THE ROLE OF PRACTICE IN DUAL-TASK PERFORMANCE: TOWARD WORKLOAD MODELLING IN A CONNECTIONIST/CONTROL ARCHITECTURE

Technical Report AIP - 28

Walter Schneider & Mark Detweiler

Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260

29 September 1987

This research was supported by the Computer Sciences Division, Office of Naval Research and DARPA under Contract Number N00014-86-K-0678; Personnel and Training Research Programs Psychological Sciences Division, ONR under Contract N0014-86-K-0107 and the Army Research Institute, under Contract No. MDA903-86-C-0149. Reproduction in whole or in part is permitted for purposes of the United States Government. Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
Distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AIP 28

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Carnegie-Mellon University

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Computer Sciences Division
Office of Naval Research (Code 1133)

6c. ADDRESS *(City, State, and ZIP Code)*
Department of Psychology
Pittsburgh, Pennsylvania 15213

7b. ADDRESS *(City, State, and ZIP Code)*
800 N. Quincy Street
Arlington, Virginia 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Same as Monitoring Organization

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-86-K-0678

8c. ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS p400005ub201/7-4-86

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
N/A	N/A	N/A	N/A

11 TITLE *(Include Security Classification)*
The Role of Practice in Dual-Task Performance: Toward Workload Modelling in a Connectionist/Control Architecture

12 PERSONAL AUTHOR(S)
W. Schneider & M. Detweiler

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM 86Sept15 TO 91Sept14

14. DATE OF REPORT *(Year, Month, Day)*
87 September 29

15. PAGE COUNT
47

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Skill acquisition, cognitive psychology, dual-task performance, practice effects, transfer effects, connectionism

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

SEE REVERSE SIDE

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION

22a NAME OF RESPONSIBLE INDIVIDUAL
Dr. Alan L. Meyrowitz

22b TELEPHONE *(Include Area Code)*
(202) 696-4302

22c. OFFICE SYMBOL
N00014

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

Abstract

The literature on practice effects and transfer from single- to dual-task performance and part-whole task learning are briefly reviewed. The results suggest that single-task training produces limited transfer to dual-task performance. Past theoretical frameworks for multi-task performance are reviewed. A connectionist/control architecture for skill acquisition is presented. The architecture involves neural-like units at the microlevel, with information transmitted on vectors between modules at the macrolevel. The simulation of the model exhibits five phases of skill acquisition. Dual-task interference and performance are predicted as a function of the phase of practice the skill has reached. Seven compensatory activities occur in the model during dual-task training that do not appear in single-task training: 1) task shedding, delay and buffer pre-loading; 2) letting go of high-workload strategies; 3) utilizing noncompeting resources; 4) time multiplexing; 5) shortening transmissions; 6) converting interference from concurrent transmissions; and 7) chunking transmissions. Future research issues suggested by the architecture include: Mapping out the marginal utility of single- to multi-task transfer; investigating the classification of multi-task compensatory activities; evaluating the role of part-task trainers for multi-task skills; and developing and testing quantitative models of skill acquisition.

28 September 1987

The Role of Practice In Dual-Task Performance: Toward Workload Modelling in a Connectionist/Control Architecture

Walter Schneider & Mark Detweiler

Learning Research and Development Center
University of Pittsburgh
3939 O'Hara St.
Pittsburgh, PA 15260

Phone: 412-624-7061

28 September 1987

Running head: Dual-task practice

Submitted to Human Factors, for Summer, 1988
Special issue

Do not quote without permission

Introduction

Despite the fact that a vast number of studies have attempted to chart the kinds and nature of interference that occurs when two or more tasks are performed simultaneously, little is known about the role of practice in reducing interference. The few studies which have investigated practice have demonstrated that human performance can change radically over time. Moreover, given its primary importance in achieving high levels of skill, there have been few attempts to provide theoretical frameworks sufficiently detailed to address issues such as: 1) why human performance becomes more accurate, 2) why it becomes faster, 3) why interference between tasks decreases with practice, and 4) why single-task training does not always transfer to dual-task performance. This paper examines issues of practice and workload. First, we summarize a number of issues concerning single- and dual-task performance and training. Next we describe a runnable simulation architecture used to investigate a variety of single- and dual-task phenomena. Then, we use this architecture as a framework to clarify a few central issues and problems surrounding the kinds and nature of dual-task interference that occur and illustrate how it many change as a function of practice. And finally, we speculate on the kind of research and modelling needed to develop more predictive and theoretically-based approaches to understanding dual-task training and performance.

Interference Between Highly Practiced Tasks

Bahrick and Shelly (1958) provide strong support for the idea that even after prolonged single-task training the addition of a secondary task can have a pronounced impact on the practiced task. Bahrick and Shelly's subjects all received 25 training sessions; a visual serial reaction time task was practiced alone on sessions 1-2, 4-13, 15-24 and combined with a secondary auditory task on sessions 3, 14, and 25. Single-task sessions consisted of 10 trials of 100 stimuli each, and dual-task sessions were 4 trials. Although performance on the visual task reached a relatively high level of proficiency within the first 20 trials, when paired with the auditory task, performance on session 3 dropped as little as 15% for a group trained on consistent stimuli to as much as 40% for a group trained on inconsistent stimuli. On sessions 14 *and* 25 performance dropped from about 8% for the group with most consistent stimuli and about 40% for the group with the least consistent stimuli. The last session showed no additional benefits in the dual task of the extra 10 single-task practice sessions. In other words, despite substantial single-task practice, subjects were unable to jointly perform the auditory task without incurring a cost on the practiced visual task--with the most inconsistent visual tasks suffering the largest costs.

Damos, Bittner, Kennedy, and Harbeson (1981) offer further evidence of the need for dual-task practice and illustrate the time needed to regain single-task performance levels after the two tasks have been combined. In this experiment subjects performed 15 trials (about 15 minutes) per day of two concurrent tracking tasks for 15 successive working days. One task required use of the left hand and the other the right. Prior to this study subjects had received 15 sessions of single-task critical tracking, 15 sessions of compensatory tracking, and 15 additional sessions of a critical-compensatory tracking dual task. In other words, subjects had received a total of 45 prior trials over a 15 week period. Damos et al. found that performance on the dual task improved sharply over the first four days of practice and continued to improve throughout the testing period. Moreover, after 15 sessions of dual-task practice, subjects' performance approximated the levels obtained after three single-task practice sessions. This study provides evidence suggesting that even after substantial single-task practice, additional practice was needed to stablize performance when two tracking tasks had to be performed concurrently.

Long ago it was noted that some people can develop the ability to read and dictate simultaneously (see Solomons and Stein, 1896; Downey and Anderson, 1915). Spelke, Hirst, and Neisser (1976) were the first in recent times however to document the effects of practice on two subjects' abilities to read short stores while copying auditorily presented words. Initially the subjects' reading speed and comprehension scores were seriously impaired, yet after six weeks of practice their dual-task performance measures matched their single-task baseline scores. Near the end of the experiment Spelke et al. introduced two transfer tasks in which the two subjects either read aloud or shadowed prose while copying words as before. Copying produced decrements in both reading aloud and shadowing. Reductions in interference were evident as the the tasks were practiced together. In a follow-on study, Hirst, Spelke, Reaves, Caharack, and Neisser (1980) replicated their earlier study and demonstrated further that with substantial practice--over 50 hours--subjects were able to read both highly redundant and less redundant materials nearly as quickly and with comparable comprehension as their single-task levels. In a second experiment subjects learned to read short stores and at the same time copy sentences varying in length from 3 to 7 words read at rates between 20 to 31 words per minute. Following substantial dual-task training, subjects achieved reading and comprehension rates similar to single-task control rates.

Shaffer (1975) has documented one highly practiced typist's (SW) abilities to perform dual tasks. First, SW was able to type visually presented material and concurrently recite nursery rhymes at a cost of about 10% in typing speed and accuracy over single-task baselines. Second, SW was able to concurrently type and shadow prose presented auditorially at a rate of 140 words per minute, again with a comparable 10% cost in speed and accuracy. Third, SW was able to concurrently type and shadow random letters presented auditorily at a rate of one per second with similar cost. However, when SW was required to type auditorily presented material and to shadow visual input, her speed and accuracy suffered greatly. Similarly, when SW was required to type auditorily presented material and shadow auditory input--a male voice in one ear and a female voice in the other ear--her performance was markedly impaired.

Allport, Antonis, and Reynolds (1972) recruited piano players who were able to play by sight reading. Subjects participated in two training sessions in which they attempted to shadow auditorily presented prose while sight reading material of varying difficulty. By the second session subjects' performance in the dual task was comparable to single-task baselines. In addition, when tested for recall of the prose materials, subjects' memories were comparable in single- and dual-task conditions.

Schneider and Fisk (1982) raised the question of whether two visual search tasks could be performed concurrently without a drop in accuracy. They showed first that practice under single-task consistent mapping (CM), i.e., where the same response is made to stimuli across trials, conditions improved search performance, whereas it did not in inconsistent or varied mapping (VM), i.e., where the response made to a specific stimulus varies across trials. Performance under the dual-task CM conditions showed a dramatic improvement with practice, benefiting the shortest display durations most. Performance under the dual-task VM condition also improved somewhat, but did not reach single-task VM performance at the end of the first experiment. A second experiment further showed that subjects were able to perform concurrent VM and CM searchs when the VM task was emphasized. While VM performance did not result in a deficit, CM performance did drop by 17%. However, when the CM task was emphasized, VM performance dropped precipitously to near chance levels. Finally, they also demonstrated that concurrent VM searches could not be performed without substantial deficit which did not diminish with additional practice.

In another set of experiments Fisk and Schneider (1983) had subjects perform three single tasks, i.e., a digit-span task, a consistent visual category search task, and an inconsistent visual category search task and two dual-tasks, i.e., the digit task joined with each of the two search tasks. When the digit and consistent search tasks were first performed together, detection accuracy dropped by about 10% from the single-task detection baseline. After 90 additional trials of dual-task practice, however, detection accuracy reached a level comparable to single-task accuracy. In contrast, when the digit and inconsistent search tasks were first performed together, detection accuracy dropped by about 25% from single-task levels. Moreover, additional dual-task practice did not improve detection accuracy. In contrast, inconsistent tasks show marked deficits in dual-task conditions and these deficits decline little with extended practice. When single tasks are combined for the first time, there is a substantial decrement in performance even if the tasks are consistent and extensively trained. Extended dual-task training substantially improves consistent-task performance but not inconsistent-task performance.

The first session of dual-task training typically produces dramatic drops in performance of either CM or VM performance. Schneider and Fisk (1984) examined the effects of practice and transfer in consistent and inconsistent category visual search tasks. Subjects first performed three single-task searches, i.e., digit search, consistent category search, and inconsistent category search. Subjects completed 1,755 single-task trials (7-8 45-minute sessions) for both the consistent and inconsistent tasks. This was followed by two dual-task conditions in which the digit search primary task was paired with each of the semantic search secondary tasks. Hence subjects were required to detect both digits and category exemplars concurrently. The first time each of the two semantic search tasks was combined with the digit-search task, the semantic search performance dropped substantially--45% for the consistent task and 49% for the inconsistent task--relative to single-task performance levels. Subjects' performance at detecting digits also dropped substantially--26% for the consistent task and 29% for the inconsistent task. Subjects' performance on the consistent category search task improved with additional dual-task practice, matching single-task levels after four additional sessions and ceiling after eight. Subjects' performance on the inconsistent category search task failed to improve with additional dual-task practice; the smallest decrement was 49% and the largest was 61%, relative to single-task performance. Finally, it should be noted that even after eight sessions of dual-task practice, performance on the digit task still incurred an 11% decrement from its comparable single-task level for the consistent search task and a 17% decrement for the inconsistent search task.

The results on CM or VM practice show that given a high degree of consistent practice on a search task, subjects could simultaneously perform dual tasks with little deficit, yet they needed practice under dual-task conditions to reach their single-task detection levels when the two tasks were combined. Further, despite high degrees of single-task practice, subjects were unable to perform the inconsistent/consistent-search and digit-span tasks without substantial deficit.

Automatic/controlled processing theory suggests that two qualitatively different forms of processing can account for the marked changes that can occur in performance with consistent practice (see Schneider, Dumais, & Shiffrin, 1984; Schneider & Shiffrin, 1977; Shiffrin & Schneider, 1977). Controlled processing is characterized as slow, effortful, capacity-limited, and largely under subject control. Automatic processing is characterized as fast, parallel, relatively effortless, and largely not under subject control. The processing demands of most complex tasks typically reflect neither purely automatic or controlled processing, but rather a mixture of the two.

A simple application of automatic/control processing theory to dual- or multi-task training might advocate practicing consistent single-task components first, prior to having the learner perform the tasks concurrently. In single-task training components become automatic, no longer requiring attention. They could then be combined to perform dual tasks. However, the data are incompatible with this simple view. In the Schneider and Fisk (1984) experiment, both the consistent and inconsistent tasks dropped nearly equally in the first dual-task block.

Rieck, Ogden, and Anderson (1980) raised several issues concerning the relative effectiveness of single- and dual-task practice on subsequent dual-task performance. In the training phase of their experiment, subjects received either four 3-minute practice trials on a discrete compensatory tracking task, four trials of the tracking task combined with a digit cancellation task, or three different practice combinations of the two tasks. Then each group performed one 3-minute transfer trial in which a discrete tracking task was performed together with a delayed choice reaction time task. This was followed by a second transfer trial in which a continuous tracking task was joined with a choice reaction time task. Rieck et al. found that practice on the single-tasks had little influence on subsequent dual-task performance. In contrast, dual-task practice did have a decided influence--the greater the prior dual-task practice, the greater the influence. In sum, the extent of dual-task practice accounted for 28 to 35% of the dual-task variance, whereas the single-task practice accounted for only 2%. Although we caution against overinterpreting results based on such small amounts of practice, we nevertheless feel they are suggestive of the potential utility of dual-task practice, especially in view of the fact that the transfer tasks were not the same as the practiced tasks.

Several studies have shown that performance on multiple tasks predicts pilots' performance better than comparable performance on single tasks. For example, North and Gopher (1976) developed a test battery consisting of digit-processing, reaction-time, and one-dimensional compensatory tracking tasks. In the first phase of their study subjects performed the three tasks separately under adaptive conditions designed to elicit maximum baseline measures for each task. In the second phase task pairs were performed concurrently with both tasks given equal priority. And in the final phase, subjects performed different combinations of tasks varying in priority. In general, the correlations obtained between the single- and dual-task conditions were low, suggesting that performance on single tasks is a poor predictor of performance on multiple tasks. Damos (1978) and Damos and Lintern (1981) offer further evidence that multiple-task performance is a better predictor of success in flight training than is single-task performance.

Part-Whole Training

Part-task training refers to practice on a subset of components comprising a whole task. The logic behind part-task training is that learning can proceed more efficiently and hardware expenditures can be reduced, e.g., using part-task trainers or simulators. Many whole tasks are multi-task situations. For example, flying an aircraft may require performing flight control, communication, and nagivation tasks. Studies of part-whole training might provide insights with respect to how much part-task training transfers to multi-task situations. Unfortunately, despite the fact that a large number of studies have appeared since the first documented experimental investigation of part versus whole learning appeared in 1900, an accompanying framework has not emerged that identifies key experimental dimensions to explain conflicting results, or a valid and reliable set of training principles or guidelines (see McGeoch, 1931; McGeoch & Irion, 1952; Naylor, 1962; Stammers, 1982; and Wightman & Lintern, 1985). Recently Stammers (1982) tested the generality of the prevailing principles.

Stammers sought to test a set of principles proposed by Naylor and his co-workers (e.g., Blum & Naylor, 1968, Naylor, 1962, and Naylor & Briggs, 1963) and Annett and Kay (1956). The so-called Naylor hypothesis is embodied in two principles: 1) as complexity is increased for relatively highly organized tasks, training the whole task should work better than training parts of the task; and 2) as complexity is increased for relatively less organized tasks, training parts should become more efficient than training the whole task. Annett and Kay's principle suggests that the more independent a task's parts, the more it should be learned as a whole; the more interdependent a task's parts, the more its parts should be separated for training. Stammers conducted four experiments to evaluate the generality of these principles by evaluating part and whole learning in a procedural control panel task and a list learning task made up of operational instructions.

On the whole, Stammers found little support for the notion that practicing parts of a task produces advantages over practicing the whole task. Differences between training groups were small, with whole training gaining a slight advantage. Nevertheless, he cautions that deciding to use one form of training over another should be made on the basis of empirical observations, rather than some analytic principles. Although Stammers has helped to illuminate a literature filled with conflicting results, it should be noted that neither the studies reviewed, nor his own experiments adequately address the issue of task interdependence and complexity as it pertains to the relative efficacy of part-task practice for concurrent tasks.

Several researchers examining part-whole learning in dual-task situations emphasize whole-task practice. An alternative to decomposing a complex task into its component parts and practicing them separately exists in the form of adaptive training. In adaptive training the task is first simplified and is then made progressively more difficult as the learner acquires greater levels of expertise (see Johnson & Haygood, 1984; Kelly, 1969; and Lintern & Gopher, 1978). Typically the learner is exposed to the whole task or almost the whole task to be mastered. However, differential pay-off conditions emphasize some tasks over others. In this way, each component is practiced in the context of the whole task. Johnson and Haygood (1984) have found that progressively challenging the learner in a primary simulated driving task (tracking) and a secondary visual detection task resulted in better performance than training in the single-task conditions.

In general, the part-whole training literature does not provide a strong support for either part or whole training. Nor does it identify critical variables that can predict when and how much part-task training should precede multi-task training. In general, most complex, real-world skills, e.g., piloting, driving, and programming, are initially developed via part-task training until at least some basic level of proficiency is reached before multi-task training is begun.

Frameworks for single- to multi-task transfer

Although psychology can offer a variety of theoretical frameworks for multi-task performance, there are no models that predict single- to multi-task transfer. Theories of multi-task performance generally involve switching, e.g., Broadbent (1958), or allocation of some limited resource, e.g., Kahneman (1973). Recent models have elaborated the nature of resource sharing, e.g., Navon & Gopher (1985), or attempted to stratify the types of resources, e.g., Wickens (1980, 1984b). These frameworks have generally not been applied to practice effects. From switching theories one might assume that practice speeds up the switching rate and facilitates learning what orders and rates of switching are most effective. From resource theories one might assume practice facilitates learning what proportions and types of resources to allocate to a task. In dual-task training, an individual can learn what combination of resource allocation produces optimum performance. For example, if a memory task can be performed spatially or verbally, training with a concurrent spatial task will encourage the subject to perform the memory task using verbal coding. Wickens (1984a) comments that practice can increase the resource efficiency, so that the same task can be accomplished with progressively less resources. The mechanism for this practice effect has not been specified.

Recent frameworks that predict learning effects in single-task experiments suggest that single-task training should transfer to dual tasks. Pew (1974), for example, extended previous models of tracking and suggested that humans develop higher-order control mechanisms. In tracking tasks, novices continuously monitor the feedback at a low level of control, e.g., monitoring momentary error between actual and intended position of the object being controlled. With practice attention moves to higher levels, e.g., monitoring the drift between the control equations for the desired and intended positions. Shiffrin and Schneider (1977) extended a long line of argument -- from at least James (1890) -- that practicing consistent task components develops automatic components that require little if any attentional resources. Anderson (1983) suggested that practice compiled productions, i.e., if-then rules, produce automatic component skills. Norman and Shallice (1985) proposed that practice reduces interference by allowing processes to function without attention. Hunt and Lansman (1986) presented a computer simulation model incorporating production systems and direct activation functions. In this model, the system begins executing general productions by comparing the input to the desired input. On a match, the appropriate response production is released. With practice, direct associations develop between input and output, allowing the input to evoke the output without requiring the attention-based comparison. Hunt and Lansman's model accounts for some dual-task procedures as a result of time-sharing a limited executive control system.

None of the present frameworks directly address the issue of transfer from single to multiple tasks. To understand issues related to transfer, much more detailed models are required. Even the limited data on transfer are ambiguous. Models are needed that can specify boundary conditions and predict the appropriateness and marginal utility of various types of practice to multi-task performance. The following model is an initial step of elaborating a model to make such predictions.

Overview: A Connectionist/Control Architecture

The proposed connectionist/control architecture provides a runnable simulation model for representing a class of models of dual-task performance and workload (see

Schneider & Mumme, 1987; Schneider & Detweiler, 1987). We first provide a brief overview of the connectionist control architecture. Connectionist models (see Schneider, 1987) represent a radical departure from energy-based metaphors, e.g., Kahneman (1973). We recommend readers new to this type of modeling carefully work through the diagrams and examples in the text. After the overview, we will illustrate how skill is acquired within this model in both single- and dual-task processing.

This architecture incorporates a variety of processing elements that provide mechanisms for accomplishing stable information processing of real-world tasks in an architecture that is neurally feasible (see Schneider & Detweiler, 1987). The model implements a five-phase account of skill acquisition in which skill development is characterized as gradual and continuous. Five assumptions underly and constrain this architecture. First, we assume that information is processed in networks of neural-like units. These units are organized into modules, and sets of modules are organized into levels and regions. The various regions, e.g., auditory, speech, lexical, etc. are connected to and communicate with one another. Second, we assume the modules exhibit local specialization of function, i.e., processing only a restricted class of inputs. Third, we assume knowledge is stored in the connection weights among the neural-like units. As skill is acquired due to learning, changes in knowledge are reflected in changes in the connection strengths among units and/or by the size of their weights. Fourth, we assume the connection weights operate under the influence of a variety of learning rate constants. These constants determine how quickly the connections change as a result of intervening learning and the duration of the retention interval. And fifth, we assume a control processing system modulates the transmission of information within and between processing regions. This architecture is a variation of the CAP1 system designed to model automatic and controlled processing (see Schneider & Shiffrin, 1977; Schneider & Mumme, 1987; and Shiffrin & Schneider, 1977).

The connectionist/control architecture can be described at three levels of scale. The microlevel structure is the first level and represents a network of neural-like units that account for associative information processing and a range of attentional phenomena (see Figure 1). A message vector is an output from one module and an input to another module, transferring information between modules. A vector is a set of activities of the units within a module. For example, the letter "A" might be coded as a message vector 0,1,1,1,1, where the 0's and 1's represent the absence and presence of features, e.g., vertical lines, horizontal lines, backward slant and forward slant. The message vector is the set of activation values. Information flowing from a module is regulated by an attenuation unit within each module (see Figure 1; Schneider & Mumme, 1987; and Schneider & Detweiler, 1987 for details). Attention is a scaler multiplication of the activity of all of the units, e.g., if half attending to the letter "A", the output would be 0,.5,.5,.5,.5,.5. The macrolevel structure is the second level of scale and represents interactions among a set of modules (see Figure 2B). The modules are organized as levels and regions of processing. The levels correspond to successive processing stages. For example, the visual module is assumed to consist of a series of stages that include units such as features, characters, and words. The system-level structure is the third level of scale and represents the interactions among processing regions made up of modules (see Figure 2). Each region consists of a series of levels of modules and their respective control structures. There are regions specializing in input, e.g., visual, auditory ect., output, e.g., motor, speech, etc., and associative processing, e.g. semantic, spatial, context, etc. The innermost levels of each region communicate, i.e., pass vector messages, to other regions.

Insert Figure 1 about here

All of the regions communication with each other. Regions communicate on an innerloop of associative connections (see Figure 2). Each module is assumed to have associative connections to the other modules on the innerloop.[1] These connections allow each module to send message vectors to other modules. There are separate connections to each receiving module. These connections can transform one output transmission to different messages in different regions in parallel, e.g., a bright flash may transmit from the visual system and associatively evoke a startle response in a motor module, a fear response in a mood module, and an attempt to retrieve related information in a semantic module. The fact that each module on the innerloop has its own connections also enables multiple modules to transmit messages simultaneously, e.g., the visual module may transmit to the motor module while the auditory module transmits to the speech module.

Insert Figure 2 about here

Note, parallel transmission on the innerloop does *not* necessarily imply parallel processing. All the message vectors coming into a module are summed (see Figure 1). When two messages are added, inter-message interference results. This is analogous to a cocktail party situation. Every speaker can speak at loud volume for extended periods of time -- there is no energy limit on output. However, each listener receives the summation of all spoken messages. Parallel transmissions may result in high interference such that no message is received (see below). In such cases more information is conveyed if individuals speak sequentially, even though each has the power to speak in parallel. Within the proposed architecture, the control processing is the mechanism by which CAP1 moderates message transmissions on the innerloop.

There are two categories of information flow in the system; these are message and control information flow. Message flow involves the transmission of a vector representing a code from one module to another, e.g., the visual module sending a vector

[1] This is a simplifying assumption in reasonable speculation based on neuroanatomy (Mishkin & Appenzeller, 1987). Most of the arguments made in this paper will apply to lattice architectures that involve substantial message convergence between messages transmitting on the innerloop.

coding the features of the letter "A" to the semantic module. Control flow involves exchange of control information between the modules and a control structure (see Figure 2B). Control information codes the importance of messages waiting to transmit and the transmission state of any of the modules, e.g., signals indicating how active a module is, how important a message to be sent is, or how much to attenuate the message. At the macro- and system-levels, control structures receive control information to moderate information flow, e.g., sequencing transmissions to reduce interference. At the system level the control flow processing involves a central control structure (see Figure 2). This structure receives activity reports from all regions. These reports are a single number from each module which indicates the importance of the message in the module awaiting transmission. This is analogous to each module raising its hand, the more critical the message the higher the hand. The control structure then ranks the requests for transmission and allows the module with the highest priority request to transmit. A three-neuron per module circuit (shown in the bottom of Figure 1) is sufficient to accomplish this control function.

Control processing is critical for performing novel tasks. The connectionist/control architecture can perform a novel task following verbal instruction (see Schneider & Mumme, 1987). This is accomplished by loading vectors into modules, comparing input vectors to vectors held in working memory, and releasing output vectors (see below). This processing is slow, serial, and effortful, i.e., it requires many shifts of which modules are allowed to transmit (see below). The control information necessary to limit message interference also enables execution of verbal rules. With each execution of a verbal rule, associative connections between the modules change such that the input will evoke the output without moderation by control processing. This transition (see below) is the mechanism through which processes become automatic.

In this architecture, messages need not all pass through a central executive; rather, regions can communicate directly with other regions. Since the transmission of concurrent messages may cause interference, the central control structure may moderate transmission among regions. Sequencing messages in this way has the potential of causing delays or omissions. The model predicts many of the item and order loss phenomena witnessed under conditions of high workload.

A context storage mechanism enables the system to model the stability of human processing and to mimic a variety of learning phenomena, e.g., episodic memory and remindings (see Schneider & Detweiler, 1987). This context mechanism acts to associate the contents of the messages on the innerloop to the temporal context in which they occur. The context vector has separate connections to all the other modules on the innerloop (see Figure 2). One context vector can evoke a different message vector on every module on the innerloop. When information vectors within modules decay or are displaced, modules can be reloaded by retransmitting the context vector. The ability to reload information via transmitting the context vector reduces the chances of catastrophic processing failure. Further, when the system is confronted with high levels of workload, context can temporarily store associations to vectors and enable the reloading of a low priority task delayed during the processing of high priority tasks.

Phases of skill acquisition

The connectionist/control architecture has been implemented as a computer simulation called Controlled/Automatic Processing Model 1 (CAP1) (see Schneider & Mumme, 1987; Schneider & Detweiler, 1987). The simulation can perform single and dual tasks comparable to typical laboratory dual tasks. We will describe the model and the phases it exhibits as it learns to perform single and dual tasks. It important that the reader try to conceptualize the operations of the model[2]. Most of the implications follow from the conceptual architecture of the model. The details of the simulation provide an existence proof that this type of architecture will produce the phenomena discussed.

The model is a connectionist simulation model; details, equations and rationale for the architecture can be found in Schneider & Mumme (1987) or Schneider & Detweiler (1987). The model simulates the behavior of populations of neural-like units. It incorporates the associative and autossociative models of J. A. Anderson's (1983) brain-state-in-a-box model. The major components are illustrated in Figure 1. Each module is made up of a 200-element vector of output units. Output units are neuron-like units in which the pattern of active units code the response. Information is coded as a vector of -1's (inactive units) and +1's (active units). Each output unit sums its input linearly with a decayed value of the activity of the previous iteration. The output of each unit is a logistic function of the input with a minimum activation of -1.3 and maximum activation of +1.3, e.g., connecting within the set. Each output unit decays to some proportion (typically .9) of its activity on the previous cycle. Each output unit connects autoassociatively to half the other output units of the module. Autoassociative feedback varies from 0 for receiving a new vector to .6 to latch a received vector.[3] Each output unit connects to half of the units of other modules to which it is connected. The autoassociative and associative connection matrices are each made up of 20,000 connections per module.[4] All associations were initialized at zero strength.

Two types of learning were accomplished using a modified Hebb-type learning rule, the delta or Widrow-Hoff learning rule (see J. A. Anderson, 1983). The first type, associative learning, modifies connections between modules. This involves changing the connection weights between the input and output units such that the input comes to elicit the output. The second type, priority learning, occurs within a module. This

[2] Walter Schneider has produced an animation of the model that runs on an IBM PC with a color monitor. It provides a dynamic two-dimensional representation of the system, illustrating the stages of skill acquisition and the problems and compensatory activities to enable dual-task performance.

[3] A feedback strength of 1.0 is one in which the feedback signal is as strong as the input signal.

[4] These results generalize to larger matrices typical of cortical hypercolumns (Mountcastle, 1979).

involves changing the autoassociative weights between the input unit and the priority unit. Stimuli that are consistently attended strongly activate the priority unit which results in the vector being transmitted from the module. Stimuli that are not attended, weakly activate the priority unit and inhibit the vector from being transmitted unless control processing enables transmission of the vector. Attention is implemented by the strength of the attenuation unit (see Figure 1). The attenuation unit is multiplied by the strength of the output unit between 0 (unattended output) to 1 (fully attended output). Controlled processing involves modulating attenuation and monitoring the activity of the modules (see below). Automatic processing involves activating the priority unit sufficiently to reliably transmit the vector to the next stage of processing in the absence of controlled processing input.

Development of automatic processing in a single task

The CAP1 simulation has been used to model single-task skill acquisition. Details of the modeling can be found in Schneider and Mumme (1987). In this paper we present only an overview of single-task learning and focus on dual-task performance.

CAP1 acquires skills through five phases. The movement between these phases is a gradual, continuous transition. The rate of movement between stages depends on the task to be learned. We will illustrate the transitions using numbers based on subjective impressions from work in search paradigms (Schneider & Fisk, 1984), and learning logic gates from electronic troubleshooting (Carlson & Schneider, 1987). The phases are:

1. Controlled comparison from buffered memory (Trials 1-4)

2. Context-maintained controlled comparison (Trials 5-20)

3. Goal-state-maintained controlled comparison (Trials 21-100)

4. Controlled-assist of automatic processing (Trials 101-200)

5. Automatic processing. (after Trials 200 per component task).

The first three phases involve extensive attentional processing. In Phases 1-3, the subject serially compares stimulus information with information in memory. Once the skill has shifted to Phase 4, there is a substantial reduction in attentional processing and a qualitative change in the processing.

Phases 4 and 5 involve direct associative retrieval of output patterns from input patterns at a series of processing levels. We will illustrate single-task learning using a category search task patterned after Fisk and Schneider (1983). In this task the subject must remember one or more categories and respond to a visually presented word by indicating whether the word is a member of the remembered categories. Subjects' initial performance is slow, e.g., 200 ms per comparison, serial, e.g., the "no" responses are twice as slow as the "yes" responses, and effortful, e.g., subjects perform poorly under a secondary-task load. This processing characterizes controlled processing. In a consistent mapping (CM) search task subjects make the same response to the same stimulus over trials. Practice in consistent search results in processing that is fast, e.g., 2 ms per comparison, parallel, e.g., equal "yes/no" slopes, and low in effort, e.g., subjects can perform a secondary task (see Fisk & Schneider, 1983). This processing characterizes automatic processing. In contrast to consistent practice, practice in a varied mapping (VM) task, where the categories searched for on one trial become distractors on the next, shows little performance change. After ten hours of VM practice, performance was about as slow, serial and effortful at the end of training as in the beginning (e.g., Fisk & Schneider, 1983).

Phase 1 -- Controlled comparison from buffered memory involves loading and maintaining memory vectors in modules, comparing comparison vectors to the input vectors, and releasing the appropriate response vector. CAP1 performs a controlled comparison by adding two vectors to determine their similarity. Vector addition entails that each output unit (Figure 1) adds the inputs it receives. The report cell receives the square or the sum of the absolute value of all of the output units.[5] To perform a comparison, two vectors are added together. If the two vectors are similar, e.g., "cat" and "animal", the added vector is nearly twice as long, since both vectors are pointing in the same direction. In contrast, if the two vectors are dissimilar, e.g., "car" and "animal", the vector elements add orthogonally, pointing at right angles. This produces a shorter vector, e.g., of length 1.41 for vectors with correlation of 0. The difference in length provides a criterion for match, e.g., a length greater than 1.85 (correlation = .5) is defined as a match. Processing is serial because adding triplets of vectors produces a more error-prone comparison process (see Schneider & Mumme, 1987).

Phase 1 processing is very effortful in the sense that it requires many shifts of attention and monitoring the received activity of vectors. For example, to perform a two-category search task with a display size of two, the subject must actively maintain the two category vectors and two response vectors ("yes" and "no" responses), shift attention four times (2 visual stimuli x 2 categories), compare the length of the added vectors to the criterion length four times, shift attention to the output module, and release a response vector. Phase 1 performance is very error prone. If the subject is interrupted, the vectors in the buffers will decay, causing errors. If the attention-switching operations are disturbed, e.g., by a secondary task, the comparisons cannot be made, resulting in response delays or omissions.

Insert Figure 3 about here

----

[5] The equation for the addition of the vector is SQRT($X^2 + Y^2 + 2r_{x\,y}$ X Y), where X and Y are the vectors, and $r_{x\,y}$ is the correlation of the two vectors.

Phase 2 -- Context-maintained controlled comparison is similar to Phase 1, except information is maintained in fast learning weights that associate vectors stored in modules to the context. Activating the context module can refresh information in modules (see Schneider & Detweiler, 1987).[6] The connections between the context module and the modules on the innerloop are assumed to be fast learning weights (see Figure 2). These connections can quickly associate the context to the current contents of modules on the innerloop. After a small number of trials, e.g., four repetitions, the context can evoke the category and response vectors. This context storage mechanism allows the system to re-activate the vectors when new visual stimuli are presented. After four trials of searching for the same categories, subjects can perform a distractor task, e.g., counting backwards by 3's between trials, without performance being disrupted. Klapp, Marshburn, & Lester (1983) have demonstrated that if subjects rehearsed the memory set for five seconds in a memory scanning experiment, performing a digit-span task between the presentation of the memory set and the probe display did not disrupt performance.

Phase 2 processing is effortful and requires attention, but it will not be as seriously disrupted by interpolated tasks. Performing a category search task requires as many shifts of attention and comparisons as in Phase 1. However, the context can now be used to activate the vectors in the buffers. Note, the context storage only works for runs of the same task and is of little value if what the subject is searching for changes from trial to trial. Fast connection learning weights show very serious proactive interference effects (see Schneider & Detweiler, 1987) and provide little information about previous associations if new associations are made to the same context. During Phase 2, processing is very reliable for runs of the same comparison set, but reverts back to Phase 1 with every change in the set to be searched for.

Phase 3 -- Goal-state-maintained controlled comparison is similar to Phase 2 except the goal state can reload the modules in addition to the context-based reloading. For example, assume the subject has to learn three rules A, B, C. In Phase 2, if the subject performs a series of A trials, the A vectors become associated to the context and can be reloaded if the information decays from the buffer. However, on the first trial with the B rule, the subject must be told what vectors to load in the buffers. This loading associates the B vector to the context, making the A vectors less available. In Phase 3, the subject learns to associate appropriate vectors to multiple goal states A, B, and C, instead of a single context vector. When the task changes, the subject needs only a short time to remind him/herself about the rule to reload the buffers. The subject then performs the same attentional operations as in Phase 1.

[6] This context learning mechanism is important for robust processing and enables the model to account for a wide range of working memory phenomena, e.g., episodic and semantic memory, retroactive and proactive interference, elaborative rehearsal, and release from proactive interference effects; it is consistent with physiological data suggesting the existence of a two-speed learning system (see Mishkin, Malamut, & Bachevalier, 1984).

Phase 3 processing requires attention and involves the slow, serial, effortful form of processing characterized in Phases 1 and 2. It is somewhat more reliable, in that performing an interpolated task in the same modules will not disrupt performance. The difficulty subjects have going from practicing a single rule to multiple rules provides evidence for the presence of Phase 3 processing. Carlson and Schneider (1987) found subjects could perform single digital logic gate judgement tasks, e.g., the output of an AND gate is high if all inputs are high, in .7 seconds if the rules were massed after about 5 trials. However, over 1000 trials per gate were needed before they could perform trials of mixed gate types, e.g., AND, OR, NOR, etc. at that speed.

Phase 4 -- Controlled assist of automatic processing produces a dramatic reduction in processing time and effort relative to the previous three phases (see Figure 3). During Phases 1-3 the subject repeatedly compared the input to the vectors in the buffers. The match processes involved four comparisons. A match was followed by a yes response, a non-match by a negative response. In Phase 4, associative learning alters the connections between the input and output modules such that the input evokes the output (see Schneider & Mumme, 1987), e.g., if the stimulus "cat" is transmitted, the learned associative responses evoke the "index finger" response. The resulting associative process eliminates the need for controlled comparison. If a target vector is transmitted from the input it will associatively evoke the appropriate response in the output. Attention is still required to transmit the vector from the input module to the output module. However, attention switching of the comparison vector and monitoring are no longer necessary. Processing is parallel in memory in the sense that the input will evoke its output, independent of the number of input/output pairs learned by the connection matrix.[7]

Phase 4 requires a small amount of attention to transmit the input vectors on the innerloop to the output modules. If there are multiple input channels, attention would still switch between the inputs. For example, in a category search task with two words being presented, i.e., display size two as in Fisk & Schneider (1983), the subject attends to the first word transmitting the vector. If the association evokes a "yes" response, then the response is made, if not, the second word is attended and transmits the second vector. If the vector evokes a positive response, then a response is made; if not, after attending to the last stimulus input, a negative response is made. Based on previous research (Fisk & Schneider, 1983) a category comparison requires about 200 ms and an associative retrieval probably 100 ms.[8] Based on these numbers we can compare the time attention is required to perform a two-word to two-category comparison. During

[7] There are capacity limits to associative networks, (see Mumme & Schneider,1987), however. Until there are about as many associations as half the number of connections in the matrix, associative retrieval can be considered independent for uncorrelated input vectors.

[8] The latter is more difficult to estimate since it is based on intercept effects. Treat this number as a rough estimate for purposes of illustration. We speculate the minimal transmission time is in the range of 30 to 100 msec for well-learned, highly discriminable stimuli.

Phases 1-3, four comparisons and one attended response are made, requiring 1.0 s (4 x 0.2 s comparisons, 0.2 s response output). During Phase 4, two attention switches are made in 0.2 seconds (2 x 0.1s visual) and one associatively evoked response transmission (0.1 s motor). When the subject has developed a reliable level of proficiency in Phase 4, there is a 70% reduction (1 s vs. .3 s) in the amount of attention and an 88% reduction (.8 s vs. .1 s) in transmission time on the innerloop. At this stage substantial attention and innerloop transmission time are available to perform other tasks.

Phase 5 -- Automatic processing occurs when automatic processing substitutes for attentional processing. During Phases 1-4, various vectors were attended in order to transmit them to later modules. A message that was transmitted prior to a positive event, e.g., the visual input "CAT" associated with a "yes" response, would be associated within the module with a high priority tag. In contrast, messages that are transmitted without follow-on events, e.g., the word "CAR" which is never responded to in the experiment, would have a low priority tag. Automatic processing occurs when a message associated with a high priority event is transmitted in the absence of attentive input. This occurs when the local circuit of the priority tag inhibits the attenuation units transmitting the message (see Figure 1, box AP). Automatic processing can occur through a series of stages. Each stage involves an association of the output of the previous vector which evokes a new vector within the module. Then, that vector is categorized via autoassociative interactions and evokes the priority tag. If the tag is high enough, the priority tag causes the vector to be transmitted out of the current module to the next module. This process then cascades through a series of stages.

Phase 5 processing requires no attention, however, the automatic transmission of messages does require some transmission time on the innerloop. In a search paradigm, priority learning eliminates the need for attentional switching. Assume the words "CAT" and "CAR" are presented, with "CAT" being a previous target and "CAR" a previous distractor. The "CAT" message will evoke a high priority tag and be transmitted, whereas the "CAR" module will evoke a low priority tag and not be transmitted. Transmitting the "CAT" vector will evoke the positive response in the motor module. The motor response will evoke a high priority tag and transmit its message and cause the "yes" response to be made. The priority tag-based filtering in Phase 4 eliminates the need to serially transmit visual vectors as in Phase 4.

Table 1 contrasts the resource demands of performing the search task as a function of phase of processing.

Table 1 Resource demands as a function of the phase of skill acquisition

Phase	Attention Time	Attention Switches	Comparisons	Innerloop Time
1 Controlled Processing (CP)	1.0 s	5	4	0.8 s
2 Context maintained CP	1.0 s	5	4	0.8 s
3 Goal state maintained CP	1.0 s	5	4	0.8 s
4 Controlled assist AP	0.3 s	2	0	0.2 s
5 Automatic processing (AP)	0.0 s	0	0	0.1 s

Problems of multi-task performance

In many applied contexts, humans must perform multiple tasks concurrently. Performance typically deteriorates sharply when subjects are asked to perform concurrent tasks. The connectionist/control processing architecture provides an interpretation of the difficulties of dual-task performance and the improvements that occur with practice. Having a subject perform multiple tasks will have different consequences depending on what phase of practice the subject has reached before dual-task processing occurs.

Having a subject perform multiple tasks during Phase 1 will show severe disruption in performance. For example, in a memory scanning experiment, if the subject briefly sees a memory set (too briefly to allow rehearsal) and then performs a distractor task, performance is very error prone. The errors result because the buffer codes can not be maintained and the comparison time requires considerable controlled processing and innerloop resources (see Table 1). Performing a secondary task must be multiplexed with the attentional processing required by the primary task, e.g., 1.0 s for the 4-comparison category search task. This attentional processing is required during Phases 1-3.

Phase 2 processing can be performed after an interruption from an irrelevant secondary task. In a search task, after the subject rehearses the words several times or performs several trials with the same memory set, the memory set vectors are associated to the current context. If an irrelevant task is performed that requires attention, the memory set vectors will decay. By re-evoking the context, the memory set vectors can be reloaded and the memory comparison process restarted. However, if the subject performs the search task with a new memory set, the new set will be associated to the current context and make the previous memory set unavailable. Therefore, the subject will not be able to timeshare a process that requires context storage of information in the same buffers.

Phase 3 processing can be performed after an interruption from a relevant secondary task. Once the goal state can evoke the vectors for comparison, the context storage is no longer necessary. For example, if the subject is looking for words from categories A, B, or C, in Phase 3, presenting the category labels is sufficient to allow the appropriate vectors to be evoked. The actual comparison still requires substantial attentional processing, e.g., 1.0 s for the category search.

Phase 4 processing requires little attentional processing. It can be timeshared with other tasks. The other tasks must not require the same modules as the Phase 4 task, and provide short periods of time when attention can be allocated to the Phase 4 task. In the category search example, the Phase 4 process required .3 s of attentional processing relative to 1.0 s for Phases 1-3.

Phase 5 processing requires only a short period of innerloop transmission time and the modules necessary to process the input and output. Phase 5 processing can occur concurrently with other tasks. The other tasks must not completely monopolize the innerloop transmission capacity. In the category search example, the Phase 5 process requires .1 seconds of innerloop transmission time. Assuming one set of stimuli is presented at a rate of one per second, the category task would require the visual modules to process the word, .1s of innerloop transmission time to transmit the visual vector to the motor module, and the motor modules to output the word. A secondary task could utilize all other modules, all of the attentional control processing, and 90% (.9s of 1 s) of the innerloop transmission time. In this way two tasks utilizing different input/output modules with modest interloop transmission, e.g., reading while taking dictation, could be processed concurrently at single-task performance levels.

Phase 5 processing is still limited. If two modules transmit on the innerloop at the same time, there will be a loss of information. In the model, when multiple messages are transmitted, the received message is the addition of the transmitted messages. For example, in a task in which the subject responds to a word with a button press and to a tone with a spoken word, the motor module might receive both the tone and the visual features of a word. If the messages are of equal strength and evoke incompatible outputs in the receiving modules, no message will be received (see below).

Compensatory activities and training in dual task

Given the above changes in performance, *why does single-task training show such limited transfer to dual task performance*? A simple view of automaticity would predict that once a task is automatic, it should be able to be combined with other tasks without deficit.[9] If one accepts this view, one should train all the single tasks and spend relatively little time in dual-task training. However, the above literature review shows this prescription to be wrong, e.g., Schneider & Fisk (1984) found nearly novice-level dual-task performance after 8 hours of single-task training.

[9] A view sometimes inappropriately ascribed to the first author.

This section focuses on what happens differently in dual-task training than in single-task training, and describes how the system compensates in the dual task to facilitate performance. In general, performance in a dual-task situation puts a premium on the use of attention and innerloop transmissions that are not critical in the single-task situation. For example, in single-task category search, a Phase 4 task requires .3 seconds of attention and .2 seconds of innerloop transmission time. Performance is fast, accurate, and involves little effort, and there is little need to change strategy. In contrast, in a dual-task experiment even the .3 second attention load may be unacceptably high.

In the dual-task situation an entirely new set of behaviors must be executed that are *not* required in the single-task situation. To illustrate this, consider a category/tone dual task. The category task requires responding to visually-presented animal words by pressing a button with the right hand. The tone task requires saying "target" whenever a high tone is presented. Assume the category and tone tasks are at Phase 4 levels of skill. In single-task category search, the subject need only switch attention between the input channels, monitor the motor channel, and release the response. In the dual task, the subject must additionally switch between the visual and auditory regions. If the regions transmit at different time scales, transmission durations must be changed. The system must switch what is being monitored, i.e., during visual transmission the motor system is monitored and during auditory transmission the speech module is monitored. If different criteria are needed for received messages, e.g., the motor association is stronger than the speech association, these must be switched between tasks. If a module is loaded inappropriately, the module must be cleared to load the message from the appropriate message, e.g., in the category/tone task, if the speech system loads a message during the transmission of the visual word, that message must be cleared before the message associated to the tone can be loaded. The system must also block the interference effect of the second message transmission, e.g., the motor module must not be cleared by the transmission of the tone message, the motor message must output to other levels in the motor region while the tone message is being transmitted.

The types of compensatory activities that are effective in multi-task situations depend on what phase of skill the subject is in on each of the tasks. If two tasks are in Phase 1, dual-task performance is very erroneous except for very simple tasks. Errors occur because the subject is unable to maintain information about the second task while performing the first task. If the subject is in Phase 2 or 3, the two tasks can be done by performing each task separately with little overlap. The subject can use context and goal-state information to reactivate the vectors of the second task after performing the first task. In Phase 4 the subject must switch attentional processing between the tasks, but the input vectors will evoke the appropriate outputs without having to reload or compare vectors. In Phase 5 the subject must modify the interloop transmissions to eliminate interference among messages.

There are seven behaviors that the current architecture illustrates in dual-task situations that either do not occur in, or are not as critical for single-task situations. These all involve decreasing the load on limited attentional resources and innerloop transmissions.

1. Task shedding, delay, and pre-loading

When two tasks must be performed concurrently, the easiest compensatory activities are to either: 1) not perform one of the tasks, or 2) delay the lower priority task. Not performing tasks eliminates the cognitive load of those tasks. Training in a dual-task enables the learner to realize which tasks can be deleted and how much cognitive capacity is made available to other competing tasks. Experience in performing multiple tasks enables one to anticipate and monitor the consequences of delaying or eliminating a task. For example, under conditions of high workload, experience guides the performer in knowing when to delay a task or to delete it altogether in order to attend to a higher priority task component.

To manage workload, it would be best to even out potential workload peaks, either by delaying or pre-executing procedures. Delay involves buffering the input from one task while performing the more critical task. In the present connectionist/control architecture, every module is a buffer that can maintain information for short periods of time (see Schneider & Detweiler, 1987). If stimuli from two tasks are presented simultaneously, the lower priority stimulus can be buffered while the higher priority task is completed, then the lower stimulus can be processed. If the two tasks require the same modules, e.g., foveal vision, the first task can be performed first and then the second task.

Pre-loading involves preprocessing information prior to the onset of the critical workload segment. If vectors can be preloaded and maintained in modules, these activities need not be executed while the stimuli are being processed. For example, the context storage system can activate vectors in many modules simultaneously. If an operator reviews the various tasks before the critical task segment begins, attending to the context can evoke vectors in multiple modules simultaneously.[10] Training combat helicopter pilots illustrates this technique. Pilots are encouraged to verbally rehearse possible actions before a "pop up" maneuver exposing themselves to enemy fire -- a high workload situation. In the present architecture this rehearsal associates task-relevant vectors to the current context. The context can then maintain the vectors so they do not decay when they are not rehearsed. Preloading eliminates the need to recall each procedure once the task has begun.

[10] Under the restriction that each task operates in separate modules.

2. Letting go of unnecessary high workload strategies

Subjects' strategies in single-task conditions may entail a greater workload than necessary. The five phases of skill acquisition described above represent CAP1's learning stages. Certain levels of practice are required before later phases become reliable. However, a subject's strategy may influence how long s/he continues a high workload strategy after the lower workload strategy is active. In our letter search experiments (see Schneider, 1985) a proportion of our subjects continued to exhibit serial search, i.e., reaction times that increase linearly with the number of comparisons long after the majority of our subjects exhibited parallel search, i.e., non-linear or flat-slope functions. When these subjects were pressured to respond faster, we often saw a dramatic break in the slope function. This break suggests they shifted from a Phase 3 to a Phase 4 strategy.

Training under high workload, e.g., in multi-task situations, encourages subjects to adopt low workload strategies. In a single-task situation, a high workload strategy, e.g., using Phase 3 when Phase 4 or 5 are reliable, results in somewhat slower and more effortful processing than necessary. A high workload strategy may also be somewhat more accurate than a lower workload strategy.[11] In the single-task situation subjects may persist in using a high workload strategy. In the multi-task situation, using a high workload strategy on one task may produce very poor performance on another task -- due to depriving the second task of resources. In order to boost total performance the learner may try to alternate strategies between trials. The learner may realize the low workload strategies are reliable, e.g., learning to read provides an illustration of this (see Laberge & Samuels, 1974). Students can voice each word as they read it and become very accurate at sight reading. However, if a reader uses this high workload strategy to decode the words, s/he is slow and cannot allocate sufficient resources to adequately comprehend the text. In a speeded reading task with a comprehension test, the reader is more likely to "let go" of the high workload word decoding strategy. Once the learner begins practicing the later phases of the skill, these phases can become reliable and enable availability of lower workload strategies.

For the CAP1 model, experience during later phases of skill acquisition results in better transfer to high workload performance than during earlier phases. For example, practice during Phases 1-3 produced associations between the added vectors, i.e., those used in the comparison operation and responses. In the category search task, the added semantic vectors of the visual word and the lexical category were associated to the motor response, e.g., the vector sum of "CAT" + "ANIMAL" are associated to the "yes" response. During Phase 4 performance, the word vector is transmitted without the category vector, e.g., the vector "CAT" alone without being summed with "ANIMAL". Phase 5 performance matches Phase 4 performance in that the word is transmitted, as in Phase 4, but not the category, as in Phase 3. Associations built up during Phase 4 are of

[11] With modest levels of practice, e.g., less than 100 trials per component, Phase 4 is less reliable than Phase 3.

the input stimuli. Transfer from Phase 3 to 4 will be depend on the degree of overlap between the combined and individual-vectors (see Schneider & Mumme, 1987). Once Phase 4 is reliable, practice using a Phase 4 strategy will produce better transfer to Phase 5 than practice using a Phase 3 strategy.

3. Utilizing noncompeting resources

During multi-task training, the subject can learn to allocate task components to minimize resource competition. Single-task practice facilitates developing a strategy that is fast and accurate. Multi-task training facilitates developing the optimal combined strategies for performing all of the tasks. For example, in an air intercept control task, one can determine the trajectories of two aircraft in order for them to intersect using either quantitative or spatial procedures. In single-task training subjects will execute the strategy stressed by the instructor and that produces fast and accurate responding. However, if the learner practices the intercept task while performing a concurrent task requiring spatial operations, e.g., navigation, or quantitative operations, e.g., calculating flying time, the optimal strategy depends on the context.

Within the proposed connectionist/control architecture multiple resources can be invoked to accomplish tasks (see Schneider & Detweiler, 1987). Baddeley, Grant, Wight, and Thomson (1974) provide an empirical demonstration of utilizing noncompeting resources to perform concurrent dual tasks. For example, digits in a spatial task can be stored either spatially, e.g., as a visual image of a grid, or verbally, e.g., as the proposition "5 left of 8." When subjects were required to perform a concurrent tracking task, performance was better when they used a verbal code to store the digits (Baddeley, et al., 1974).

Wickens (1984) has proposed that human factors designers should allocate tasks to modalities to minimize resource competition. Early statements of this view, e.g., Wickens (1980), suggested that modalities had different resource pools, and hence putting two tasks into different modalities would be advantageous. Recent reviews have shown that dividing tasks between modalities sometimes improves and sometimes deteriorates performance (Wickens, Fracker, & Webb, 1988 for this conference). In the CAP1 architecture, no simple prescription of dividing tasks among modalities is applicable over a wide range of tasks or practice levels.

In the present architecture, placing tasks in different regions, e.g., modalities, is sometimes advantageous and sometimes disadvantageous. Placing tasks in different regions has the benefit that there is no competition between multiple vectors in the same modules, e.g., letters can be stored in a visual module and tones in an auditory module with little interference. However, it may be more difficult to switch attention between modalities than within a modality. Remember that Phase 1-3 level skills require a great deal of attention switching, whereas Phase 4 requires little, and Phase 5 none. This suggests that dividing tasks between modalities may produce inferior performance for modestly practiced tasks, e.g., under 200 trials per stimulus in a search task, while at the same time producing superior performance for consistent, well-practiced tasks which do not require attention switching.

4. Time-multiplexing skills

Training in the multi-task situation provides experience in learning to time-multiplex the transmissions on the innerloop to accomplish the combined tasks. If two modules transmit messages at the same time, the potential of intermessage interference exists. For example, in the dual category/tone task, if both the visual and tone vectors transmit at the same time, the receiving modules will receive the summed vector, e.g., the sum of the semantic vector of "CAT" and "HIGH TONE". These vectors are likely to be unrelated and the interference will result in both messages being blocked (see Figure 4A).[12] If the messages are multiplexed, i.e., transmit "CAT" then "HIGH TONE" (see Figure 4B), both messages can be received accurately after a small delay for the second message.

Insert Figure 4 about here

Multi-task training provides the learner opportunities to exercise different multiplexing schemes. Given that two tasks, A and B, must be accomplished, there are many multiplexing schemes. Should A be completed before B is started? Should all the inputs be multiplexed before the outputs are multiplexed? Should A be multiplexed at a higher rate, e.g., process task A twice as often as task B? Senders (1983) and Moray (1984, 1986) have shown that after extended training, human operators learn to sample instrument gages at the optimal rate, based on the relative information rate of each channel. The allocation of internal control processing may be tuned through experience in a manner comparable to the way the operators allocate attention between gages.

5. Shortening transmissions

In multi-task situations innerloop transmission time is at a premium. Learning to transmit messages with shorter transmissions enables more transmissions per unit time and hence better multi-task performance. In a single-task condition there is no benefit for shorter transmissions. When the visual module transmits to the motor module, a transmission of 100 or 500 ms may have effectively the same result. When the motor module receives the message in 100 ms, the response is begun. If the visual system transmits for an additional 400 ms, no damage is done. Long transmission times have the potential benefit of increasing the reliability of the transmission in the event that some messages require more than 100 ms to complete. Single-task training is likely to lengthen the transmission time of messages transmitting on the innerloop.

[12] The degree of inter-message interference depends on the relative strength of the vectors, orthogonality of the vectors, and the degree to which the receiving module has learned to receive each vector. If the messages are of equal strength, orthogonal, and equal in degree of learning, neither message will be received reliably.

Multi-task training encourages the learner to shorten transmissions to the minimum length sufficient to transmit the information (see Figure 4C). This is analogous to what is seen in air traffic control tasks. When traffic is light, controllers' radio communications are often long, social, and more is communicated than is essential. In contrast, under heavy traffic conditions, communications become almost telegraphic, relying on a minimal number of jargon phrases to convey necessary information. In high workload situations short communications allow more tasks to communicate critical information between sender and receiver.

In the simulation model, multi-task training enables the system to determine the minimal transmission time and to tune the receivers to reliably detect short duration transmissions. Under the pressure of high workload, the operator is likely to vary transmission times on one task to free capacity for other tasks. The algorithm for varying transmission time is simple -- if the transmission was successful, reduce transmission time on the next trial; conversely, if transmission was unsuccessful, increase transmission time. The net result is that the system finds the minimum transmission time necessary to transmit the message reliably. A secondary benefit of practicing with shorter transmission times is that short transmissions can tune the receivers to categorize the noisy short transmissions into the appropriate message. Reception of short, noisy transmissions becomes more reliable if the receiving module increases the degree of feedback and the amount of autoassociation (see Schneider & Mumme, 1987). Practice under high workload allows the system to improve performance by shortening transmission times and strengthening the within-module associations or feedback. In single-task practice these changes produce negligible performance improvement, and the system may asymptote with particularly long transmissions.

6. Converting interference from concurrent transmissions

In addition to avoiding interference by procedures such as multiplexing, associations can be modified to effectively tune out specific interferring messages. Such a tuning effect can be illustrated in the dual category/tone task in which the subject responds to animal names by pushing button A and non-animal names by pushing button B. The tone task requires the subject to say "target" (response C) to a high tone. To illustrate interference effects, assume the subject also learned to make the motor response (push button D) to the high tone. During initial dual-task training the conflict between the visually-evoked response (A or B) and the tone-evoked response (D) would produce message interference in the motor module. The module would receive the combined message (A+D) which is ambiguous. This interference necessitates multiplexing the messages (see point 4 above). To eliminate interference, the tone input would have to either elicit no response in the motor system, or elicit a response that does not interfere with the A or B responses.

Multi-task training can associate an irrelevant input to the relevant outputs such that the irrelevant input does not alter the outputs. The process of converting interference from concurrent transmissions occurs through changes in the association matrices between modules. The process is one of orthogonalizing the vectors, i.e., each connection matrix is altered so that the transmission of the message from one task does not bias the receiver modules of the other task. In the dual category/tone task, multiplexing can associate the input tone with the motor responses for the visual input. On a dual-task trial, the visual stimulus would be transmitted and evoke the appropriate motor response, e.g., the word "CAT" evoking the A response. Shortly thereafter the tone would be transmitted and evoke the appropriate response in the speech region, e.g., if high tone say "target". In a short period, the motor response would therefore receive two input messages (word and tone) and make one response (response A). Associative learning would cause the input to be associated to the output. Thus the tone association to the previous motor response (response D) would be weakened, and the association to the current response (response A) would be strengthened. Similarly, the response between the word and the speech output would be modified by the word determined by the tone. On other trials, the tone would be associated to the motor response of the responses of alternative category stimuli. With practice, the tone would no longer evoke its motor response, but rather evoke a response made up of the average of the two motor responses to the visual input. After dual-task training, the motor module would receive the summed code of A for an animal name and the average of A+B for the tone. The net input (1.5A+ 0.5B) would be highly correlated to the A response and would generally be categorized as the A response. At this point the tone stimulus could be transmitted simultaneously with the word without deficit, thus allowing true parallel transmission of both stimuli with little loss (see Figure 4D).

Note that interference conversion is specific to the messages trained, and all possible combinations of the messages must be trained. For example, to learn to associate 10 visual stimuli to 3 motor responses and 5 auditory stimuli to 2 speech responses requires learning 20 visual-to-speech patterns (10 visual stimuli to 2 speech responses) and 15 auditory-to-motor responses to convert the interference. A stimulus that does not evoke an interferring response, e.g., the word "blue" normally evokes a key press, would require little training for interference training. However, if it evokes a strong interference, e.g., the visual word "blue" evoking the speech output of the word as in a Stroop (1935) task, extensive training will be required to convert the interference. If new stimuli or responses are introduced, these will also have to be trained. Interference conversion is specific to particular messages and does not disconnect regions. For example, practice in the category/tone task may orthogonalize the motor output from a high tone message. However, such practice would minimally affect the motor association of dissimilar auditory inputs, e.g., respond to the word "jump".

The specificity of interference effects predicts the asymmetric interference effects often observed in attention and multi-task situations. For example, in Stroop (1935) interference the word interferes with naming the color but not vice-a-versa. Since one set of messages is extensively trained relative to another, e.g., naming printed words rather than naming the colors of printed letters, there is an asymmetric interference effect. Shaffer's (1975) report on dual-task typing also illustrates this. He found concurrent visual typing and auditory shadowing were done with little cost, whereas concurrent auditory typing and visual shadowing produced severe interference. In the present model one would expect extensive practice at copy typing with some concurrent comprehending of concurrent messages would strengthen visual word-to-motor connections, strengthen auditory-to-semantic connections, and somewhat weaken the cross connections. These changes would support parallel visual typing and auditory transmission while resulting in very poor auditory typing and visual comprehension.

7. Chunking transmissions

If the system can transmit chunks or compact codes, more transmission time can be available for other tasks. Consider the behaviors of a copy typist. The letter pattern "THE" might be transmitted to the fingers as the "T", "H", "E", requiring three transmissions (see Figure 5A). In contrast, if the visual system can transmit a combined code of "THE", only a single transmission is needed (see Figure 5C). Transmitting syllable chunks would reduce innerloop transmissions by about 66%. With a chunk size of three, three tasks can be accomplished with only a minor delay in each task.

The motor system could then decode the chunks and output the individual messages. Practice in the single-task condition will show little benefit from chunking unless the output is limited by the transmission time, e.g., until a person types faster than 10 characters per second there is no speed advantage in developing chunk codes for a single task. In contrast, in the multi-task situation any reduction in attention or transmission time on one task provides more attention and innerloop transmission time for other tasks.

Developing chunk codes can occur when modules transmit in parallel from input to output. In the simulation model (see Schneider & Mumme, 1987), the system modifies the connections between the input and output modules so the input comes to evoke the output. To develop proper associations, the correct output must be in the output module before the to-be-learned input vector is transmitted. This can occur if all the input modules first sequentially transmit each message individually, then all the input modules transmit simultaneously before the output is associated to the input. For example, consider transmitting the word "THE". Initially, if all three messages are transmitted simultaneously no letter is received, due to inter-message interference. However, if the letters are transmitted sequentially, the "T", "H" and the "E" can be evoked in the output region, translated to the appropriate response, and buffered (see Schneider & Detweiler, 1987). If all the input modules transmit simultaneously, followed by the transmission of the output, the combined input code will come to evoke the combined output code.

Insert Figure 5 about here

A chunk can be transmitted by transmitting all the elements of the chunk in parallel (Figure 5B). To accomplish parallel transmission of the chunk, interference conversion must occur (see above, Figure 4D) such that the message transmitted to the first module does not inappropriately bias the second module. For example, in the context of transmitting "TH" from the first 2 modules, the "E" code in the third motor module would be activated strongly, other vowels weakly, and inhibit most consonants. Transmission of "TH" would not interfere with the transmission of "E" or "A" but would interfere with the transmission of "C" in the third position. This is the same process as discussed in multi-task training in which the visual and auditory transmissions did not interfere. In chunking, it is important that parallel transmissions from modules in the same region (see Figure 2B level 3 visual modules) do not interfere with concurrent transmissions from the same region. Developing such transmission capacity requires training in the "multi-task" situation of transmitting the codes concurrently. Training is specific to the codes practiced, e.g., practice at sending "THE" will result in no transfer to sending "AND".

A second method of transmitting a chunk is by compacting the information into a chunk code and transmitting the chunk code from a single module (Figure 5C). In the current model, the visual region can transmit either letters or syllables on the innerloop. In acquiring typing, the motor system would first learn to respond to individual letters, e.g., for the visual transmission of a "T" move the left index finger to the "T," position and press the key. A word would produce a series of letter codes in the second level of the visual region and a series of movements in the third level motor region (see Figure 2B). With training the last level of the visual region would develop a combined code for the visual pattern "THE" and the first level of the motor region would develop the combined code for the motor task of outputting "THE" on the keyboard.[13]

Developing chunk codes has an important advantage over parallel transmission in that it allows for greater capacity to buffer transmissions. In the current model, Schneider & Detweiler (1987) assume that regions can maintain vectors in only a small number (three to four) of modules for a given level of processing. A typist who could visually chunk syllables could maintain information for 12 letters in the visual system and 12 keys in the motor system. A typist without visual chunking could only maintain 4. By chunking the encoding and decoding one can operate asynchronously. The eyes can move at a rate determined by word frequency, and the fingers by the typing configuration of the keys with an average of four transmissions per every 12 characters. For a 60 word per minute typist this would entail only 2 transmissions per second. Delays in visual transmission would have no impact on output until the average delay reduced transmissions below 2 per second or long delays occurred together, such that all the motor buffers would be output before the next visual transmission occurs. In contrast, transmitting the letters in parallel (Figure 5B) will cause delays whenever the encoding time for any 4-letter string is greater than the typing time of that string and will require holding the next visual input until the current motor output is completed.

[13] This could be accomplished by training a set of hidden units to encode all the features of the previous layer (see Rumelhard, Hinton, & Williams 1986).

This is because in parallel transmission, the same motor modules are used for reception of the letter and output of the letters. Hence, the visual system cannot transmit the next set of letters until the first set of letters is completely output. If chunk transmissions occur, the motor system can receive the second chunk while the first chunk is being output (see Schneider & Detweiler 1987).

The three modes of outputting chunks shown in Figure 5 produce dramatically different output rates and innerloop loads. Using a typing example, assuming a 50 word per minute (wpm) typist, 6 characters per word, .2s transmission times, 1-5 letter syllable chunks (average 3 letters): sequential output would average a 100% innerloop processing load and the parallel or chunk transmission a 33% load (one transmission every 3 letters). If one assumed that the encoding and decoding occurred at one character per .1 seconds, a typist would have an average typing rate of: 50 wpm for sequential transmission (.2s per letter transmission); 60 wpm for parallel transmission (.2s per transmission and .3s delay for the output 3 characters from the buffer); and 100 wpm for chunked output (.3s to concurrently input, transmit and output a 3 letter chunk).

Research Agenda

The present theoretical framework provides an agenda for the study of single- to multi-task transfer. The literature review at the beginning of this article illustrates that there is very little research on how single-task training transfers to multi-task situations in which practice levels are sufficient to develop skilled performance. Learning how to optimize this transfer is important for applied questions such as optimizing simulator time in skill acquisition and answering basic questions such as how cognitive processing changes with practice.

Research and formal modeling of single- to multi-task transfer is still in its infancy. To understand the acquisition of high performance skills will require collaboration among the applied and basic research communities. Due to the high cost of training, most high-performance skill acquisition occurs in applied training programs. The basic research community can perform laboratory studies tracking the phases of skill development and produce quantitative models of skill acquisition. The following research issues are listed to suggest directions for future research that seem particularly fruitful within a research program evaluating skill acquisition within the proposed connectionist/control architecture.

Marginal utility of single to multi-task transfer

The present model predicts there is a declining benefit of single-task training for performance in multi-task situations. However, there is a benefit for single-task training. In order for associative learning and priority learning to occur, the learner must be able to maintain the necessary vectors in memory and to perform the task accurately. Starting a learner in a multi-task situation is likely to overload processing during Phases 1-3. Hence it is beneficial to instruct the learner on each component task individually. The learner should then perform the task in the single-task mode until performance is fast and accurate. To assure that at least a goal-state-maintained skill level (Phase 3) is developed, it is important that subjects be able to perform well, even when required to randomly execute the behaviors.[14] Failure to provide this single-task training can hinder progress because the component associations never become reliable.

Single-task training has a reducing marginal utility. After a certain level of skill is reached, continued single-task training can be inefficient compared to multi-task training. First, multi-task training allows the subject to develop the seven compensatory activities described above, including task shedding/delay, letting go, utilization of non-competing resources, time multiplexing, shortening transmissions, chunking, and converting interference from concurrent transmissions. In addition, multi-task training typically provides more practice on both tasks for the same total practice time, allows components to become integrated, and generally is more motivating to the learner.

Systematic research is needed to identify parameters predicting the marginal utility of single-task training and the optimal sequencing of component- and total-task training to maximize training effectiveness. Task complexity variables need to be identified to predict how many trials are needed to transition a skill through various phases. More studies comparing one part-task training scheme to one whole-task scheme will be of little benefit. A sufficient set of such studies (see above, Stammers, 1982) already exists and provides an ambiguous picture. The CAP1 model shows that training can be inefficient if either too little or too much part-task training is provided. A meta analysis lumping studies with varying degrees of practice together is expected to show inconclusive results.

To develop guidelines to improve training, criterion variables must be identified to predict when to shift from part training to aggregate training. The current model suggests one should move from part- to whole-task training once the individual's component skills have reached at least the controlled assist phase (Phase 4) of proficiency. The phase of skill development can be independently verified using secondary task tests to see how well the skill can be performed under workload (see Schneider & Detweiler, 1986). Studies mapping out the functional relationships between proportion of time that should be spent in part- and whole-task training for a variety of tasks and training times would be particularly useful.

Examination of learning multi-task compensatory activities

The present architecture suggests seven compensatory activities that develop in multi-task situations that have little significance in single-task situations. For the most part, these compensatory activities have not been extensively studied. For example, what is the functional relationship between delaying one task and the probability of correctly executing the task without restarting the task? In multiplexing, how many

[14] Phases 1-2 become fast and accurate as long as the same task is executed repeatedly after instruction on the component.

channels can the central control structure keep track of before some requests are lost? How sensitive is transmission reliability to the duration of transmissions? How long does it take to build a higher-level chunk code, and does it require explicit practice aimed at building chunk transmission codes? How hard is it to switch between modules within a region and between regions? How fast and how fully can dual-task training orthogonalize transmissions to reduce intermessage interference? Can individuals sample information from different regions at different rates? Do they naturally develop different sampling rates? Do individuals discover, on their own, what tasks should optimally be shed in high workload situations? Real world multi-task performance requires the development of compensatory activities. Basic research and training guidelines relating to understanding compensatory activities is severely needed.

Role of part-task trainers for multi-task skills

It is important to identify how part-task training can be modified to increase transfer to multi-task performance. In many complex training systems, part-task training can be much cheaper than full-task training. For example, training a pilot how to operate radio gear in a part-task desk-top computer simulation might cost only one thousandth as much as training the skill in a full-motion/visual scene flight simulator. However, if the single-task training does not transfer to the multi-task situation, then the part-task training is of little benefit. Since most of the compensatory activities for dealing with high workload are not present in single-task situations, training a single component may have limited utility.

Training single tasks under high workload may have substantial benefits over single-task training for transfer to multi-task situations. Of the seven compensatory activities described above, all except converting intermessage interference develop under most high workload situations and do not require the exact messages to be sent for compensatory activities to develop. For example, any dual-task situation will encourage the learner to delay tasks, adopt low-workload strategies, develop time multiplexing skills, shorten transmissions, and chunk transmissions.

Within the current architecture, training under high workload is critical, but training in the full task may not be. Let us illustrate this by assuming the goal is to teach four tasks A,B,C,D. If the tasks are trained individually, appropriate compensatory skills for dealing with high workload will not develop. Hence, single-task training should be limited to insuring that the individual tasks be fast and reliable. After the individual tasks are trained, dual tasks should be practiced to develop the compensatory activities, e.g., train A,B,C,D then AB,CD, then ABCD. The dual-task combinations should be chosen such that tasks that can be integrated, or need to convert the interference of other tasks' transmissions (due to need for concurrent transmission) are practiced together. During dual-task practice compensatory activities develop that transfer to other multi-task situations. For example, training during dual-task AB may develop the A task such that automatic transmissions occur in chunks transmitted in short bursts multiplexed at a fixed rate. The resulting A transmissions produce little load either on attentional processing or on innerloop message traffic. Therefore, the A skill developed in the AB training condition should transfer well to multi-task situations, e.g., AC, AD, ABC, ABD, and ABCD.

The current modeling suggests that part-task trainers should be multi-task trainers. We speculate that most of the training time required to develop high skill levels involves practice moving the skill from the goal-state-maintained phase (Phase 3) to the automatic processing phase (Phase 5). To accomplish this, secondary-task loading is critical. The high workload can be produced either via presenting a calibrated workload task or by concurrently practicing other high workload tasks. A calibrated workload task might be a varied mapping auditory search task that requires considerable attention and innerloop transmission but does not improve with practice (see e.g. Fisk, Derrick & Schneider, 1987). Let us refer to this as task X. The training simulators would train A, B, C, D, AX, BX, CX, DX, then ABCD. Such part-task training simulators would cost about the same as the single-task trainers and might produce substantially more transfer. Perhaps, it would be more efficient to build multi-task trainers so the learner would use the practice time to develop skill on task-relevant procedures. The training simulators would train A, B, C, D, AB, CD, then ABCD, with the bulk of the training time being in the AB and CD dual-task training. It is likely that part-task trainers that can train multiple tasks will be far less expensive to produce than full-system simulators. Research is required to determine the effectiveness of such trainers and to develop guidelines for task analysis and the division of tasks across training devices.

Quantitative modeling of skill acquisition

There is a critical need for developing and testing quantitative models of skill acquisition with emphasis on multi-task performance. Most previous modeling of high workload performance has been at too coarse a level of analysis to have had a strong impact on the training process. We feel that general resource theories (e.g., Kahneman, 1973; Wickens, 1984; Navon & Gopher, 1980) have neither dealt explicitly with practice, or differentiated the resources to a level of detail to suggest guidelines for training. An analogy to economics illustrates our concern. Macro-economic theory at the level of predicting GNP has had a very limited success in predicting economic shifts or providing business managers data to make production decisions. In contrast, linear programming techniques that predict production costs as a function of specific resources, e.g., the cost of ice-cream as a function of the cost of sugar, milk, chocolate, allow managers to evaluate alternative configurations of training time and training devices to build better skills for a fixed total cost of resources.

Models need to specify: 1) What types and quantities of resources exist; 2) How the resources are utilized to accomplish specific tasks; and 3) How resource utilization changes with practice. The connectionist/control architecture illustrates the beginning of such a model. The resources involve the number and kinds of modules, attention switching, transmission time on the innerloop, number and strength of connections, etc. The computer simulation can perform specific tasks such as visual search and acquisition of simple digital troubleshooting skill. Resource utilization changes dramatically as skill is acquired. Initially performance (Phases 1-3) is slow, serial, and effortful, e.g., very attention demanding. With practice, automaticity develops and performance becomes fast, parallel, and requires little effort, e.g., no attention and little innerloop transmission time.

Modeling efforts should emphasize cognitive architectures rather than single models. A cognitive architecture identifies a space of models rather than an individual model (see J. R. Anderson, 1983; Laird, Rosenbloom, & Newell, 1986). The present connectionist/control architecture defines such a space of models. Within this architecture there may be many possible individual models, e.g., postulating different connection patterns among modules on the innerloop provides a family of related models. [15]

Model predictions should be compared with human data to tune the modeling effort. The model's predictions of practice data should be compared to human skill acquisition data. The models should be able to predict the entire practice function. In any modeling effort of this type there are many parameters and possible configurations within the architecture. Empirical data are needed to determine the appropriate constants in the model. It is possible that physiological data may provide suggestions as to what connective patterns to explore (see e.g., Mishkin & Appenzeller, 1987). The modeling should predict practice data for a variety of tasks and training procedures.

[15] Schneider is developing a software library to run simulations within connectionist/control architecture on IBM AT or PS2 computers. The programs will be made available for research and instructional purposes to facilitate exploration of the modeling space.

Summary

Review of the multi-task training and the part-whole task literature shows that performance on consistent tasks changes dramatically with practice. Single-task training can transfer to multi-task performance; however, that transfer can be very limited and dual-task training produces substantial performance improvement even after extended single-task training. Existing theoretical frameworks for multi-task performance generally do not predict the observed limited single- to dual-task transfer effects.

A connectionist/control architecture for skill acquisition and multi-task training effects provides an interpretation of the limited nature of single to dual-task transfer. The model details interactions at the microlevel (neural like interactions), macrolevel (module interactions), and the system level (regions of processing and control structures). The model identifies five phases of skill acquisition including: 1) controlled comparison; 2) context-maintained controlled comparison; 3) goal-state maintained controlled comparison; 4) controlled-assist of automatic processing; and 5) automatic processing.

As a skill progresses through the five phases, there is a qualitative change in processing that enables performance of multiple tasks. During the controlled-comparison phases (Phases 1-3) performance is slow, serial, and very effortful. During Phase 1, dual-task time sharing is very error prone. By Phase 3, dual tasking is possible as long as the tasks are accomplished sequentially, e.g., performing task A, then task B, without overlapping processes. A qualitative shift in processing occurs by Phase 4, when associative retrieval of the response substitutes for the controlled comparison during Phases 1-3. Performance is fast, parallel, i.e., retrieval is fairly independent of the number of comparisons, requires little effort, and utilizes little attentional or innerloop processing. Phase 4 skills can be processed concurrently with other tasks as long as there is sufficient controlled processing capacity to briefly assist the transmission of messages and to transmit messages on the innerloop. A Phase 5 skill can be performed reliably without the aid of controlled processing and requires only small periods of innerloop time for transmitting messages from input to output regions. Phase 5 skills (automatic processing) can be performed concurrently as long as the two tasks do not require the same modules and can time share innerloop transmission time or transmit non-interfering messages.

Seven compensatory activities occur in the model during multi-task training that either do not appear in single-task training or are not as critical in single-task situations: 1) task shedding, delay and buffer pre-loading; 2) letting go of high workload strategies; 3) utilizing noncompeting resources; 4) time multiplexing; 5) shortening transmissions; 6) converting interference from concurrent transmissions; and 7) chunking transmissions. The development of these compensatory activities provides an interpretation of the large practice effects served in dual-task situations even after extensive single-task training.

The connectionist/control architecture model of skill acquisition and multi-task performance suggests an extensive research agenda of basic and applied issues relating to skill acquisition for high workload tasks. It emphasizes that models of dual-task performance must deal with issues of practice. Researchers should not ask simply whether single-task training transfers to dual-task performance or whether part- versus whole-task training is better. Rather, research should map out quantifiable performance variables assessing the marginal utility of practice and predicting the optimal points to shift from single- to multi-task performance. The learning and capacity of compensatory activities that develop during dual-task training must also be investigated. Understanding these issues might greatly facilitate development and use of part-task trainers for developing high performance multi-task skills.

The present connectionist/control architecture is just beginning to deal with the complexity of practice effects during skill acquisition. We need to enrich the set of quantitative models we have available to understand and predict skill acquisition. Human performance changes dramatically with practice, thus making variables critical initially become less important or irrelevant after practice. Any model of high workload performance that does not deal with practice effects is at best an approximation to the system, that through practice, produces a variety of knowledge structures and compensatory activities to perform multi-task skills. Allowing basic and applied researchers to conceptualize and predict the effects of practice will advance the understanding of skill and the training of high performance skills.

Acknowledgments

This work was sponsored by the Army Research Institute, under contract No. MDA903-86-C-0149 and Personnel and Training Research Programs Psychological Sciences Division, Office of Naval Research under Contract Nos. N-0014-86-K-0107 and N-00014-86-K-0678.

References

Allport, D.A., Antonis, B., & Reynolds, P. (1972). On the division of attention: A disproof of the single channel hypothesis. Quarterly Journal of Experimental Psychology, 24, 225-235.

Anderson, J. A. (1983). Cognitive and psychological computation with neural models. IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, 799-815.

Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Annett, J., & Kay. H. (1956). Skilled performance. Occupational Psychology, 30, 112-117.

Baddeley, A. D., Grant, S., Wight, E., & Thomson, N. (1974). Imagery and visual working memory. In P. M. A. Rabbit & S. Dornic (Eds.), Attention and performance, VIII. Hillsdale, NJ: Erlbaum.

Bahrick, H.P., & Shelly, C. (1958). Time sharing as an index of automatization. Journal of Experimental Psychology, 56, 288-293.

Blum, M. L., & Naylor, J. C. (1968). Industrial psychology. New York: Harper and Row.

Broadbent, D. E. (1958). Perception and communication. Oxford, MA: Pergamon Press.

Damos, D. L. (1980). Residual attention as a predictor of pilot performance. Human Factors, 20, 435-440.

Damos, D.L., Bittner, A.C., & Kennedy, R.S., & Harbeson, M.M. (1981). Effects of extended practice on dual-task tracking performance. Human Factors, 23, 627-631.

Damos, D., & Lintern, G. (1981). A comparison of single- and dual-task measures to predict simulator performance of beginning student pilots. Ergonomics, 24, 673-684.

Damos, D., & Wickens, C.D. (1980). The identification and transfer of time-sharing skills. Acta Psychologica, 46, 15-39.

Downey, J. E., & Anderson, J. E. (1915). Automatic writing. The American Journal of Psychology, 26, 161-195.

Fisk, A. D., Derrik, W.L. & Schneider, W. (1986-87).*Current Psychological Research & Reviews, 5,* 315-327.

Fisk, A.D., & Schneider, W. (1983). Category and word search: Generalizing search principles to complex processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9*, 177-195.

Hirst, W., Spelke, E.S, Reaves, C.C., Caharack, G., & Neisser, U. (1980). Dividing attention without alternation or automaticity. *Journal of Experimental Psychology: General*, *109*, 98-117.

Hunt, E., & Lansman, M. (1986). Unified model of attention and problem solving. *Psychological Review*, *93*, 446-461.

James, W. (1890/1983). *The principles of psychology*. Cambridge, MA: Harvard University Press.

Johnson, D.F., & Haygood, R.C. (1984). The use of secondary tasks in adaptive training. *Human Factors, 26*, 105-108.

Kahneman, D. (1973). *Attention and effort*. Englewood, NJ: Prentice-Hall.

Kelley, C. R. (1969). What is adaptive training? *Human Factors*, *11*, 43-64.

Kerr, B. (1973). Processing demands during mental operations. *Memory & Cognition*, *1*, 401-412.

Klapp, S. T., Marshburn, E. A., & Lester, P. T. (1983). Short-term memory does not involve the "working memory" of information processing: The demise of a common assumption. *Journal of Experimental Psychology: General, 112,* 240-264.

Laberge, P., & Samuels, S. J. (1974). Toward a theory of automatic information processing in reading. *Cognitive Psychology*, *6*, 293-323.

Laird, J., Rosenbloom, P., & Newell, A. (1986). *Universal subgoaling and chunking: The automatic generation and learning of goal hierarchies*. Boston, MA: Kluwer.

Lintern, G., & Gopher, D. (1978). Adaptive training of perceptual-motor skills: Issues, results, and future directions. *International Journal of Man-Machine Studies, 10*, 521-551.

McGeoch, G. O. (1931). Whole-part problem. *The Psychological Bulletin*, *28*, 713-739.

McGeoch, J. A., & Irion, A. L. (1952). The psychology of human learning. 2nd Ed. New York: Longmans, Green, & Co.

Mishkin, M., & Appenzeller, T. (1987). The anatomy of memory. Scientific American, 256, 80-89.

Mishkin, M., Malamut, B., & Bachevalier, J. (1984). Memories and habits: Two neural systems. In G. Lynch, L. McGaugh, & N. M. Weinberger (Eds.), Neurobiology of learning and memory. New York: The Guilford Press.

Moray, N. (1984). Attention to dynamic visual displays in man-machine systems. In R. Parasuraman & D. R. Davies (Eds.) Varieties of attention. Orlando, FL: Academic Press.

Moray, N. (1986). Monitoring behavior and supervisory control. In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.), Handbook of perception and human performance. Vol. II. New York: John Wiley & Sons. Inc.

Navon, D., & Gopher, D. (1980). Task difficulty, resources, and dual-task performance. In R. S. Nickerson (Ed.), Attention and performance. Vol. VIII. Hillsdale, NJ: Erlbaum.

Naylor, J. C., & Briggs, E. G. (1963). Effects of task complexity and task organization on the relative efficiency of part and whole training methods. Journal of Experimental Psychology, 65, 217-224.

Norman, D. A., & Shallice, T. (1985). Attention to action: Willed and automatic control of behavior. In R. J. Davidson, G. E. Schwartz, & D. Shapiro (Eds.), Consciousness and self-regulation: Advances in research. Vol. IV. New York: Plenum Press.

North, R.A., & Gopher, D. (1976). Measures of attention as predictors of flight performance. Human Factors, 18, 1-14.

Pew, R. W. (1974). Human perceptual motor performance. In B. H. Kantowitz (Ed.), Human information processing: Tutorials in performance and cognition. Hillsdale, NJ: Erlbaum.

Rieck, A. M., Ogden, G. D., & Anderson, N. S. (1980). An investigation of varying amounts of component-task practice on dual-task performance. Human Factors, 22, 373-384.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds), Parallel distributed processing: Explorations in the microstructure of cognition Volume 1: Foundations. Cambridge, MA: MIT Press.

Schneider, W. (1985). Training high-performance skills. Human Factors, 27, 285-300.

Schneider, W. (1987). Connectionism: Is it a paradigm shift for psychology? Behavior Research, Methods, Instruments and Computers, 19, 73-83.

Schneider, W. & Detweiler, M. (1986). Changes in performance in workload with training. In Proceedings of the Human Factors Society 30th Annual Meeting. (1128-1132). Santa Monica, CA: Humans Factors Society.

Schneider, W., & Detweiler, M. (1987). A connectionist/control architecture for working memory. In G. H. Bower (Ed.), The psychology of learning and motivation. Vol. 21). Orlando, FL: Academic Press.

Schneider, W., & Mumme, D. (1987). A connectionist control architecture for attention, automaticity and the capturing of knowledge. Manuscript for submission.

Schneider, W., Dumais, S.T., & Shiffrin, R.M. (1984). Automatic processing and attention. In: R. Parasuraman, R. Davies, & R.J. Beatty (Eds.), Varieties of attention. New York, N.Y.: Academic Press.

Schneider, W., & Fisk, D. (1982). Degree of consistent training: Improvements in search performance and automatic process development. Perception & Psychophysics, 31, 160-168.

Schneider, W., & Fisk, A.D. (1982). Concurrent automatic and controlled visual search: Can processing occur without resource cost? Journal of Experimental Psychology: Learning, Memory, and Cognition, 8, 261-278.

Schneider, W., & Shiffrin, R.M. (1977). Controlled and automatic human information processing: I. Detection, search, and attention. Psychological Review, 84, 1-66.

Senders, J. (1983). Visual scanning processes. The Netherlands: The University of Tilburg Press.

Shaffer, L.H. (1975). Multiple attention in continuous verbal tasks. In P.M.A. Rabbitt & S. Dornic (Eds.) Attention and performance. V. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Shiffrin, R.M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending and a general theory. Psychological Review, 84, 127-190.

Solomons, L., & Stein, G. (1986). Normal motor automatism. Psychological Review, 3, 492-512.

Spelke, E., Hirst, W., & Neisser, U. (1976). Skills of divided attention. Cognition, 4, 215-230.

Stammers, R. B. (1982). Part and whole practice in training for procedural tasks. Human Learning, 1, 185-207.

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. Journal of Experimental Psychology, 18, 643-662.

Underwood, G. (1974). Moray vs. the rest: the effects of extended shadowing practice. Quarterly Journal of Experimental Psychology, 26, 368-372.

Wickens, C. (1980). The structure of attentional processes. In R. Nickerson (Ed.) Attention and performance. VIII. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Wickens, C. D. (1984a). Processing resources in attention. in R. Parasuraman & D. R. Davies (Eds.), Varieties of attention. New York, N.Y.: Academic Press.

Wickens, C. D. (1984b). The multiple resources model of human performance: Implications for display design. In AGARD Conference Proceedings No. 371. Human factors considerations in high performance aircraft.

Wickens, Fracker, & Webb (1988). This issue.

Wightman, D.C., & Lintern, G. (1985). Part-task training for tracking and manual control. Human Factors, 27, 267-283.

Figure Captions

Figure 1. Microlevel structure of the CAP1 simulation. Processing is assumed to occur in networks of neural-like units. Units are organized into modules (the box labeled M3 outlines the third module) that process a particular class of inputs. Information between modules is transferred as a message vector (MV) on fibers connecting the output of one module to the input of the next. In the diagram information flows from left to right (e.g., the top left MV might encode visual features, the two left modules letters, and the right module words). Each module contains a vector of output units. The output units receive input from other modules and connect autoassociatively to themselves. The recurrent connections from the bottom of each output unit going up and connecting to the other output units in the same module represent the autoassociative connection. Each of the crossing points above the output units (to message vector or autoassociative fibers) represents an associative connection that can change the strength of connection with learning. In the rest of the diagram the reverse arrow-type connections represent excitatory influences and the flat connections represent inhibitory influences. A module's output is controlled by an attenuation unit within the module. The attenuation unit regulates information flow from the module. Each module's activity is regulated by a control structure (the box labeled C3 represents the control structure for the third module). Each module reports its activity to the lower-level control structure via activity report and priority report units. The lower units (labeled 1, 2, 3) illustrate a potential control circuit. Cell 1 receives the activity reports from the module and inhibits the activity of neighboring modules. Cell 2 inhibits Cell 3, reducing the attenuation activation, thus reducing inhibition of the output units, and enabling a message vector to transmit. Cell 2 is assumed to habituate resulting in a burst of output and sequential switching or attention. The AP box in M3 illustrates the local circuit for automatic processing. For an automatic process, the priority report unit inhibits the attenuation unit and causes the vector to transmit from the module.

Figure 2. System-level description of the model. The top portion of the figure (2A) shows the message vector connections between regions. The bottom portion (2B) shows the macrolevel view of some of the regions. The squares and rectangles in Figure 2B represent the modules and control structures, e.g. Figure 1 module M3, control C3. This is a top-down view of the regions of processing within the system. Each region represents a series of processing levels. The first or last level of a region (last level for input regions and first for output regions) is assumed to input to the innerloop of connections between regions. The modules on the innerloop have separate message vectors to each of the other modules they connect to. All the lines in Figure 2A represent message vectors (as MV in Figure 1). Each module sends a message vector to all the other modules on the innerloop. The output for the visual module is highlighted to illustrate this connection pattern. This figure represents a simple view of one of many possible connection patterns for regions on the innerloop. Figure 2B illustrates the processing of a visual word to produce a button press. The first level of the visual region processes letters, the second characters, and the third words. The message is then transmitted to all modules on the innerloop. The dotted set of modules represent all the

other regions on the innerloop receiving the visual message. The motor regions illustrate the motor output of the system. The first level of the motor system stores motor tasks, e.g. K1 represents the code for pressing the first key. The second motor level codes the sequences required to execute the motor task, e.g L-lift finger, M-move to position of key, and P-press key. The third motor level illustrates the components of the lift sequence, e.g. A-accelerate upward, W-wait some time, and D-decelerate. The darker horizontal lines represent message vectors, the thin lines the control signals. Each module (square) sends an ACTIVITY report to its controller (rectangle below it) and receives a FEEDBACK and TRANSMIT control signal (the three lines from the squares to the rectangles). Control modules exchange LOAD and NEXT signals (diagonal lines between rectangles) to control sequential processing between modules. For modules on the innerloop, the Central Control exchanges control signals for ACTIVITY, RESET, TRANSMIT, NEXT and, LOAD to modulate innerloop message traffic (see Schneider & Detweiler, 1987).

Figure 3. Timing diagrams of the system as a function of phase of skill acquisition in a category search task. The bottom line provides the time scale in seconds. **Phases 1-3** involve multiple transmissions and monitoring of messages. The elevated bars in line a show the transmission of the vectors from visual module 1 (V1) or 2 (V2), e.g the transmission of "CAT" as in Figure 2B. The two visual modules would alternate between transmitting two words, e.g. "TOP" from V1 and "CAT" from V2. Line b shows the transmission of the lexical vectors for the words stored in the lexical buffers, e.g. "FRUIT" and "ANIMAL" of L1, L2. Line c shows the power or activity report of the cells in the semantic module. The activity increases during transmission due to summation and feedback effects within the module. The first wave is for the sum of the V1 and L1 vectors, e.g. the sum of the semantic codes for "TOP" + "FRUIT". If the activity is below the match threshold, the next pair is transmitted, e.g. "CAT" + "FRUIT". When the visual vectors have all been sent, attention is switched to the next lexical module, e.g., "ANIMAL", the visual module pointer is reset and the V1 + L2 comparison occurs. This continues until either there is a match or the last comparison is complete. The last wave in line c illustrates a match, "CAT" + "ANIMAL", i.e., a high activity report. After a match, the "YES" response is transmitted from the motor module (line d) and initiates a motor response. **Phase 4** occurs after associations are built up between the visual input and motor response, such that the target "CAT" associatively evokes the "YES" response and bypasses the semantic match process found in Phases 1-3. Two transmissions are still needed (line e) from the visual to the motor region to prevent the V1 and V2 messages from interfering. Line f shows the received motor activity from the visual transmissions. Line **g** shows the transmission of the motor response after it was received. The visual-to-motor associations were built up during Phases 1-3. **Phase 5**, automatic processing, occurs after priority learning occurs. The vectors with a high priority tag are transmitted automatically (dashed pulse line h) without the need for attention. This transmission evokes the response (line i) that produces an automatic response transmission (line h).

Figure 4. Compensatory activities developed during dual-task training. The

rectangular pulses indicate transmitted vectors, the waves the received information. Initially transmitting the visual (line 1) and auditory (line 2) vectors in parallel results in little information reception in the motor (line 3) or speech modules (line 4). Time multiplexing the signals (lines 5 and 6) results in greater reception with some delay of the second message (lines 7 and 8). Dual-task training can shorten transmissions (lines 9-12) and convert interference (lines 13-16) (see text for details).

Figure 5. Chunking transmissions for the output of the letters "T", "H", "E" as in a typing task. Initially the letters are transmitted sequentially from three visual modules to three motor modules (lines 1-6). After interference conversion occurs (see text), all three messages can be transmitted as a set (lines 7-12). If the visual system develops a chunk code for "THE", it can be transmitted from a single visual module (line 13) and evoke a motor code for all three outputs (line 14). Lines 15-22 show transmissions within the motor region that do not occur on the innerloop (see Figure 2B for levels in the motor region). After the message is received, the level 1 motor module transmits a vector (line 15) to the level 2 modules and evokes the "T", "H", and "E" code in three modules (lines 16-18), decoding the "THE" chunk at motor level 1. The second level sequentially transmits the "T", "H", "E" (lines 19-21) and evokes the letters in the level 3 module (line 22). The level three module then decodes the movements of each letter just as the level 2 module decoded the letters for each word. When the level 3 module returns a NEXT signal (see Figure 2B and text), the next transmission outputs from the level 2 modules.

Fig 1.

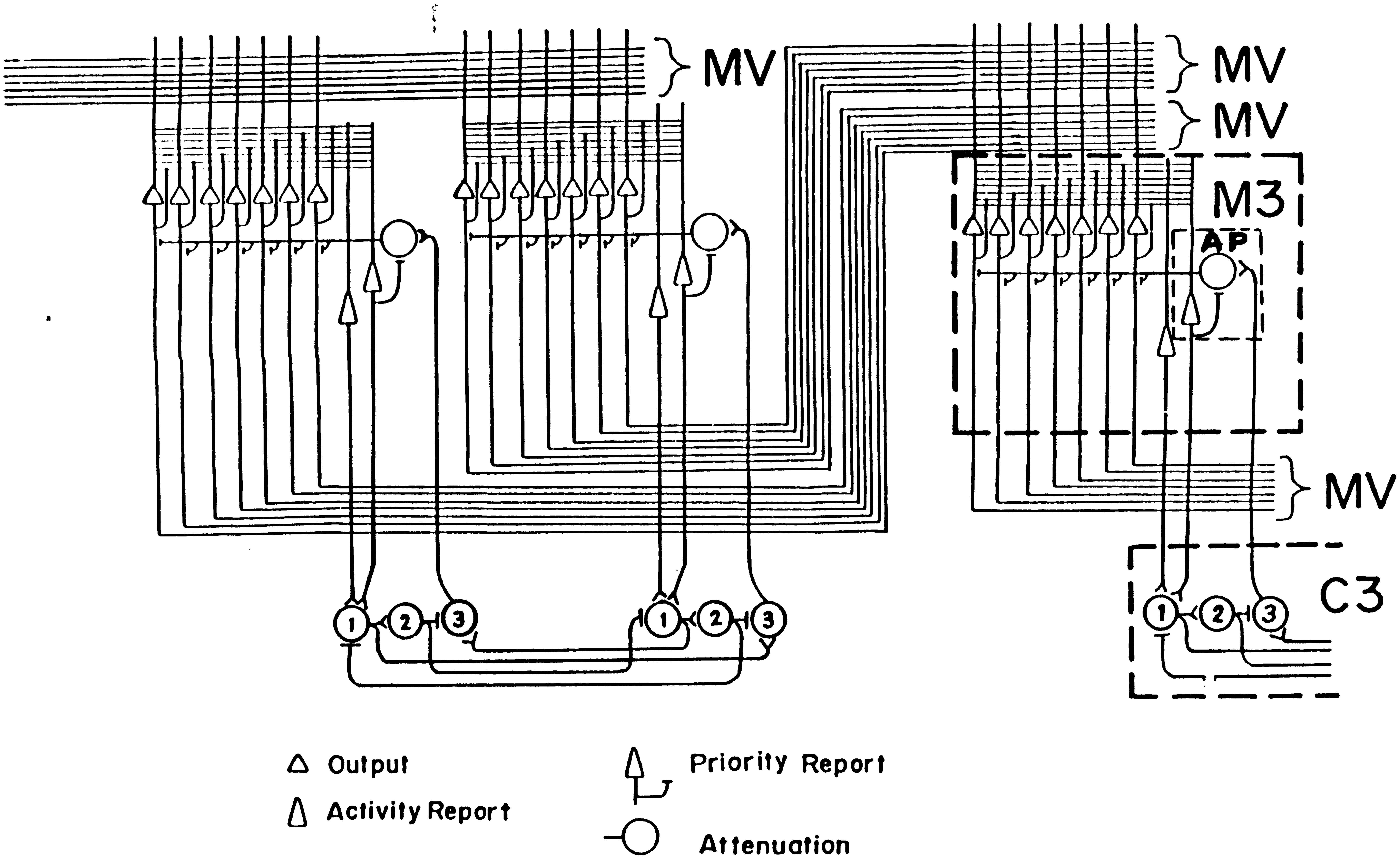
MV
MV
MV
M3
AP
MV
1
2
3
1
2
3
1
2
3
C3
Output
Activity Report
Priority Report
Attenuation

Fig. 1

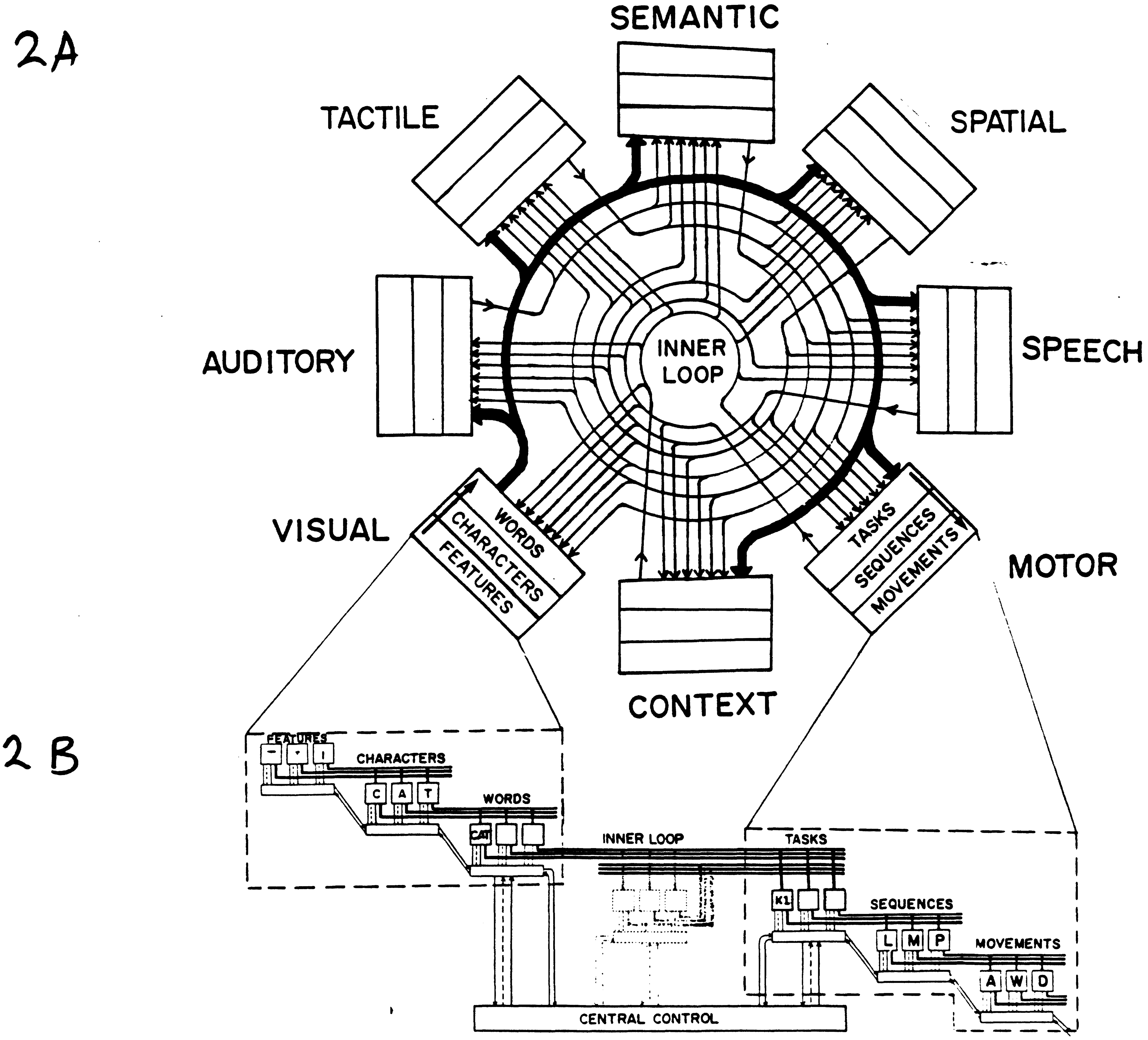
2A
SEMANTIC
TACTILE
SPATIAL
AUDITORY
INNER LOOP
SPEECH
VISUAL
WORDS
CHARACTERS
FEATURES
TASKS
SEQUENCES
MOVEMENTS
MOTOR
CONTEXT
2 B
FEATURES
CHARACTERS
C
A
T
WORDS
CAT
INNER LOOP
TASKS
K1
SEQUENCES
L
M
P
MOVEMENTS
A
W
D
CENTRAL CONTROL

Fig. 2

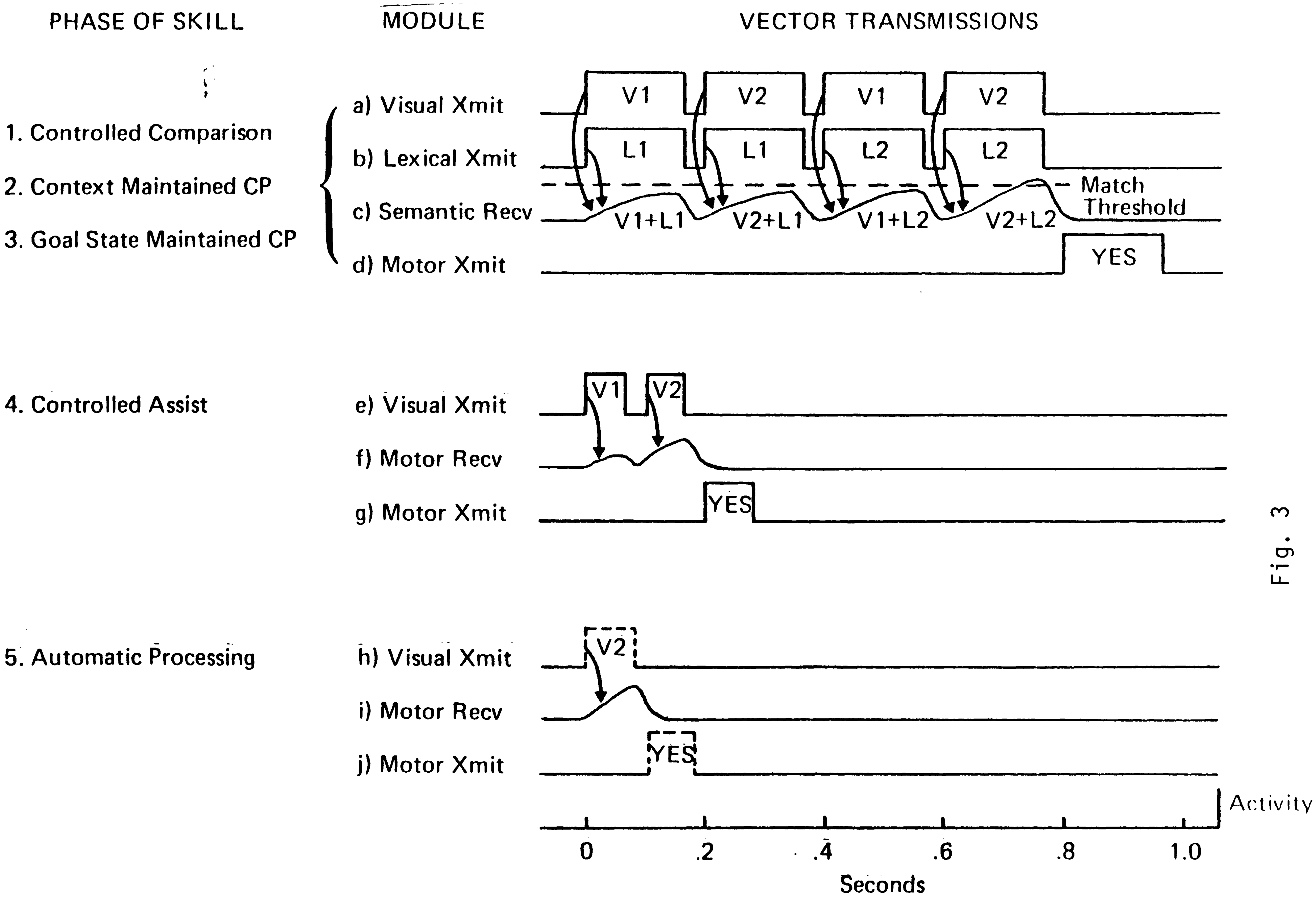
PHASE OF SKILL
MODULE
VECTOR TRANSMISSIONS
1. Controlled Comparison
2. Context Maintained CP
3. Goal State Maintained CP
a) Visual Xmit
b) Lexical Xmit
c) Semantic Recv
d) Motor Xmit
V1
V2
V1
V2
L1
L1
L2
L2
V1+L1
V2+L1
V1+L2
V2+L2
Match Threshold
YES
4. Controlled Assist
e) Visual Xmit
f) Motor Recv
g) Motor Xmit
V1
V2
YES
5. Automatic Processing
h) Visual Xmit
i) Motor Recv
j) Motor Xmit
V2
YES
Activity
0
.2
.4
.6
.8
1.0
Seconds

Fig. 3

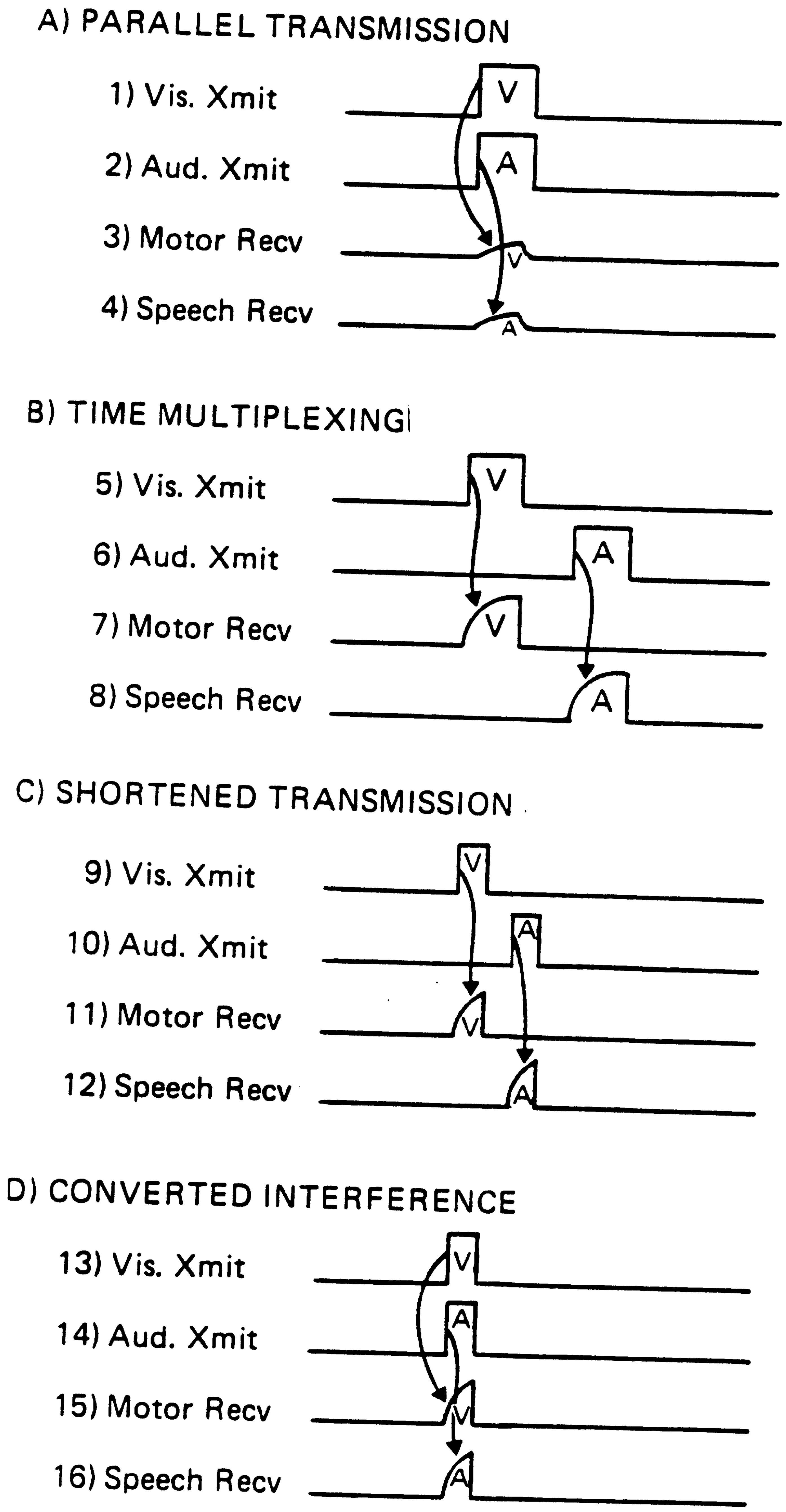
A) PARALLEL TRANSMISSION
1) Vis. Xmit
2) Aud. Xmit
3) Motor Recv
4) Speech Recv
V
A
V
A
B) TIME MULTIPLEXING
5) Vis. Xmit
6) Aud. Xmit
7) Motor Recv
8) Speech Recv
V
A
V
A
C) SHORTENED TRANSMISSION
9) Vis. Xmit
10) Aud. Xmit
11) Motor Recv
12) Speech Recv
V
A
V
A
D) CONVERTED INTERFERENCE
13) Vis. Xmit
14) Aud. Xmit
15) Motor Recv
16) Speech Recv
V
A
V
A

Fig. 4

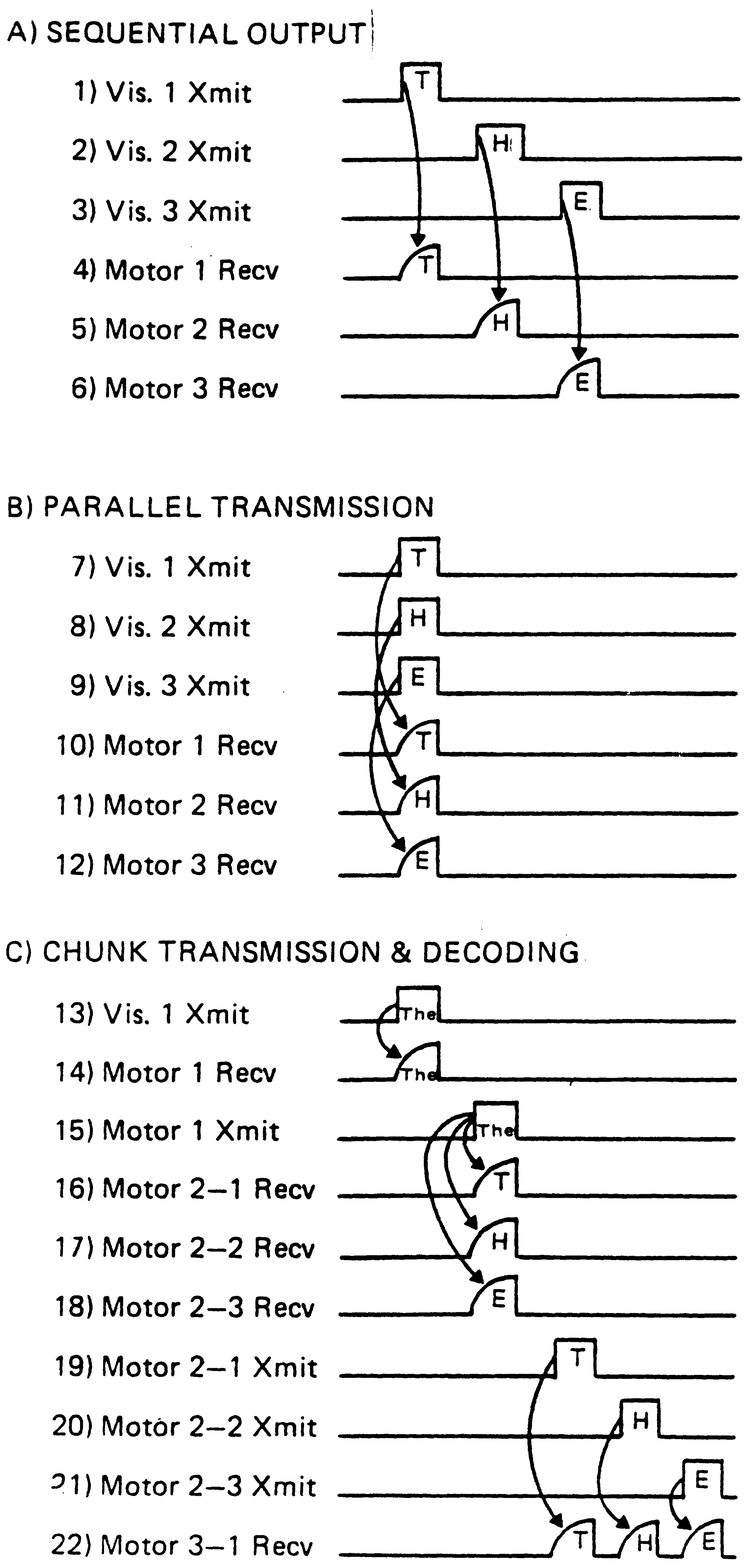
A) SEQUENTIAL OUTPUT
1) Vis. 1 Xmit
2) Vis. 2 Xmit
3) Vis. 3 Xmit
4) Motor 1 Recv
5) Motor 2 Recv
6) Motor 3 Recv
T
H
E
B) PARALLEL TRANSMISSION
7) Vis. 1 Xmit
8) Vis. 2 Xmit
9) Vis. 3 Xmit
10) Motor 1 Recv
11) Motor 2 Recv
12) Motor 3 Recv
C) CHUNK TRANSMISSION & DECODING
13) Vis. 1 Xmit
14) Motor 1 Recv
15) Motor 1 Xmit
16) Motor 2–1 Recv
17) Motor 2–2 Recv
18) Motor 2–3 Recv
19) Motor 2–1 Xmit
20) Motor 2–2 Xmit
21) Motor 2–3 Xmit
22) Motor 3–1 Recv
The

Fig. 5