule - a rule for determining the required angular resolution of attitude and heading measurements.
autocorrelation function - a signal processing technique for measuring how alike two signals are.
azimuth - the rotation of something about a vertical axis.
azimuth scanner - the configuration of a laser scanner where azimuth is the first rotation of the beam.

B

bandwidth-conceptualization gap - the difficulty in communication that occurs between strategic and control layers of the standard model because of different time scales and conceptualizations of the world.
beam dispersion - the angular width of a laser beam.
benign terrain assumption - the assumption that the terrain is not rough.
bias - constant (with respect to input) error in any measuring device.
bias stability - a measure of how constant bias is over time.
braking angle - the angle travelled while the brakes are actuated before coming to a stop.
braking coefficient - the nondimensional ratio of braking distance to reaction distance.
braking distance - the distance travelled while the brake is actuated before coming to a stop.
braking reaction time - the time it takes from image acquisition until the brakes are engaged.
Brownian motion - the motion that results from integrating a white noise velocity.

C

cell aging feature - a feature of the terrain map interface routines that renders old data invisible instead of physically copying memory.
characteristic time - a measure of the delay associated with a system from input to output.
clothoid - a linear polynomial for curvature expressed in terms of arc length. The trajectory corresponding to this polynomial.
clothoid generation problem - the problem of determining the curvature commands to issue to a vehicle to cause it to follow a given spatial trajectory.
coefficient of lateral acceleration - the ratio of lateral acceleration and gravitational acceleration. Equal to the later acceleration in units of g's.
command following problem - the problem of causing a servo-controlled device to follow its commands acceptably well.
Command Generator - the part of the Tactical Controller that generates command alternatives.

command vector - the vector which contains all command inputs to the vehicle or its controller.

communication latency - the latency associated with communications.

communications load - the rate at which information must be transmitted.

complete subsampling - an approach to adaptive scan for stereo vision that homogenizes resolution in all stereo processing.

completeness - the property of a planning algorithm of finding a solution if it exists.

computational spiral effect - the notion that response depends on throughput in a software system.

configuration space - any abstract space of variables which completely determines the positions of all points on a vehicle or mechanism.

constant flux - a property of a sensor of scanning a fixed solid angle at a fixed rate.

constant scan - a property of a sensor of scanning at fixed angular resolution.

continuity assumption - the assumption that important variables can be predicted in the immediate future based on their current or most recent values.

control laws - the essential mathematics of that part of the navigator concerned with the control of the vehicle.

control layer - a conceptual layer in a software hierarchy characterized by high bandwidth and implemented servo control laws.

control time - the time scale on which the control layer operates.

controllability - the property of a system of being able to reach a desired state in finite time.

Controller - the software module responsible for vehicle control.

coordinate frame - an entity popular in robotics and computer graphics for representing the position and attitude of rigid bodies.

correlated - the property of two quantities to vary in a somewhat or completely dependent way.

correlated disparity assumption - the assumption that disparity is constant across the correlation window in stereo vision.

correlated measurement and process noise - a Kalman filter formulation which permits measurement and process noise correlation.

correlation curve - in stereo vision, the correlation scores for a particular pixel as a function of disparity.

correlation model - in estimation, a model of the dynamics of a noise source.

correlation tensor - in stereo vision, the complete 3 dimensional table of correlation scores for an image.

correspondence problem - the problem of establishing the correspondence between points in two images taken from different vantage points.

cross power spectral density - the Fourier transform of the autocorrelation function of a signal.

crossrange - the horizontal direction transverse to the sensor optical axis.

crosstrack vector - the vector from the end of the lookahead vector to the goal point in path following.

C-space - see configuration space.

curvature - the derivative of heading with respect to distance travelled.

D

data flow diagram - a software engineering diagramming technique that emphasizes the transformations performed on data instead of the logical sequence of operations.

data logger - any tool for recording test data. Here, the software data logger integrated with the navigator.

dead reckoning - any navigation technique which computes position from measurements of speed or higher derivatives, and heading and attitude.

dead zone - that region of the local environment which a vehicle is committed to travelling at any particular time.

deduced reckoning - see dead reckoning.

degrees of freedom - the number of independent variables in a system.

deliberation - the act anticipating the future consequences of current actions.

deliberative - possessing the capacity to deliberate.

depression - the angle of a line in space measured downward from the horizontal.

detection zone - the region of the local environment for which time is almost up to react to hazards.

detection zone tracker - a regulator for sensor azimuth based on maintaining the detection zone in the center of the image.

deterministic inputs - deliberately or otherwise known inputs to a dynamic system.

deviation - the error induced in a compass by the vehicle residual magnetic field.

differential observations - observations of how something has moved or changed instead of where it is in an absolute sense.

disparity / correlation coupling - the notion that the correlation score computed at a pixel in stereo vision depends on the disparity gradient as well as the disparity while disparity itself depends on correlation.

disparity - the difference in the position of two corresponding points in a pair of images.

disparity gradient - the rate of change of disparity with position in the image plane.

disparity image - an image whose intensity values correspond to disparity values.

disparity window - a range of disparities defined by a maximum and a minimum value.

disturbance sequence - a discrete time random signal which excites the system model in a Kalman filter.

downrange - the horizontal direction aligned with the sensor optical axis.

dynamic braking coefficient - see braking coefficient.

dynamic braking regime - the speed regime for which the braking coefficient exceeds unity.

dynamic steering limit - the limit imposed on steer angle or curvature by a given speed and lateral acceleration limit.

dynamic turning coefficient - the ratio of lateral acceleration to lateral acceleration limit.

dynamic turning regime - the speed regime for which the dynamic turning coefficient exceeds unity.

dynamically feasible - the property of a spatial trajectory of satisfying the dynamic model of the vehicle.
dynamics constraint - the constraints imposed on a spatial trajectory by the dynamic model of the vehicle.
dynamics matrix - the matrix of coefficients which relates the state vector time derivative to the state vector.
dynamics reaction time - the reaction time of the actuators and the vehicle body.

E

elevation - the rotation of something about a horizontal axis
elevation angle window - a range of elevation angles defined by some maximum and minimum value.
elevation scanner - the configuration of a laser scanner where elevation is the first rotation of the beam
ensemble - a related family of functions of time.
epipolar geometry - a property of images pairs where the epipolar lines corresponding to a pixel in the first image are aligned with the rows or columns of a second image.
epipolar geometry assumption - the assumption that an image pair exhibits epipolar geometry.
epipolar geometry constraint - the constraint that relates any pixel in one image to a line in a second image.
ergodic - a property of a random process where time averaging is equivalent to ensemble averaging.
Euler angles - a set of three angles used to fix the attitude of a rigid body.
execution monitoring - a planning technique of monitoring that a plan is being executed correctly.
extended Kalman filter - the linear Kalman filter formulation that results from linearizing a nonlinear system about a reference trajectory.

F

feasible set - in optimization, the set of values of the independent variables which satisfy the constraints.
feature based - a designation for an image processing algorithm which concentrates on discrete interesting points instead of all pixels, see iconic.
feedback - the process of measuring the system output and using it in the control of a system.
feedback control - see feedback.
feedforward - the process of predicting system response in the future in order to modify the command inputs of the moment.
fidelity - a fancy word for accuracy. Often used in the context of sound reproduction or dynamic simulation.
fidelity adaptive planning - the technique of adapting planning decisions based on knowledge of system command following and position estimation errors.
fidelity problem - the problem of maintaining adequate fidelity in computations

and measurements.

fidelity ratio - the ratio of actual or predicted system error to some allowable threshold of error.

filter - any frequency sensitive transformation from an input signal to an output signal.

first order filters - those filters which can be characterized by a first-order differential equation.

fix - any technique of navigation which generates a direct measurement of position and not one of position derivatives such as speed.

flat world assumption - the assumption that the vehicle operates in extremely benign terrain.

focal length - the distance behind a lens of an image of an object at infinity. An indirect measure of field of view.

focus of attention - in perception and planning, a region of space to which computations which be confined.

forward model - a model of a dynamic system which is of the form state = function of state and command, see inverse model.

forward modelling - the use of forward models.

frame - short for coordinate frame.

free zones - regions of the local environment that the vehicle cannot reach within some time horizon.

Fresnel Integrals - the two dimensional equations of dead reckoning.

fundamental tradeoff - the tradeoff of speed for resolution that occurs in any system with a throughput limit.

G

Gauss Markov process - a random process with an exponentially decreasing autocorrelation function.

Gaussian filtering - the convolution of any signal with a Gaussian operator.

GDOP - the degradation of precision that results from a nonlinear measurement relationship.

generate and test - a problem solving paradigm which generates potential solutions in a straightforward way and then test if they are the solution.

geometric decorrelation - the reduction in correlation of two identical signals due to warping of one or both of them.

geometric dilution of precision - see **GDOP**.

geometric efficiency - an overall measure of the degree of over or under sampling of a sensor.

goal vector - the vector from the vehicle control point to the goal point in path following.

guaranteed detection - the policy of ensuring adequate resolution in computations, sensing, and actuation.

guaranteed localization - the policy of ensuring adequate accuracy in computations, sensing, and actuation.

guaranteed response - the policy of ensuring adequate response time.

guaranteed safety - the policy of guaranteeing vehicle survival.
guaranteed throughput - the policy of ensuring adequate system throughput.

H

halo effect - the artificial increase in the size of an object due to poor angular resolution in stereo vision.
hardware reaction time - the total reaction time of all hardware components.
hazard space - an abstract space for which every point represents the safety of the vehicle in terms of multiple hazardous conditions.
hazard vector - a point in hazard space.
Heeling error - the error induced in some compasses when the device is oriented vertically.
high throughput assumption - the assumption that sensors and processors have adequate throughput capabilities.
higher order filter - a Kalman filter based on second or higher order propagation of uncertainty.
hill occlusion rule - a relationship between vehicle configuration parameters that must be satisfied to see behind a small hill.
hole occlusion rule - a relationship between vehicle configuration parameters that must be satisfied to see inside a small hole.
homogeneous coordinates - a representation of a space of dimension n in terms of $n+1$ variables and a convention to project an $n+1$ vector onto an n vector.

I

iconic - a designation for an image processing algorithm which processes all pixels in a region, see feature-based.
identify - the process of calibrating system parameters in control theory.
image registration problem - the problem that redundant measurements of the same geometry do not agree.
imaging density - a measure of the number of images which contain the same point in the environment.
impulse turn - a turn from zero curvature to the maximum.
impulse turn maneuver - see impulse turn.
impulse turning distance - the distance measured along the original straight trajectory consumed in an impulse turn.
incremental lookahead angle - the angular width of the region of planner lookahead.
incremental lookahead distance - the linear width of the region of planner lookahead.
inertial navigation system - a system which uses inertial principles to determine its position in space.
instantaneous field of view - the solid angle subtended by a single range measurement.
integrated random walk - the integral of a random walk signal.
intelligent autonomous mobility - the act, problem, or capacity of a robot vehicle

to drive itself.
intensity crosstalk - the effect of intensity of a measurement on an indicated quantity that is not dependent on intensity.
interpolation - the process of providing values for a signal between actual measurements.
inverse model - a model of a dynamic system which is of the form command = function of state, see forward model.

J

Jacobian - a first order derivative of a vector with respect to a vector.

K

Kalman gain - the mapping from the measurement residual onto the required change in the state estimate.
kinematic braking coefficient - the proportion of the stopping distance consumed in deciding to brake.
kinematic braking regime - the speed regime for which the braking coefficient does not exceed unity.
kinematic steering limit - the kinematic limits on steer angle or curvature imposed by the steering mechanism.
kinematic turning coefficient - see turning coefficient.
kinematic turning regime - the speed regime for which turns of the minimum radius are safe.

L

landmark Jacobian - the measurement matrix in a Kalman filter associated with a visual landmark.
latency problem - the problem of latencies that are too large for the current speed or sensory lookahead.
lateral occlusion problem - the problem of subtenion of a large portion of the horizontal field of view by an object.
linearized Kalman filter - a feedforward Kalman filter generated from linearizing a nonlinear process.
linearized system model - the linearized form of a nonlinear system model.
local correlation maximum problem - the problem of mistaking a local extremum of the correlation curve for a global one.
local planning problem - the problem of deciding where to go based only on what can be seen at the moment.
localization - the process of locating objects or the vehicle.
locally-planar terrain assumption - the assumption that the terrain is flat enough that all four wheels remain in contact with the terrain.
logistic layer - another name for control layer.
longitudinal aspect ratio - the ratio of sensor height to wheelbase.
lookahead - the process of looking at or computing future positions of the vehicle.

lookahead angle - the amount of angular lookahead.
lookahead distance - the amount of linear lookahead.
lookahead ratio - the inverse of the number of system cycles of lookahead being used.
lookahead vector - in path following, the vector from the vehicle control point to the point directly forward at the lookahead distance.
low dynamics assumption - the assumption that velocity can be considered constant over a single cycle of any algorithm.
low latency assumption - the assumption that latencies are small enough to be neglected for any particular speed and sensory lookahead.

M

main event loop - the highest level loop in the navigator.
Map Manager - the part of the navigator responsible for building the world model from range imagery.
map matching - the process of fitting range images together based on vehicle motion between frames and/or some measure of best fit.
map resolution - the horizontal spatial resolution of the terrain map.
mapping - the process of generating a map of any kind.
maximal visual similarity assumption - the assumption that two image regions that look the same actually are images of the same physical point.
maximum acuity - the highest resolution that should be needed.
maximum acuity rule - a rule for determining sensor angular resolution.
maximum fidelity - the highest accuracy level that should be needed.
maximum fidelity ratio - the fidelity ratio corresponding to the maximum fidelity criterion. The ratio of obstacle clearance to map resolution.
maximum fidelity rule - the condition that the maximum fidelity ratio not exceed unity.
maximum sensor acuity rule - the condition that the maximum acuity ratio not exceed one-half.
measurement - the quantity actually indicated by a sensor.
measurement gradient - the gradient of a scalar measurement with respect to the state vector.
measurement matrix - the matrix of all measurement gradients.
measurement noise sequence - the random signal that corrupts measurements in a Kalman filter measurement model.
measurement residual - the difference between the actual measurement and the predicted one in a Kalman filter.
mechanism model - a kinematic model of a mechanism.
minimum acuity - the minimum resolution that should be needed.
minimum acuity rule - the condition that the minimum acuity ratio not exceed one-half.
minimum fidelity rule - the lowest accuracy level that can be tolerated.
minimum potential energy principle - the principle of modelling physical systems with an energy minimization process.

<p>minimum sensor acuity rule - a rule for determining sensor angular resolution.</p> <p>minimum significant delay - the smallest unmodelled delay that affects the fidelity of a model.</p> <p>mission analysis - the human process of analyzing a robot mission.</p> <p>modelling - the process of generating a model.</p> <p>modelling problem - the problem of model generation.</p> <p>monotone arc length assumption - the assumption in path tracking that the vehicle never fails so badly as to go in the wrong direction.</p> <p>monotone range assumption - the perception assumption that range is monotone in elevation in the image plane.</p> <p>motion distortion problem - the distortion of the world model due to unmodelled delays or unmodelled motion.</p> <p>multivariable state space system - a differential equation model of a system with more than one state variable.</p> <p>myopia problem - the problem of inadequate sensory lookahead.</p>	
N	
<p>negative obstacle - holes and depressions in the terrain.</p> <p>nonholonomic - the property of a differential constraint that it cannot be integrated.</p> <p>nonholonomy problem - the problem of path generation for a nonholonomic system.</p> <p>normalized baseline - the ratio of stereo baseline to sensor height.</p> <p>normalized correlation - the correlation of two signals normalized to the interval of minus unity to unity.</p> <p>normalized disparity - the angular measure of disparity formed by dividing disparity by the focal length.</p> <p>normalized incremental lookahead distance - the ratio of lookahead distance to reaction distance.</p> <p>normalized map resolution - the ratio of map resolution to range.</p> <p>normalized range difference - the ratio of the width of a range window to its range.</p> <p>normalized time constant - the ratio of a characteristic time to a temporal planning horizon.</p> <p>normalized undercarriage clearance - the ratio of undercarriage clearance to range.</p> <p>normalized wheel radius - the ratio of wheel radius to range.</p> <p>normalized wheelbase - the ratio of wheelbase to range.</p> <p>Nyquist rate - the minimum sampling rate required to recover a signal from its samples.</p>	
O	
<p>objective function - the quantity being optimized in an optimization process.</p> <p>observability - the property of a measurement process of being able to recover the state of a system from available measurements.</p> <p>observability problem - the problem of achieving observability.</p> <p>observation matrix - the matrix of coefficients relating system state to measure-</p>	

ments.

obstacle sparsity assumption - the assumption that obstacles are not dense in the environment.

occlusion constraint - the constraint on stereo range images which prevents them from ranging geometry that can only be seen in one image.

occlusion problem - the problem of missing parts in a world model because radiation reflects from the first reflecting surface only.

on demand interpolation - the process of interpolating only when immediately needed.

onto - the property of a mathematical mapping from one space to another that an inverse mapping exists.

operator - a matrix that is interpreted to move a point in a fixed coordinate system.

optimality - the property of generating the best possible answer in an optimization problem.

output vector - the vector of outputs in a dynamic system.

oversampling - the process of sampling beyond the Nyquist rate.

oversampling factor - the ratio of actual resolution to the minimum required.

P

panic stop - the obstacle avoidance maneuver of slamming on the brakes.

panic stop maneuver - see panic stop.

partial subsampling - an approach to adaptive scan for stereo vision that homogenizes resolution in only the stereo processing stages from disparity on.

path features - distinctive features in a path such as points of high curvature.

path replanning - the processing of replanning on a regular basis or because some error has occurred.

Perception - the component of the navigator responsible for interpreting imagery.

perception ratio - the ratio of sensor height to measured range. Equal to the tangent of the range pixel incidence angle for flat terrain.

perceptual software efficiency - the inverse of the number of floating point operations used to process an average pixel.

perspective - the imaging projection typical of video cameras.

piloting mission - a simple mission of getting safely from one point to another.

pitch - the degree to which the vehicle forward axis is not level.

pitch regulator - an algorithm for keeping the region of interest centered in the image by driving the sensor tilt.

planning angle - the angle through which the vehicle turns in the reaction distance.

planning window - the focus of perceptual attention expressed in terms of the point of actuation.

plant dynamics - the dynamics of the plant being controlled.

policy layer - the highest level entity in the standard architectural model of robotic architectures.

pose - position and orientation treated as a unit.

Position Estimator - the part of the navigator responsible for determining where the vehicle is.

positive obstacle - a hill or rise in the terrain.
potential field - any scalar field which is a potential for a generalized force.
power spectral density - the Fourier transform of the autocorrelation function of any signal.
predictive measurement - a prediction of a measurement used to generate a residual in a Kalman filter.
principle motion assumption - the assumption that the vehicle rotates only about the vertical and translates only in the forward direction.
principle of characteristic times - the notion that required bandwidth is correlated with the lookahead horizon of any level in the standard model.
principle of optimality - the notion that the optimum solution to the overall problem is composed of optimum solutions to its component problems.
process noise distribution matrix - the matrix of coefficients which weights the influence of the process noise sequence on the state vector in a Kalman filter.
process noise sequence - the random sequence which excites the system model in a Kalman filter.
processing reaction time - the total response time of the software components of the system.
processing latency - the time it takes for an algorithm to transform its inputs into its outputs
processor load - the number of floating point operations used to process an average pixel.
projection plane - the plane off of which an incident reflects in geometric optics.
pure pursuit - an algorithm for tracking paths characterized by steering towards a point on the path at a distance in front of the vehicle.

R

random bias - a constant random process.
random process - a family of time signals of which one is selected randomly.
random walk - see Brownian motion.
range gating - a technique of range imaging where all pixels of a particular range window or gate are acquired simultaneously.
range image - an image whose intensity values correspond to the range to the first reflecting surface in the environment.
range rate - the time derivative of range.
range ratio - the ratio of maximum and minimum range.
range window - a region of interest specified in terms of a maximum and a minimum range.
range window tracker - an algorithm for stabilizing a sensor in the tilt axis by keeping the range window centered in the image.
reaction distance - the product of vehicle speed and some appropriate reaction time.
reaction time - the time it takes for inputs to be transformed into outputs in any system.
reactive - a designation for planning systems which emphasizes quick reflexes and

an absence of sophisticated world modelling.

reactive ratio - the proportion of reaction time caused by software.

real time kernel - the real-time component of the navigator.

real-time - a designation for any problem where reaction time and/or throughput are important considerations.

real-time approach - an approach to problem solving that emphasizes response and throughput requirements.

redundancy - a designation for a mechanism where the inverse kinematic solution is many-valued.

reference points - distinguished points on the vehicle body.

reflection plane - the plane formed by the incident and reflected ray of a specular reflection in geometric optics.

region of interest - in image processing, a small region where computations will be confined.

relative accuracy - a measure of the accuracy of the difference of two measurements.

relative crosstrack error - in roadfollowing, the difference between how far off the path the system is and how far off it thinks it is.

repetitive texture problem - a well-known failure mode in stereo vision where repetitive texture is incorrectly assumed to correspond physically.

resolution - the smallest difference that a system can resolve.

response adapted lookahead rule - a condition on lookahead distance which ensures that guaranteed response is maintained.

response adapted speed - the speed computed in response adaptive speed.

response adaptive lookahead - any planning mechanism which adapts lookahead distance to ensure guaranteed response.

response adaptive speed - any planning mechanism which adapts speed to ensure guaranteed response.

response problem - the problem of maintaining adequate response time.

response ratio - the ratio of reaction distance to sensory lookahead.

reverse turn - a turn from one curvature extreme to the other.

rigid suspension assumption - the assumption that vehicle suspension does not deflect at all.

rigid terrain assumption - the assumption that the terrain does not deflect at all.

roll - the degree to which the vehicle forward sideways axis is not level.

S

safe terrain assumption - the assumption that there are no obstacles in the environment.

sampling problem - the problem that the mapping of resolution is not uniform.

sampling property - a mathematical property of the delta function of sampling a signal.

sampling theorem - a famous theorem of signal processing that a single must be sampled at twice its highest component frequency in order to recover it from its samples.

scale factor - the slope of a calibration curve which relates the reading to the quantity indicated.

scanning density - a measure of the average degree of oversampling of the terrain map in a single image.

scatter matrix - a nonnormalized covariance matrix.

selection problem - in perception, the problem of extracting the region of interest from an image.

sensing reaction time - the component of total system reaction time due to sensory hardware.

sensitivity problem - a high degree of sensitivity of one parameter with respect to another.

sensor dwell - the amount of time it takes to acquire a measurement.

sensor flux - the solid angle scanned per unit time by a sensor.

sensor throughput - the sensor measurement rate in range pixels per second.

sign correlation - a stereo technique which reduces the normalized image to a binary or trinary one before correlation.

singularity - loss of rank in a Jacobian matrix.

small incidence angle assumption - the assumption that the perception ratio is small.

small pitch assumption - in inverse kinematics, the assumption that the vehicle pitch is small.

soundness - in planning, the property of guaranteeing that the constraints are satisfied.

spectral amplitude - the amplitude of a frequency spectrum.

stability - the property of a system of generating a bounded output for a bounded input.

stability problem - the problem of maintaining stability.

stabilization problem - the problem of stabilization of sensors.

stacking - a state vector augmentation technique in Kalman filters.

standard model - an architectural model of robotics control software.

state model - a state space model.

state space - an abstract space of all variables of a system necessary to define its response to inputs.

state space model - a model based on a state space representation.

state space simulator - a simulation of dynamics based on a state space model.

state vector - a point in state space.

state vector augmentation - a technique used in Kalman filters to increase the information provided or the accuracy.

static stability margin - the angle remaining before the vehicle tips over.

stationary - having statistics that do not vary with time.

stationary process - a random process that is stationary.

steady-state regime - the regime of operations where the system output is not changing in a material way.

steering regulator - an algorithm for controlling sensor pan to track the steering mechanism.

stereoscopy - stereo vision.

stopping angle - the angle through which a vehicle turns while executing a turning stop.

stopping distance - the sum of reaction distance and braking distance.

stopping region - the region in front of the vehicle which includes all possible stop-ping trajectories.

Strategic Controller - the part of the navigator responsible for goal-seeking.

strategic goal - the goal trajectory of the strategic controller.

strategic layer - in the standard model, the layer responsible for strategic planning.

strategic planning problem - the problem of deciding what a robot should do in order to achieve its goals.

strategic time - the time scale on which the strategic planner operates.

subpixel disparity estimation - a method for generating the highest possible range resolution in stereo vision.

sum of absolute differences - a measure of match between two correlation win-dows in stereo vision.

sum of squared differences - a measure of match between two correlation win-dows in stereo vision.

survey point - in position estimation, a point whose position is known in navigation coordinates.

sweep rate - the angular elevation rate of scanning of a range sensor.

sweep rate rule - a design rule for determining sweep rate for a given vehicle speed.

system reaction time - the total reaction time of the vehicle from image acquisition to executed response.

T

tactical control - in the standard model, the layer responsible for dynamics feed-forward, obstacle avoidance, and goal-seeking.

Tactical Controller - the part of the navigator responsible for obstacle avoidance.

tactical goal - the goal of the tactical controller, usually this is to avoid obstacles.

tactical layer - see tactical control.

tactical time - the time scale on which tactical control operates.

task space - an abstract space in which tasks can be easily expressed, usually this is the world coordinate system.

Taylor remainder theorem - a famous theorem of calculus which bounds the error incurred in using an interpolating polynomial.

temporal planning horizon - the amount of time a system or subsystem looks ahead.

terrain smoothness assumption - the assumption that the terrain is not rough.

test environment - a layer of the navigator which surrounds the real-time kernel and connects it to the vehicle.

thrashing - excessive exception handling in a hierarchical software system.

throughput - a measure of amount of information processed per unit time.

throughput adapted speed - the speed computed in throughput adaptive speed.

throughput adapted sweep - the sweep computed in throughput adaptive sweep.

throughput adaptive speed - any planning mechanism which adapts speed to en-

sure guaranteed throughput.

throughput adaptive sweep - any planning mechanism which adapts sweep to ensure guaranteed throughput.

throughput performance limit - a limit on speed and reliability imposed by a system throughput limit.

throughput problem - the problem of maintaining adequate throughput.

throughput ratio - the ratio of the area covered by the vehicle in one cycle to the area measured by the sensor in the same time.

time constant - the coefficient of the first derivative in a first order system.

transient regime - the regime of operations where the system output is changing in a material way.

transient turning coefficient - the proportion of time that the turning actuator is in the transient regime in a given turn.

transition matrix - the matrix of coefficients which maps state onto state in a discrete time linear differential equation.

triangulation - in navigation or ranging, any process which solves a partially known triangle in order to measure something.

tunnel vision problem - the problem of inadequate horizontal field of view.

turning coefficient - see kinematic turning coefficient.

turning distance - the distance travelled in a turning maneuver.

turning fidelity ratio - the minimum fidelity ratio evaluated for a kinematic turning model.

turning reaction time - the time it takes to execute a turn maneuver.

turning stop - a panic stop maneuver issued while in a turn.

turning stop maneuver - see turning stop.

U

unbiased - a property of an estimator where the estimated mean is equal to zero for all time.

uncorrelated disparity assumption - the assumption that the range of a given pixel reveals nothing about the range of adjacent pixels.

uncorrelated measurement error assumption - the assumption that the errors of two different sensors are uncorrelated, and hence that the measurement covariance matrix is diagonal.

undercarriage tangent - the tangent of the angle from the center of the undercarriage to the bottom of a wheel.

undersampling - the process of sampling below the Nyquist rate.

unforced Kalman Filter - a Kalman filter formulation with no forcing function or control inputs.

uniform scan assumption - the assumption that pixel resolution is inherently adequate.

unknown hazard assumption - the pessimistic assumption that unknown areas of the environment must not be traversed.

update rate - the rate at which an algorithm or hardware component updates its output.

V
<p>variation - the degree to which the earth's magnetic field does not point to geographic north.</p> <p>Vehicle - the component of the navigator concerned with modelling vehicle dynamics and encapsulating state information.</p>
W
<p>wander azimuth - a designation for an inertial navigation system which avoids the problems of navigating near the poles by particular mechanical or computational means.</p> <p>Weiner process - see Brownian motion.</p> <p>wheel fraction - the ratio of wheel radius to wheelbase.</p> <p>wheelbase - the length of the vehicle measured from back to front wheels.</p> <p>white - a designation for a random signal which contains all frequencies.</p> <p>wide depth of field assumption - the assumption that a sensor exhibits wide dynamic range.</p> <p>workspace - the volume of space that a mechanism can reach.</p> <p>world tree - a data structure which relates all significant components of the vehicle.</p>
Y
<p>yaw - rotation about the vertical axis.</p> <p>yaw rate - time derivative of yaw.</p>
Z
<p>zup - short for zero velocity update - a mechanism for self calibration of a Kalman filter, also called zupt.</p>

