Working Papers in Speech Recognition

- II -

CMU Computer Science Speech Group
August, 1973

This report contains three previously published papers and two unpublished ones:

D. Raj Reddy, Lee D. Erman, and Richard B. Neely, "A Model and a System for Machine Recognition of Speech", **IEEE Trans. Audio and Electroacoustics,** AU-21 (3), June, 1973.

D.R. Reddy, L.D. Erman, R.D. Fennell, and R.B. Neely, "The HEARSAY Speech Understanding System: An Example of the Recognition Process", 3rd Inter. Joint Conf. on Artificial Intelligence, Stanford, Ca, Aug, 1973.

L.D. Erman, R.D. Fennell, V.R. Lesser, and D.R. Reddy, "System Organizations for Speech Understanding: Implications of Network and Multiprocessor Computer Architectures for AI", 3rd Inter. Joint Conf. on Artificial Intelligence, Stanford Ca, Aug, 1973.

Janet MacIver Baker, "A New Time-Domain Analysis of Human Speech", Apr, 1973.

James Baker, "Machine-Aided Labeling of Connected Speech", Apr, 1973.

The research reported here was partially supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-73-C-0074) and monitored by the Air Force Office of Scientific Research.

A Model and a System for Machine Recognition of Speech

D. RAJ REDDY, LEE D. ERMAN, and RICHARD B. NEELY

Abstract—**This paper presents a model for machine recognition of connected speech and the details of a specific implementation of the model, the HEARSAY system. The model consists of a small set of cooperating independent parallel processes that are capable of helping in the decoding of a spoken utterance either individually or collectively. The processes use the "hypothesize-and-test" paradigm. The structure of HEARSAY is illustrated by considering its operation in a particular task situation: voice-chess. The task is to recognize a spoken move in a given board position. Procedures for determination of parameters, segmentation, and phonetic descriptions are outlined. The use of semantic, syntactic, lexical, and phonological sources of knowledge in the generation and verification of hypotheses is described. Preliminary results of recognition of some utterances are given.**

Introduction

Most papers on speech recognition conclude by saying that it is necessary to use higher level linguistic cues to obtain acceptable recognition. The terms context, syntax, semantics, and phonological rules are used but attempts to utilize these sources of knowledge have not been successful because of the ill structuredness of these concepts. This paper represents a summary of several years of investigation to formulate an information processing model that would lead to efficient recognition of speech and in which the role of various sources of knowledge would be well defined.

At the 1969 spring meeting of the Acoustical Society, we presented several papers on the structure of a speech recognition system that was used to recognize a list of 500 isolated words and a syntax-directed connected speech-recognition system using a finite state grammar and a 16-word vocabulary (Vicens [37], Reddy [31], Neely [22]). Six amplitude and zero-crossing parameters of the incoming utterance were sampled every 10 ms and segmented. The segments were labeled to specify the phonetic class; the syntax was used for sentence analysis and word boundary determination, and prelearned acoustic and phonetic segmental descriptions of lexical items were used for word recognition.

Several inherent limitations were apparent even as we developed the system. First, the vocabulary had to be reduced to 16 words because of word boundary ambiguity problems. For example, the word "large" had to be changed to "big" because of assimilation of the reduced vowel of "the" into the semivowel /l/ of "large" in the utterance: "Pick up the large block."

Second, we had to overcome the limitations of the syntax-directed methods. One could not blindly parse from left to right; rather, we had to locate anchor points from which parsing could proceed both backwards and forwards. This was necessary to compensate for machine errors in earlier stages and to compensate for the idiosyncrasies in speaker performance such as introduction of spurious words, repetition of words, and inclusion of hmm- and ha-like sounds.

Third, the simple hierarchical structure in which output from one process forms the input to the next was not adequate for the task. Errors introduced in each process tend to have multiplicative effect, i.e., if each of four processes introduced 10 percent errors, the cumulative error would be 34 percent. Further, the lack of feedback and feedforward of the simple hierarchical model meant any errors that got through were uncorrectable. The main virtue of the system was that it was the first demonstrable system to use syntactic and lexical constraints to recognize connected speech sentences (such as: "Pick up the big block at the bottom right corner").

For the past four years the authors have been attempting to develop a model and a system for connected speech recognition that did not suffer from the limitations mentioned previously, and that would serve as a research tool for speech-recognition research over a wide range of tasks. The following sections present the resulting model and an outline of the system implemented on a PDP-10 computer.

The Model

We were interested in developing a system capable of recognition of connected speech from several speakers with graceful error recovery, in close to real time, and easily generalizable to operate in several different task domains. We started with several requirements for the model.

1) Contributions of syntax, semantics, context, and other sources of knowledge towards recognition should be clearly evaluatable. Exactly what and how much does each contribute towards improving the performance of the system?

2) The absence of one or more sources of knowl-

Manuscript received April 30, 1972. This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract F44620-70-C-0107 and monitored by the Air Force Office of Scientific Research.

D. R. Reddy and L. D. Erman are with the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa. 15213.

R. B. Neely was with the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa. 15213. He is now with the Xerox Palo Alto Research Center, Palo Alto, Calif. 94505.

edge should not have a crippling effect on the performance of the model. That semantic context should not be essential for perception is illustrated by overheard conversations among strangers. That syntactic or phonological context should not be essential is illustrated by conversations among children. That lexical representation is not essential is illustrated by our recognition of new words and nonsense syllables.

3) When more than one source of knowledge is available, interactions between them should lead to a greater improvement in performance than is possible to attain by the use of any subset of sources of knowledge.

4) Since the decoding process is errorful at every stage, the model must permit graceful error recovery.

5) Increases in performance requirements, such as the real time requirement, increase in vocabulary, modifications to the syntax, or changes in semantic interpretation, should not require major reformulation of the model.

The model we have arrived at to satisfy these requirements consists of a small set of cooperating independent processes capable of helping in the decoding process either individually or collectively and using the "hypothesize-and-test" paradigm.

Each of the processes in our model is based on a particular source of knowledge, e.g., syntactic, semantic, or acoustic-phonetic rules. Each process uses its own source of knowledge in conjunction with the present context (i.e., the presently recognized subparts of the utterance) in generating hypotheses about the unrecognized portions of the utterance. This mechanism provides a way for using (much talked about but rarely used) context, syntax, and semantics in the recognition process.

The notion of a set of independent parallel processes, each of which is capable of generation and verification of hypotheses, is needed to satisfy the requirements 1) and 2) mentioned previously. In our model, the absence of a source of knowledge implies deactivating that process, and recognition proceeds (albeit more slowly and with lower accuracy) using the hypotheses generated by the remaining processes. The independence of the processes permits us to deactivate a source of knowledge and measure how and by how much that source of knowledge improves the system.

The need for parallel processes can be derived from the real-time performance requirement. If the system is to ever approach human performance, it must be able to answer trivial questions as soon as they are uttered (some times even before they are completed). This implies that various processes of the system should be able to operate on the incoming data as soon as they are able to do so without waiting for the completion of the whole utterance (as in a simple hierarchic model). The "coroutine" model, in which each process passes control to the next level when a "chunk" is perceived and regains control when a new chunk is needed, would be satisfactory. But this organization can lead to irrevocable loss of data if a higher level process does not return control in time to process new chunks of incoming speech. Thus, there must be at least two parallel processes, one of which is continuously monitoring the input speech and the other proceeding with recognition. This, in addition to requirements 1) and 2), suggests a model with parallel processes.

An important aspect of the model is the nature of cooperation between processes. The implication is that, while each of the processes is independently capable of decoding the incoming utterance, they are also able to cooperate with each other to help recognize the utterance faster and with greater accuracy. Process "A" can guide and/or reduce the hypothesis generation phase of process "B" by temporarily restricting the parts of the lexicon that can be accessed by B, or by restricting the syntax available to process B, and so on. This assumes that process A has additional information that it can effectively use to provide such a restriction. For example, in a given syntactic or semantic situation only a small subset of all the words of a language may appear.

The need for a hypothesize-and-test paradigm arises from 4). The "errorful" nature of speech processing at every stage implies that every source of knowledge has to be brought to bear to resolve ambiguities and errors at every stage of processing. This implies rich connectivity among various processes and involves both feedforward and feedback. The hypothesize-and-test paradigm represents an elegant way of obtaining this cooperation in a uniform manner.

The notion of hypothesize-and-test is not new. It has been used in several artificial intelligence programs (Newell [25]). It is equivalent to analysis-by-synthesis (Halle and Stevens [10]) if the "test" consists of matching the incoming utterance with a synthesized version of the hypothesis generated. In most cases, however, the test is of a much simpler form; for example, it is not necessary to generate the whole formant trajectory when a simpler test of the slope can provide the desired verification. This not only has the effect of reducing the computational effort but also increases the differentiability between phonemically ambiguous words.

Extendability and generalizability of the model is mainly an issue of implementation. It requires that representation of sources of knowledge be separate from and independent of mechanisms that operate on them. One way of achieving this is to represent the knowledge in a form most suitable for modification by the user and have a set of preprocessors that then transform the knowledge into the representation required by the system.

HEARSAY System

HEARSAY is a speech-recognition system that incorporates many of the ideas presented in the previous section and is presently under development at Carnegie-Mellon University. It is not restricted to any particular recognition task. Given the syntax and the vocabulary of a language and the semantics of a task, HEARSAY will attempt recognition of utterances in that language.

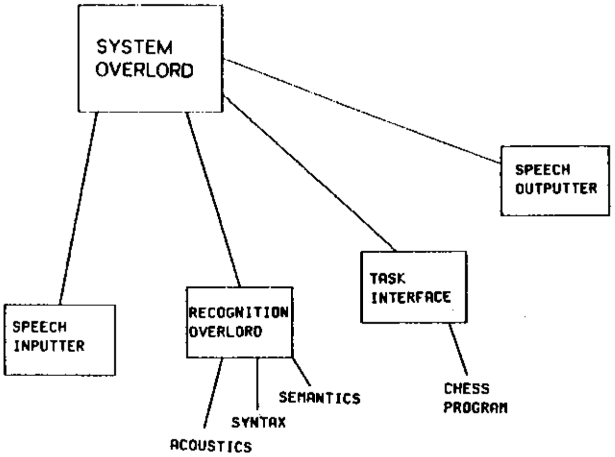


Fig. 1. Overview of the HEARSAY system.

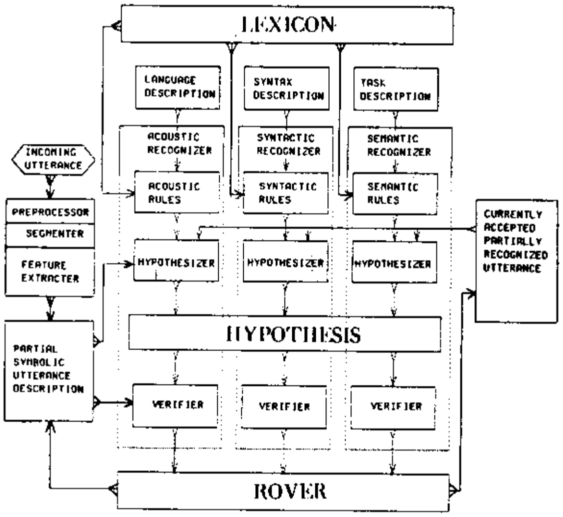


Fig. 2. Detail of the recognition process.

Fig. 1 gives an overview of the HEARSAY system. The EAR module accepts speech input, extracts parameters, and performs some preliminary segmentation, feature extraction, and labeling, generating a "partial symbolic utterance description." The recognition overlord (ROVER) controls the recognition process and coordinates the hypothesis generation and verification phases of various cooperating parallel processes. The TASK provides the interface between the task being performed and the speech recognition and generation (SPEAK-EASY) parts of the system. The system overload (SOL) provides the overall control for the system. A more detailed, but earlier, description of the goals and various components of this system are given in Reddy *et al.* [33] and Reddy [32].

Here we will describe the operation of the HEARSAY system by considering a specific task: voice-chess. The task is to recognize a spoken move in a given board position. In any given situation there are generally 20-30 legal moves and several thousand different ways of expressing these moves. The syntax, semantics, and vocabulary of the task are restricted, but the system is designed to be easily generalizable to larger tasks, which was not the case for our earlier systems. Larger syntax (e.g., a subset of English) and vocabularies (1000-5000 words) for a more complex semantic task will make HEARSAY slower and less accurate but are not likely to be crippling.

Fig. 2 shows the recognition process in greater detail. At present, it contains three independent processes: acoustic, syntactic, and semantic. We will give a short description of how these processes cooperate in recognizing "king bishop pawn moves to bishop four." Let us assume that this is a legal move (otherwise, at some stage of processing, the system will reject it as semantically inconsistent).

Parametric Level Analysis

The speech from the input device (microphone, telephone, or tape recorder) is passed through five octave bandpass filters (spanning the range 200-6400 Hz) and an unfiltered band. Within each band the maximum intensity and the number of zero crossings are measured for every 10-ms interval.

This results in a vector of 12 parameters every 10 ms. These parameters are smoothed and log transformed and a subset of the parameters is chosen for further processing. Fig. 3 gives the parameters used, at present, for part of the utterance "king bishop pawn . . . ." Each column represents a 10-ms time unit. Rows $P1$, $P2$, $P3$, and AU represent the log-amplitude parameters in the frequency bands 200-400, 400-800, 800-1600 Hz, and the unfiltered band, respectively. The amplitudes are quantized to 32 levels and represented as a single character (blank, 0-9, A-U, and *, which represents a value greater than 31). Rows $P4$ and $P5$ represent values that are functions of both amplitude and zero crossing in bands 1600-3200 and 3200-6400. Details of various operations on these parameters are given in Erman [6].

This vector of parameters ($P1$-$P5$ and AU) are compared with a standard set of parameter vectors to obtain a minimum distance classification for each time unit using a highly modified version of a procedure proposed by Astrahan [1]. The row labeled *PP* gives the classification for each 10-ms unit. The standard

```
           k      ɪ         ŋ      b      l       ʃ         ə       p         ɔ         n
         .    :     .     :     .     :     .     :     .     :     .     :     .     :     .     :
P1         BMLJJKKKJJJJJJJJJKKKKJCBEPSSTTTRPMDDA73200 LNOMG96311         8GGHIJJLLLMNNNNMMLKKKKKKF62
P2         6FHHFEEEDDDDDDDCB97622  DLLLLLJIEC72220    2DJHD51       00 BKOSRRRSRRQQRRRQQPPNHHFDC9731
P3               122232223200        13331              1650           AFJKLMNOQQQPQRRRRQNHA6631
P4    8HFHNT******RKLLMMNMKIHC      J******PMLNPPOMMJ7OPOL20            9CKLMNQSUUUUUTPPOONMMLO22
P5    9IJJC 2 2   8                           269FGEGGFE6                              21
AU    20025LNONPPPMLLMMLLLKJIHGF61BLQSSSSQLIEDEDCB9 2BLNI0600          CNQQRRRSSSSTTUTSSQOMLLLKJF84
PP   -FCC$$ellIeeeINNNNNINNNNYMMDdVNIeeeeeIIIII$$$$SF+INIUD+++++--++-daaaaaaeeeeeeeeeAAee&NN&UWdd+
SP   -CCC$$IIIIIIIINNNNNNNNNNNNMMdddIIeeeeeIIIII$$$$+++NNN+++++++--++--aaaaaaeeeeeeeeeeeeNNNNWWdd+
VF    .ffffvvvvvvvvvvvvvvvvvvvvvvv-vvvvvvvvvvvvvvffff...vvv............vvvvvvvvvvvvvvvvvvvvvvvvvv.
```

Fig. 3. Parameters and segmentation for "king bishop pawn · · · ." *P*1-*P*5 and *AU* (amplitude) are the input parameters. *PP* is the phone-like name given to the segment. *SP* is the locally smoothed *PP*. *VF* is a segmentation based on the *SP*'s: · unvoiced, nonfricated; *f* unvoiced, fricated; *v* voiced, nonfricated; and *z* voiced, fricated.

set of parameters is obtained by selecting cluster centers from a training set of utterances containing various phonemes in neutral contexts. When a phoneme is represented by several articulatory gestures, more than one cluster center may be added to the standard set. Speaker characteristics and the noise characteristics of the environment or the transducer may be reflected in the standard set of clusters by recording the training set in that environment. Fig. 4 gives cluster centers for several representative sounds. A complete list of clusters used and the details of the speaker normalization program are given in Erman [6].

Remark 1: The labels in row *PP* of Fig. 3 are not to be confused with phonetic transcription. Accurate phonetic transcription, where possible, would require modifying the labels taking into account segment and sentence level context.

Remark 2: If one wanted to use formant frequencies and amplitudes (assuming they can be determined without mislabeling) one would reanalyze the training set for this parametric representation to determine the new cluster centers. Representing the parameters as a vector with a weighted distance metric defined on the vector space is all that is needed to use a new parametric representation in the HEARSAY system. There are several disadvantages to this approach, e.g., errors in labels, inability to take advantage of special features of a parametric representation, etc. However, this approach provides a convenient way of obtaining the best first approximation to the phonetic representation.

Remark 3: The tendency is to blame every error on inadequate parametric representations. We have gone from one set of amplitude and zero crossing parameters to three sets and now to five. Others divide the frequency range into 12, 17, 24, 32, and 48 regions or the full resolution given by FFT. The increase in noisiness of the parameters with increasing resolution makes it imperative that one transform the high resolution data to a smaller number of robust parameters such as the efforts by Li *et al.* [16] and Pols [28] in dimensionality reduction of spectra.

Remark 4: The parameters we use represent a crude spectrum. A mixed strategy in which finer analysis is performed only when necessary (Reddy [30]) seems more appropriate for an efficient realization of the system than obtaining every possible parameter at the start.

PP	P1	P2	P3	P4	P5	AU
d	22	14	5	8	0	18
s	0	0	0	47	39	9
■	30	10	2	2	0	33
u	43	30	11	7	0	39
a	37	62	44	38	0	59

Fig. 4. Several typical *PP*-cluster centers.

Remark 5: Spectral representation appears to be more robust than formant representation because of the likelihood of mislabeling a formant.

Remark 6: Parcor parameter representation (Itakura and Saito [14]) has also been used successfully (Nakano *et al.* [21]) and may have efficient machine realizations within the framework of the HEARSAY system.

Remark 7: Zero-crossing measurements and formant frequency measurements are more prone to error than energy measurements in a noisy environment. It appears more difficult to devise noise subtraction algorithms for frequency than for amplitude (Neely and Reddy [24]).

Segmentation

The purpose of segmentation is to divide the continuous parameter sequence into discrete phone-size chunks. This is usually based on an acoustic similarity measure (Reddy and Vicens [34]). Labeling every 10-ms unit by a phone-like cluster name permits the segmentation to be divided in terms of these labels. Fig. 3 shows two levels of segmentation for "king bishop pawn . . . ." The first level is derived by doing a local "smoothing" of the *PP* names assigned to each of the 10-ms segments; this is displayed on the row labeled *SP*. A segment is defined to be a contiguous run of a single *PP*, flanked by *PP*'s not the same as those in the run. This segmentation is approximately at the phoneme level but is, by itself, very unreliable.

A second level of segmentation is derived by associating a voiced/unvoiced decision and a fricated/nonfricated decision with each *PP*. These binary decisions, when applied to the *SP*'s (and modified with a few simple rules for smoothing and breaking of long segments according to significant local amplitude peaks), segment the signal very reliably. The row in Fig. 3 labeled *VF* indicates this segmentation for the sample.

Remark 1: It is now commonly agreed among all researchers that some form of segmentation of acoustic signals is necessary for connected speech recognition (see Fant and Lindblom [8], Reddy [29], Denes and von Keller [4], Broad [2], Medress [19], Dixon and Tappert [5], Klatt and Stevens [15], Stalhammar and Karlsson [35], Hemami and Lehiste [11]). No systematic evaluation has been made of these and other methods of segmentation that have been proposed or implemented. Our present view is that almost any of the schemes, given enough careful tuning, will work in a large majority of the cases; the more important question is then not how to segment, but rather how to use the segmentation without being crippled by the inevitable errors.

Remark 2: This use of segmentation represents a trend away from segmentation-free recognition schemes (Halle and Stevens [10]). However, segmentation-free recognition still seems to be a useful concept if one is mainly interested in isolated word recognition (Hill [12], White [39]).

Acoustic Recognizer

The role of the acoustic recognizer is to predict and verify syllables and words based on the features present in the incoming utterance, the present context, and the lexicon. The structure and phonetic description of syllables and words in the lexicon is prespecified. An entry for a word in the lexicon contains the phonemic spelling(s) of the word and annotations that are used to describe expected anomalies that cannot be predicted by rule from the phonemic spelling. A more detailed description of the lexicon and the preprocessing is given in Erman [6].

The acoustic recognizer has three sources of knowledge available for the generation and verification of hypotheses: acoustic, phonological, and vocabulary restrictions. The acoustic knowledge appears in the form of expected parameters (or features) for a phoneme in a neutral context. The phonological knowledge appears in the form of a coarticulation model that modifies the expected features based on context. The between-word coarticulation effects have to be determined wherever applicable through the use of the "currently accepted partially recognized utterance" (Fig. 2), which provides the boundary phonemes. The vocabulary restriction appears in the form of a valid subset of words in the lexicon that contain a given sequence of features.

The acoustic recognizer uses these sources of knowledge in two stages: the hypothesis and the verification. The acoustic hypothesizer does not have any knowledge of the syntax or semantics of the situation, but can use the gross features (such as /ʃ/ of "bishop") in the "partial symbolic utterance description" (Fig. 2) to retrieve those words of the lexicon that are consistent within the features present.

The task of a verifier is to determine whether a given hypothesis is consistent with the context presently available to it. For example, let us assume that alternative hypotheses of the words "king's," "pawn," "bishop," "queen's," and "knight" have been made in the context "king ··· pawn · · ·" (where "···" represents the hypothesized words) and that the word actually spoken was "bishop." Detailed verification, by the acoustic verifier, of every phoneme of every option word is not necessary. All that is needed, in this example, are some simple tests that notice that there is a strong fricative indicated near the middle of the area of interest, which causes "pawn" and "knight" to be rejected, and some other simple tests on the vowel portion, e.g., duration, high/low, and front/back, which would indicate that both "queen's" and "king's" are unlikely, whereas "bishop" is highly likely.

A more detailed matching of features and the use of coarticulation rules at the word boundaries may, of course, be needed for other cases. Detailed matching often implies generation of a test. For example, if the verification to be made is among "sit," "spit," and "split," the presence of /s/, /I/, /t/ and the transitions between /I/ and /t/ are irrelevant. What is needed is the test for the presence or absence of a stopgap and for the presence of /l/-like formant structure following the stopgap.

Remark: That some form of hypothesization and verification is needed seems to be recognized by many researchers at this point. Halle and Stevens [10] proposed synthesis and match as a means of verification in their analysis-by-synthesis model. Hypothesis and verification for isolated word recognition was used in the Vicens–Reddy system (Vicens [38]). More recently, similar techniques have also been used by Klatt and Stevens [15], Lindblom and Svensson [18], Tappert *et al.* [36], and Itahashi *et al.* [13].

Syntactic Recognizer

The role of the syntactic recognizer is to predict phrases based on the syntactic structure of the language to be recognized and the context. The predicted phrases induce (specify) words that might appear in that context. The grammar for the voice-chess language is context free. The voice-chess grammar, specified as a set of BNF productions, is given in Fig. 5. For example, in this grammar, "<move>" is defined to be either "<move1>" followed by "<checkword>" or "<move1>." The total number of different utterances permitted by this grammar is about five million.

The role of the syntax hypothesizer is to use the syntactic source of knowledge to predict words. In

```
1.  <move>          ::= <move1> <check-word> | <move1>

2.  <move1>         ::= <regular-move> | <capture> | <castle>

3.  <castle>        ::= <castle-word> ON <uniroyal> SIDE
                        | <castle-word> <uniroyal> SIDE
                        | <castle-word>

4.  <regular-move> ::= <man-loc> <move-word> <square>

5.  <capture>       ::= <man-loc> <capture-word> PAWN EN-PASSENT
                        | <man-loc> <capture-word> <man-loc>

6.  <castle-word>  ::= CASTLE | CASTLES

7.  <move-word>     ::= TO | MOVES-TO | GOES-TO

8.  <capture-word> ::= TAKES | CAPTURES

9.  <check-word>    ::= CHECK MATE | CHECK

10. <man-loc>       ::= <man-spec> ON <square> | <man-spec>

11. <man-spec>      ::= <uniroyal> <unipiece> PAWN
                        | <uniroyal> <piece> | <uniroyal> pawn
                        | <unipiece> pawn | <man>

12. <square>        ::= <uniroyal> <piece> <rank> | <nopawn> <rank>

13. <man>           ::= KING | QUEEN | BISHOP | KNIGHT | ROOK | PAWN

14. <uniroyal>      ::= KING | QUEEN | KING'S | QUEEN'S -

15. <unipiece>      ::= BISHOP | KNIGHT | ROOK
                        | BISHOP'S | KNIGHT'S | ROOK'S

16. <nopawn>        ::= KING | QUEEN | BISHOP | KNIGHT | ROOK

17. <piece>         ::= BISHOP | KNIGHT | ROOK

18. <rank>          ::= ONE | TWO | THREE | FOUR
                        | FIVE | SIX | SEVEN | EIGHT
```

Fig. 5. Voice-chess syntax.

hypothesization the syntax recognizer uses only very local context to predict words. Predictions may be made either to the right or the left of already existing words. For example, if "--- moves-to ---" is given, then words may be hypothesized to the left of "moves-to" or to the right of "moves-to." Hypothesization uses only inexpensive methods, and often generates words that would not fit in the complete context of the sentence.

Traditional parsing schemes are not very useful in generating hypotheses. Further, the syntax recognizer must be capable of processing errorful strings containing spurious words and repetition of words. This implies that it must be capable of working both forwards and backwards. This is achieved in HEARSAY by the use of antiproductions.

CENTER	LEFT	RIGHT	HEAD
CASTLE	↑	↑	<castle-word>
CASTLES	↑	↑	<castle-word>
EN-PASSENT	PAWN	↑	<capture>
ON	<castle-word>	<uniroyal>	<castle>
PAWN	<capture-word>	EN-PASSENT	<capture>
SIDE	<uniroyal>	↑	<castle>
SIDE	<uniroyal>	↑	<castle>
<move1>	↑	<check-word>	<move>
<move1>	↑	↑	<move>
<check-word>	<move1>	↑	<move>
<regular-move>	↑	↑	<move1>
<capture>	↑	↑	<move1>
<castle>	↑	↑	<move1>
<castle-word>	↑	ON	<castle>
<castle-word>	↑	<uniroyal>	<castle>
<castle-word>	↑	↑	<castle>
<uniroyal>	ON	SIDE	<castle>
<uniroyal>	<castle-word>	SIDE	<castle>
<man-loc>	↑	<move-word>	<regular-move>
<man-loc>	↑	<capture-word>	<capture>
<man-loc>	↑	<capture-word>	<capture>
<man-loc>	<capture-word>	↑	<capture>
<move-word>	<man-loc>	<square>	<regular-move>
<square>	<move-word>	↑	<regular-move>
<capture-word>	<man-loc>	PAWN	<capture>
<capture-word>	<man-loc>	<man-loc>	<capture>

Fig. 6. Antiproductions for a subset of the syntax of Fig. 5. (The subset consists of productions 1-6.)

Antiproductions act as a concordance for the grammar giving all the contexts for every symbol appearing in the grammar. They are used to predict words that are likely to occur following or preceding a word using only limited context. Fig. 6 gives antiproductions for productions 1-6 of the grammar of Fig. 5. These are produced automatically by a preprocessing program. In this figure, the symbols in the column labeled CENTER are the entries in the concordance. Each symbol in the subset of the grammar appears in this column once for each occurrence of it in the subset. The entries in the LEFT and RIGHT columns denote symbols that can appear to the left and right of the entry in the center column. When an ↑ appears in the LEFT or RIGHT column, it indicates that the original production did not have an entry to the left or right of that symbol.

When the LEFT (or RIGHT) context given in an antiproduction is satisfied, then the RIGHT (or LEFT) context is hypothesized for recognition. If the hypothesized symbol happens to be a nonterminal, then all the possible terminal symbols that can appear at the left of this nonterminal are hypothesized. Detailed descriptions of the structure and use of antiproductions will be given in Neely [23].

The role of the syntactic verifier is to accept or discard hypotheses using syntactic consistency checks. This is usually a more expensive process than hypothesization because it involves complete parsing of the partially recognized sentences. The verifier may work both on hypotheses that the syntactic hypothesizer has generated, as well as those generated by other hypothesizers.

Semantic Recognizer

The role of the semantic recognizer is to predict concepts based on the semantics of the task and semantics of the preceding utterance. A predicted concept (a legal move for voice-chess) is used in conjunction with the present context to predict a word that might appear in the utterance. The semantics of the task and the preceding utterances are captured for chess by the current board position. The board position for the utterance in discussion, "king bishop pawn moves to bishop four," is shown in Fig. 7.

HEARSAY has, as a subpart, a chess program (Gillogly [9]) that generates an ordered list of moves that are possible in that situation. A partial list of legal moves with numbers representing the likelihood of occurrence is given in Fig. 8.

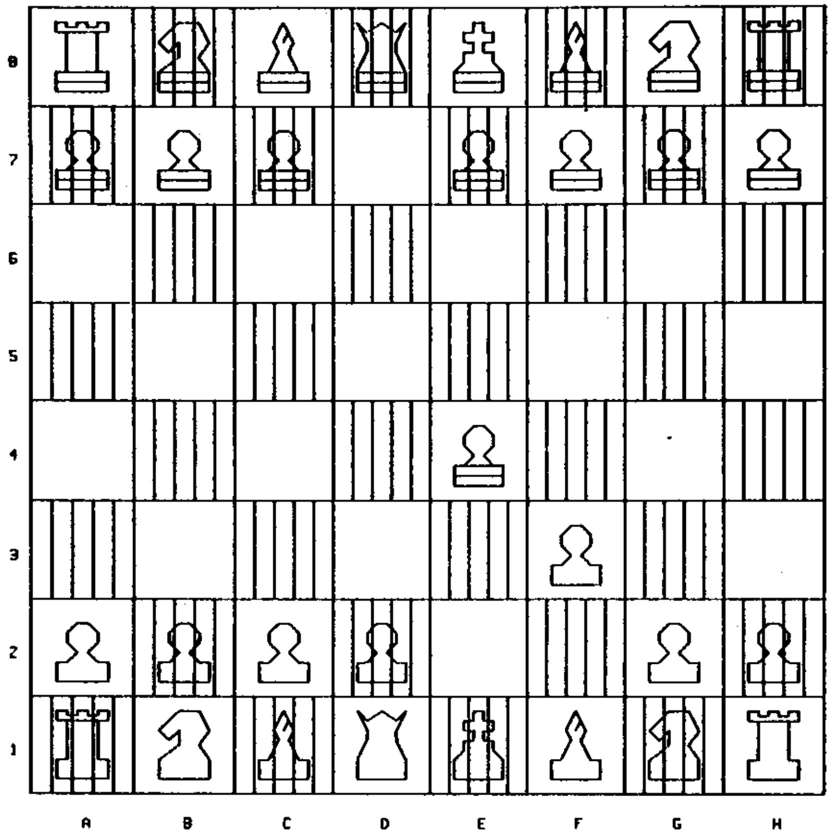


Fig. 7. Board position for utterance in discussion.

```
KBP/KB3XKP/K4     100
QP/Q2-Q4           50
QN/QN1-QB3         49
KB/KB1-QB4         48
KN/KN1-K2          47
QP/Q2-Q3           46
KB/KB1-K2          45
Q/Q1-K2            44
QBP/QB2-QB4        43
QBP/QB2-QB3        42
K/K1-KB2           41
K/K1-K2            40
KRP/KR2-KR4        39
KNP/KN2-KN4        38
QNP/QN2-QN4        37
QRP/QR2-QR4        36
KN/KN1-KR3         35
KNP/KN2-KN3        34
QNP/QN2-QN3        33
KRP/KR2-KR3        32
QRP/QR2-QR3        31
QN/QN1-QR3         30
KB/KB1-QN5CH       25
KBP/KB3-KB4        24
KB/KB1-QR6         12
KB/KB1-Q3           6
```

Fig. 8. Ordered list of legal moves supplied by the chess-playing program for the board position of Fig. 7.

The semantic hypothesizer uses the ordered list of moves for hypothesis generation. In our example the hypothesizer would concentrate only on the "noncapture" moves that start with the word "king." If there are none, then there is an inconsistency in the currently accepted partially recognized utterance. This may be due to an illegal statement or incorrect recognition. In the latter case, the partially recognized utterance is modified by replacing the weakest link by the second best choice for that position.

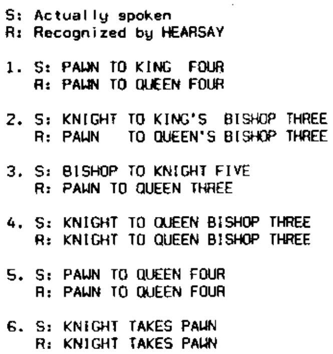
```
       bishop      62
       knight      62
       bishop's    44
       rook        41
       on          41
       knight's    38

ds hypothesized by semantic hy
```

Fig. 9. Words hypothesized by semantic hypothesizer.

Fig. 9 gives the words hypothesized by the semantic hypothesizer in the context of "king ---." Associated with each hypothesis is a rating (ranging from 1 to 100) indicating the semantic likelihood of the hypothesis. This likelihood is derived from the likelihoods of the projected legal moves from which the hypotheses are taken, and from intrasentence semantic clues. The semantic hypothesizer uses word- and phrase-level semantic consistency checks to restrict hypothesization. The structure and the mechanism used by the semantic hypothesizer are described in Neely [23].

Control of the Processes

Since the different recognizers are independent, the recognition overlord needs to synchronize the hypothesis generation and verification phases of various



```
S: Actually spoken
R: Recognized by HEARSAY

1. S: PAWN TO KING FOUR
   R: PAWN TO QUEEN FOUR

2. S: KNIGHT TO KING'S BISHOP THREE
   R: PAWN   TO QUEEN'S BISHOP THREE

3. S: BISHOP TO KNIGHT FIVE
   R: PAWN TO QUEEN THREE

4. S: KNIGHT TO QUEEN BISHOP THREE
   R: KNIGHT TO QUEEN BISHOP THREE

5. S: PAWN TO QUEEN FOUR
   R: PAWN TO QUEEN FOUR

6. S: KNIGHT TAKES PAWN
   R: KNIGHT TAKES PAWN
```

Fig. 10. Some preliminary results from one run. (Approximately 4–7 times real-time processing on a PDP-10 computer.)

processes. Synchronization ensures that hypotheses generated by one process will be verified by all the other processes in the subsequent time slice. Several strategies are available for deciding which subset of the processes generates the hypotheses and which verify. At present this is done by polling the processes to decide which process is most confident about generating the correct hypothesis. In voice-chess, where the semantic source of knowledge is dominant, that module usually generates the hypotheses. These are then verified by the syntactic and acoustic recognizers. However, when robust acoustic cues are present in the incoming utterance, the roles are reversed with the acoustic recognizer generating the hypotheses.

The verification process continues until a hypothesis is found that is acceptable to all the verifiers with a high enough level of confidence. All the unverified hypotheses are stored on a stack for the purpose of backtracking at a later stage. Given an acceptable hypothesis, ROVER updates the currently accepted partially recognized utterance and updates the partial symbolic utterance description with additional features that were discovered during the process of hypothesis generation and verification. If the utterance still has unrecognized portions of speech and if the interpretation of the utterance is still unclear, then all the active processes are reactivated to generate hypotheses in the new context. If there are no unrecognized portions of speech in the utterance and the sentence is uninterpretable, the knowledge acquisition part of the system (unimplemented in the present system and not shown in Fig. 2) is activated to update the lexicon and the acoustic, syntactic, and/or semantic rules.

Preliminary Results

The system described in the preceeding sections has been operational since June 1972. We view HEARSAY as a continually evolving system that is expected to serve as a research tool for explorations in speech-recognition research at Carnegie-Mellon University. Fig. 10 gives some preliminary results of recognition by the system. More comprehensive results containing time, accuracy, and error analyses will be given in Erman [6] and Neely [23].

Discussion

Models of Speech Perception

This paper presents a model of speech perception that has been arrived at not so much by conducting experiments on how humans perceive speech but in the process of constructing several speech-recognition systems using computers. The emphasis has been on developing efficient recognition algorithms, with little attention to modeling of known human perceptual behavior. The general framework (for a model) that evolved is different from some previously proposed models by Liberman *et al.*, [17] and Halle and Stevens [10], which imply that perception takes place through the active mediation of motor centers associated with speech production. Our results tend to support "sensory" theories advanced by Fant [7], and others, in which speech decoding proceeds without the active mediation of speech motor centers.

If one eliminates the synthesis part of analysis-by-synthesis, then our model is most similar to that of Halle and Stevens [10]. The important distinction to remember is that once a hypothesis is generated, say of the words "sit," "slit," and "split," one should never want to verify the hypotheses by generating formant trajectories for the word or phrase. That phonemes /s/, /I/, /t/ occur in the hypothesized words is no longer relevant. All that is needed is a verification of the presence of stopgap and the /l/-like formant transition preceding the vowel. Another limitation of synthesis and match is that the noise might swamp the finer distinction required, i.e., the variability in speaker performance of /s/, /I/, /t/ might overshadow the positive contributions of a /p/ or an /l/.

Information-Processing Models

The model proposed in this paper raises several issues that may be of interest to speech scientists and cognitive psychologists interested in human speech perception. We would like to propose that, in addition to stimulus-response studies and neuro-physiological models, speech scientists should also make extensive use of information-processing models in the study of speech perception. The notion of an information-processing model reflects a current trend in cognitive psychology to view man as an information processor, i.e., that his behavior can be seen as the result of a system consisting of memories containing discrete symbols and symbolic expressions and processes that manipulate these symbols (Newell [26]). The main advantage of this approach to speech perception studies is that it permits a researcher to look at the total problem of speech perception at a higher functional and conceptual level than is possible with the other two approaches. (To attempt to study the total problem of speech perception by formulating a neurophysiological model would be like attempting to understand the workings of a TV set by looking at the flow of electrons through a transistor.)

One question that arises in this context is the nature of serial and parallel processing mechanisms used by humans. It is known that, at a higher problem-solving level, a human being behaves essentially as a serial information processor (Newell and Simon [27]). It is also known that parallel processing occurs at the preprocessing levels of vision and speech. What is not known is whether there are several independent processes or a single sophisticated process at the perceptual level that can use effectively all the available sources of knowledge.

The second question is how various sources of knowledge cooperate with each other. There are experiments (Miller and Isard [20], Collins and Quillian [3]) that can be interpreted to show that perception is faster or more intelligible depending on the number of available sources of knowledge. Any model of speech perception must deal with the nature and structure of the interaction between various sources of knowledge. Earlier models tend to ignore this question.

Summary and Conclusions

A casual reader of this paper would probably only notice the superficial aspects of the system: that it accepts voice commands to play chess, uses crude parameters, and is not very smart at using the acoustic-phonetic and other sources of knowledge. That is beside the point. The main contribution of this research is to provide a model and a framework in which the role of phonology, syntax, semantics, and other sources of knowledge can be systematically studied and evaluated. It is no longer necessary for us to be content with vacuous statements about the importance of syntax or semantics.

We chose voice-chess as a task not because it is important to play chess with a computer over telephone, but because chess provides a good area to evaluate our ideas about the role of various sources of knowledge in speech perception. Chess plays the role in our system that the fruit fly plays in genetics. Just as the genetics of *drosophila* are studied not to breed better flies, but to learn the laws of heredity, so we choose chess as a task because the syntax, semantics, and vocabulary of discourse are well defined and are amenable to systematic study.

Similarly, the acoustic parameters and phonological, syntactic, and semantic rules currently used by the HEARSAY system are not particularly important or interesting. What is important to note is that while each module is "stupid," the system still works and does do a creditable job in spite of its weaknesses. The interesting features are the interaction and cooperation among various modules and the correction of errors by various sources of knowledge.

The system described in this paper was demonstrated in June 1972, at a workshop on speech recognition. It represents the first system to demonstrate live, connected speech recognition using nontrivial syntax and semantics. We expect to actively modify the system to greatly increase its performance, as well as use it as an experimental tool for studying speech understanding, recognition, and perception.

Acknowledgment

The authors wish to thank A. Newell whose intellectual and moral support has been essential to sustain a long-term research effort of this kind in the midst of pessimistic predictions. We wish to acknowledge the important contributions of R. Fennell in the design and programming of ROVER and the global data structure. P. Karlton and B. Lowerre also helped in programming parts of the system. B. Broadley, P. Newbury, and J. Teter interfaced the speech devices to the PDP-10. G. Robertson and H. Wactlar made the necessary modifications to the PDP-10 monitor.

References

[1] M. Astrahan, "Speech analysis by clustering or the hyperphoneme method," Dep. Comput. Sci., Stanford Univ., Stanford, Calif., AI Memo 124, 1970.

[2] D. J. Broad, "Formants in automatic speech recognition," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 295–298.

[3] A. M. Collins and M. R. Quillan, "Retrieval time from semantic memory," *J. Verbal Learn. Behav.*, vol. 8, 1969, pp. 204–267.

[4] P. B. Denes and T. G. von Keller, "Articulatory segmentation for automatic recognition of speech," in *Proc. 6th Int. Congr. Acoust.*, vol. B, 1968, pp. 143–146.

[5] N. R. Dixon and C. C. Tappert, "Derivation of phonetic representation by combining steady-state and transemic classification in automatic recognition of continuous speech," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 319–321.

[6] L. D. Erman, Ph.D. dissertation, in preparation.

[7] G. Fant, "Auditory patterns of speech," in *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, Ed. Cambridge, Mass.: M.I.T. Press, 1964.

[8] C. G. M. Fant and B. Lindblom, "Studies of minimal speech sound units," Speech Transmission Lab., Quarterly Prog. Stat. Rep., vol. 2, pp. 1–11, 1961.

[9] J. J. Gillogly, "The TECHNOLOGY chess program," *Artif. Intel.*, vol. 3, pp. 145–163, 1972.

[10] M. Halle and K. Stevens, "Speech recognition: A model and a program for research," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 155–159, Feb. 1962.

[11] H. Hemani and I. Lehiste, "Interactive automatic speech segmentation," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 291–294.

[12] D. R. Hill, "Man-machine interaction using speech," in *Advances in Computers*, vol. 11, F. L. Ait *et al.*, Ed. New York: Academic, 1971, pp. 165–230.

[13] S. Itahashi, S. Makino, and K. Kido, "Automatic recognition of spoken words utilizing dictionary and phonological rule," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 327–330.

[14] F. Itakura and S. Saito, "Speech analysis-synthesis system based on the partial autocorrelation coefficient," presented at the 1969 Acoust. Soc. Jap. Meeting (see also "On the optimum quantization of feature parameters in the parcor speech synthesizer," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 434–437).

[15] D. H. Klatt and K. N. Stevens, "Sentence recognition form visual examination of spectrograms and machine-aided lexical searching," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 315–318.

[16] K.-P. Li, G. W. Hughes, and A. S. House, "Correlation characteristics and dimensionality of speech spectra," *J. Acoust. Soc. Amer.*, vol. 46, pp. 1019–1025, 1969.

[17] A. M. Liberman, F. S. Cooper, K. S. Harris, and P. F. MacNeilage, "A motor theory of speech perception," in *Proc. Speech Commun. Seminar*, vol. 2, 1962.

[18] B. Lindblom and S.-G. Svensson, "Interaction between segmental and non-segmental factors in speech recognition," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 331-333.

[19] M. Medress, "A procedure for the machine recognition of speech," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 113-116.

[20] G. A. Miller and S. Isard, "Some perceptual consequences of linguistic rules," *J. Verbal Learn. Behav.*, vol. 2, pp. 217-228, 1963.

[21] Y. Nakano, A. Ichikawa, and K. Nakata, "Evaluation of various parameters in spoken digits recognition," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 101-104.

[22] R. B. Neely, "Experimental conversational computer system," *J. Acoust. Soc. Amer.*, vol. 46, p. 89(A), 1969.

[23] ——, Ph.D. dissertation, in preparation.

[24] R. B. Neely and R. D. Reddy, "Speech recognition in the presence of noise," in *Proc. 7th Int. Congr. Acoust.* (Budapest, Hungary), vol. 3, 1971, pp. 177-180.

[25] A. Newell, "Heuristic programming: Ill-structured problems," in *Progress in Operations Research*, vol. 3, J. S. Aronofsky, Ed. New York: Wiley, 1971.

[26] ——, "Remarks on the relationship between artificial intelligence and cognitive psychology," in *Non-Numerical Problem Solving*, R. Banerji and M. D. Mesarovic, Ed. Berlin, W. Germany: Springer-Verlag, 1970, pp. 363-400.

[27] A. Newell and H. A. Simon, *Human Problem Solving.* Englewood Cliffs, N.J.: Prentice-Hall, 1972.

[28] L. C. W. Pols, "Dimensional representation of speech spectra," in *Proc. 7th Int. Congr. Acoust.* (Budapest, Hungary), vol. 3, 1971, pp. 281-284.

[29] D. R. Reddy, "Segmentation of speech sounds," *J. Acoust. Soc. Amer.*, vol. 40, pp. 307-312, 1966.

[30] ——, "Computer recognition of connected speech," *J. Acoust. Soc. Amer.*, vol. 42, pp. 329-347, 1966.

[31] ——, "Segment-synchronization problem in speech recognition," *J. Acoust. Soc. Amer.*, vol. 46, p. 89(A), 1969.

[32] ——, "Speech recognition: Prospects for the seventies," in *Proc. IFIP*, vol. 71, 1971, pp. I-5-I-3.

[33] D. R. Reddy, L. D. Erman, and R. B. Neely, "The C-MU speech recognition project," in *Proc. IEEE Syst. Sci. Cybern. Conf.*, 1970.

[34] D. R. Reddy and P. J. Vicens, "A procedure for segmentation of connected speech," *J. Audio Eng. Soc.*, vol. 16, pp. 404-412, 1968.

[35] U. Stalhammar and I. Karlsson, "A phonetic approach to ASR," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 125-128.

[36] C. C. Tappert, N. R. Dixon, and A. S. Rabinowitz, "Application of sequential decoding for converting phonetic to graphemic representation in automatic recognition of continuous speech," in *Proc. Int. Conf. Speech Commun. Processing*, 1972, pp. 322-326.

[37] P. J. Vicens, "Use of syntax in the analysis of connected speech," *J. Acoust. Soc. Amer.*, vol. 46, p. 89(A), 1969.

[38] ——, "Aspects of speech recognition by computer," Dep. Comput. Sci., Stanford Univ., Stanford, Calif., AI Memo 85, Ph.D. dissertation, 1969.

[39] G. White, private communication, Xerox Palo Alto Res. Cen., Palo Alto, Calif., 1972.

Reprinted by permission from **IEEE TRANSACTIONS ON AUDIO AND ELECTROACOUSTICS**
Vol. Au-21, No. 3, June 1973, pp. 229-238

THE HEARSAY SPEECH UNDERSTANDING SYSTEM:
An Example of the Recognition Process

D.R. Reddy, L.D. Erman, R.D. Fennell, and R.B. Neely*

Computer Science Department**
Carnegie-Mellon University
Pittsburgh, Pa. 15213

ABSTRACT

This paper describes the structure and operation of the Hearsay speech understanding system by the use of a specific example illustrating the various stages of recognition. The system consists of a set of cooperating independent processes, each representing a source of knowledge. The knowledge is used either to predict what may appear in a given context or to verify hypotheses resulting from a prediction. The structure of the system is illustrated by considering its operation in a particular task situation: Voice-Chess. The representation and use of various sources of knowledge are outlined. Preliminary results of the reduction in search resulting from the use of various sources of knowledge are given.

Keywords: **speech recognition, understanding, hypothesize-and-test.**

INTRODUCTION

The factors influencing the structure and operation of a speech understanding system are many and complex. The report of Newell et al. (1971) discusses these issues in detail. Our own goals and efforts in this area have been described in several earlier papers (Reddy et al., 1972). The goals for our present effort were outlined in Reddy, Erman, and Neely (1970). The initial structural description of the Hearsay system was given in Reddy (1971). The model and the system that evolved after several design iterations were described in Reddy, Erman, and Neely (1972a).* The main additions to the initial proposed system were in the specification of the interactions among various sources of knowledge. In this paper, we describe the structure and operation of the Hearsay system from a different point of view, i.e., by considering a specific example to illustrate the various stages of the recognition process.

Machine perception of speech differs from many other problems in artificial intelligence in that it is characterized by high data rates, large amounts of data, and the availability of many sources of knowledge. Thus, the techniques that must be employed differ from other problem-solving systems in which weaker and weaker methods are used to solve a problem using less and less information about the actual task. In addition, there is a marked difference in the expectations for system performance. In tasks such as chess and theorem-proving, the human has sufficient trouble himself so as to make reasonably crude programs of interest. But humans perform effortlessly (and with only modest error) in speech or visual perception tasks, and they demand comparable performance from a machine. Thus, it is important that the structure and organization of a system be such that it is not a dead-end effort, i.e., it should be capable of approaching human performance without major reformulation of the problem solution. The Hearsay system effort represents an attempt to produce one such system. The main distinguishing characteristic of this system is that diverse sources of knowledge can be represented as cooperating independent parallel processes which help in the decoding of the utterances using the hypothesize-and-test paradigm.

The system is designed for the recognition of connected speech, from several speakers, with graceful error recovery, performing the recognition in close to real-time. The structure and implementation of the system are to a large extent dictated by these concerns. One feature that characterizes a speech understanding system is the existence of errors at every level of analysis. The errorful nature of processing implies that every source of knowledge has to be invoked to resolve ambiguities and errors at every stage of the processing. One way to accomplish this is through the use of the hypothesize-and-test paradigm, where each source of knowledge can accept, reject, or re-order the hypotheses produced by other sources of knowledge. For example, in the Voice-Chess task, if the word "captures" appears in a partially-recognized utterance, the

* The general framework that evolved for the model is different from some previously proposed models by Liberman et al. (1962) and Halle and Stevens (1962) which imply that perception takes place through the active mediation of motor centers. Our efforts tend to support "sensory" theories advanced by Fant (1964) and others. If one modifies the "synthesis" part of analysis-by-synthesis, then our model is most similar to that of Halle and Stevens.

* Present address: Xerox Palo Alto Research Center, Palo Alto, Ca. 94305.

** This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under contract no. F44620-70-C-0107 and monitored by the Air Force Office of Scientific Research.

semantic source of knowledge can reject all the hypotheses that do not lead to a capture move.

The Hearsay system is not restricted to any particular recognition task. Given the syntax and the vocabulary of a language and the semantics of the task, it attempts recognition of utterances in that language. It is designed to serve as a research tool in which the contributions of various sources of knowledge towards recognition can be clearly evaluated. Since each source of knowledge is represented as an independent process, it can be removed without crippling the system.

Figure 1 gives an overview of the Hearsay system. The EAR module accepts speech input, extracts parameters, and performs some preliminary segmentation, feature extraction and labeling, generating a "partial symbolic utterance description." ROVER (Recognition OVERlord) controls the recognition process and coordinates the hypothesis generation and verification (testing) phases of the various cooperating knowledge processes. The TASK provides the interface between the task being performed and the speech recognition and generation (SPEAK-EASY) parts of the system. SOL, the System OverLord, provides the message communication facilities for the system.

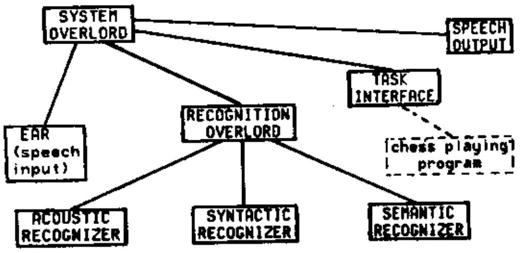


Figure 1: Structure of the Hearsay system.

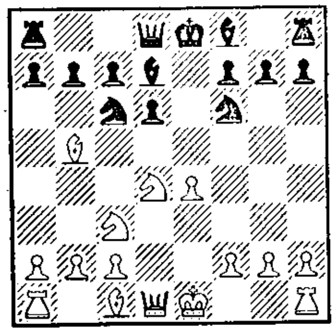
AN EXAMPLE OF RECOGNITION

Here we will illustrate the operation of the Hearsay system by considering in detail the recognition process of an utterance within a specific task environment: Voice-Chess. The task is to recognize a spoken chess move in a given board position and respond with the counter-move.

Figure 2 gives the board position and a list of legal moves in that position at the time the move is spoken. The speaker, playing white, wishes to move his bishop on queen's bishop one to king knight five. This is one of 46 different legal moves. These moves have been ordered on the basis of their goodness in the given board position. This judgment was based on a task-dependent source of knowledge available to the program (Gillogly, 1972). Note that the move chosen by the speaker was only the fourth best move in that situation.

Having chosen the move, there are many possible ways of uttering the move. The syntax of the language permits many variations, usually of the form <piece> <action> <position>. The piece can have qualifiers to indicate the location. The action may be of the form: "to", "moves-to", "goes-to", "takes", "captures", and so on. The position can be of the form: "king three", "king bishop four", or "queen's knight five", and so on. The actual move spoken in this context was "bishop moves-to king knight five". Note that "queen bishop on queen bishop one" can be specified as just "bishop" because there is no ambiguity in this case.

Figure 3 shows the speech waveform of the utterance with manual segmentation, showing the beginning and ending of each word and each phoneme within the word. (The manual segmentation and labeling indicated in this and succeeding figures is for our benefit only -- it is not available to the system while it is attempting recognition.) The utterance was about 2 seconds in duration and the waveform is displayed on ten consecutive rows, each row containing 200 milliseconds of the utterance. The first line of text under each row contains the word being articulated. The word label is repeated for the duration of the word. Thus, the word "bishop" was articulated for 400 milliseconds and occupies the first two rows of the waveform. The second line of text under each row contains the intended phoneme being articulated. The phoneme (represented in IPA notation) is repeated for the duration of the phoneme.



Move	Move
0-0	KNP/KN2-KN3
QB/QB1-K3	QNP/QN2-QN3
QB/QB1-KB4	QRP/QR2-QR3
QB/QB1-KN5	KN/Q4-K2
KB/QN5-QB4	KN/Q4-QN3
KR/KR1-KB1	QN/QB3-Q5
Q/Q1-Q3	KP/K4-K5
KB/QN5KQN/QB6	K/K1-K2
KBP/KB2-KB4	QN/QB3-K2
KR/KR1-KN1	KNP/KN2-KN4
KBP/KB2-KB3	QNP/QN2-QN4
Q/Q1-Q2	QN/QB3-QR4
QR/QR1-QN1	QN/QB3-QN1
K/K1-KB1	QB/QB1-Q2
KB/QN5-K2	KB/QN5-Q3
KN/Q4-KB5	Q/Q1-KB3
KB/QN5-QR4	Q/Q1-K2
KN/Q4XQN/QB6	QB/QB1-KR6
KN/Q4-KB3	K/K1-Q2
KRP/KR2-KR4	KN/Q4-K6
QRP/QR2-QR4	KB/QN5-QR6
KB/QN5-KB1	Q/Q1-KN4
KRP/KR2-KR3	Q/Q1-KR5

Figure 2: The chess board position and the ordered list of legal moves for White.

Several interesting problems of speech recognition arise in the context of recognition of this utterance. The end of Row 2 of Figure 3 shows the juncture between "bishop" and "moves". Note that the ending /p/ in "bishop" and the beginning nasal /m/ in "moves" are homorganic, i.e., they both have the same articulatory position. This results in the absence of the release and the aspiration that normally characterizes the sound /p/. Row 6 of Figure 3 illustrates a word boundary problem. The ending nasal of "king" and the beginning nasal of "knight" tend to be articulated from the same tongue position even though in isolation they would have been articulated from two different positions. This results in a single segment representing two different phonemes in two adjacent words. Further, it is impossible to specify the exact location of the word boundary. In the manual segmentation, the boundary was placed at an arbitrary position. Another type of juncture problem appears on Row 8 of Figure 3 at the boundary of "knight five". The release and aspiration of the phoneme /t/ are assimilated into the /f/ of "five".

Feature Extraction and Segmentation

The speech input from the microphone is passed through five band-pass filters (spanning the range 200-6400 Hz) and through an unfiltered band. Within each band the maximum intensity is measured for every 10 milliseconds (the zero crossings are also measured in each of the bands but they do not play an important role in the recognition process at present). This results in a vector of 6 amplitude parameters every 10 milliseconds. These parameters are smoothed and log-transformed. Figure 4 shows a plot of these parameters as a function of time for part of the utterance of Figure 3. The top line shows the utterance spoken. The second line of text indicates where the word boundaries were marked during the manual segmentation process (this will permit manual verification of the accuracy of the machine recognition process in the later stages).

This vector of parameters (labeled 1, 2, 3, 4, 5, and U in Figure 4) is, for each centisecond, compared with a standard set of parameter vectors to obtain a minimum distance classification

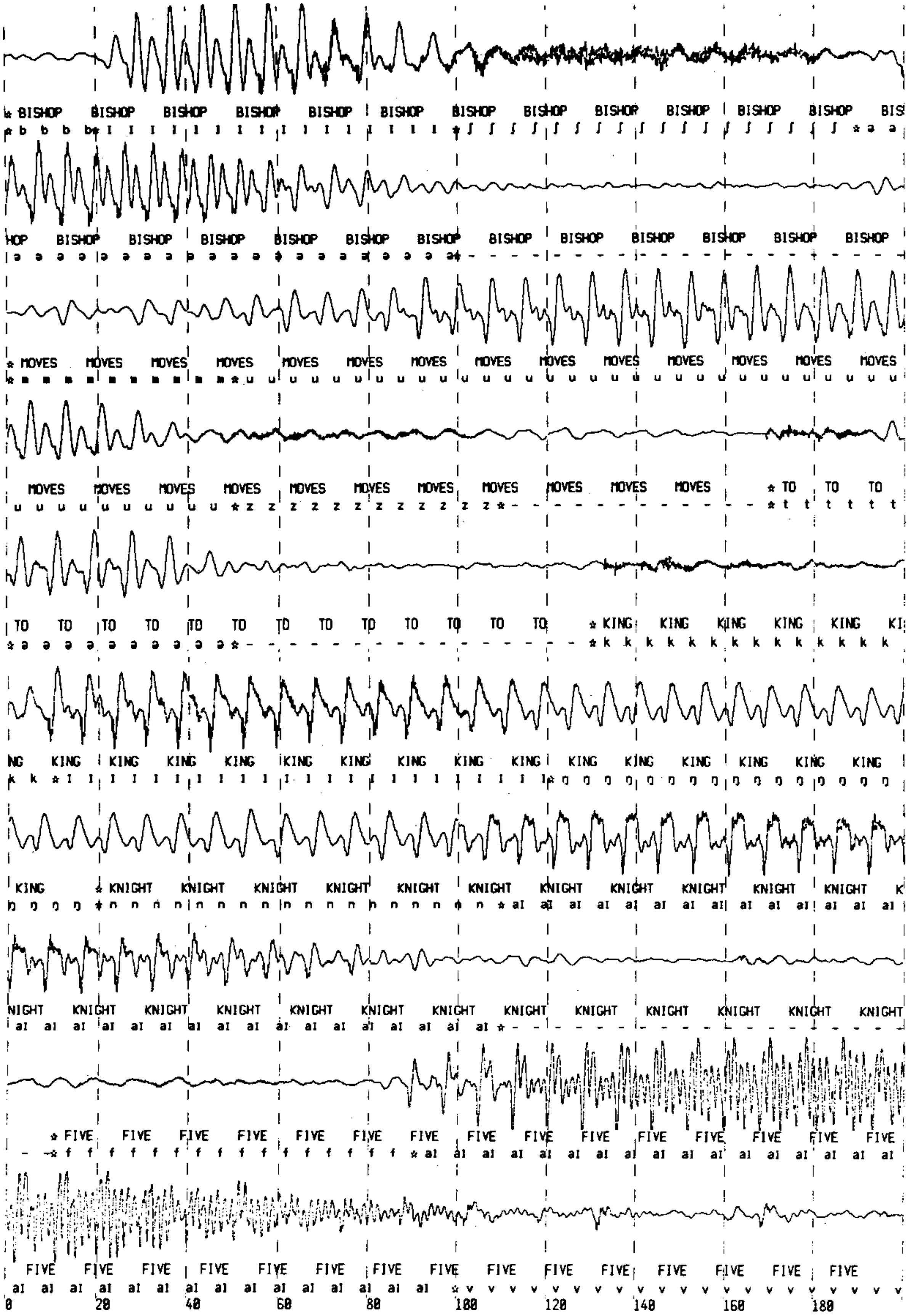


Figure 3: Waveform of the utterance with the "actual" word and phoneme boundaries.

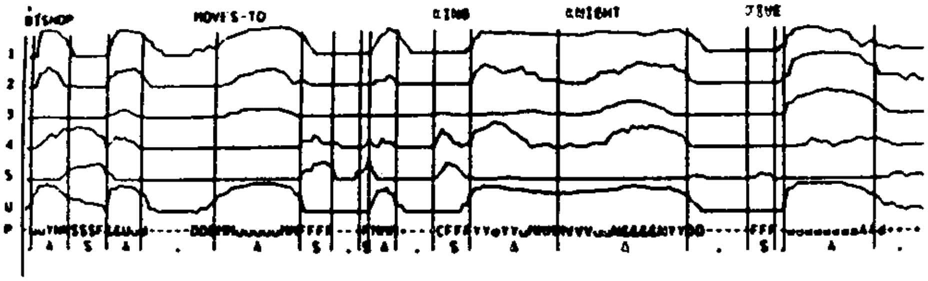


Figure 4. Parametric representation of the utterance showing the results of feature extraction and segmentation

using a modified nearest-neighbor classification technique. The purpose of this operation is to assign a (single character) label to each centisecond of speech using a compact pseudo-phonetic notation representing the actual local characteristics of the speech signal. The line of text labeled P in Figure 4 gives the classification for every 10-millisecond unit.

The classification of labels for each centisecond obtained by this match procedure (row P in Figure 4) is then used to specify a list of features, such as voicing and frication, which are then used in the segmentation of the utterance, shown in Figure 4. The boundaries of segments are indicated by vertical lines through the parameters, and the letter at the center of each segment (following the row P in Figure 4) indicates the type of segment that is present. The "A" indicates a sonorant segment, i.e., all the voiced unfricated segments; the "S" indicates a fricated segment, and the period (".") indicates a silence segment. The first use of an acoustic-phonetic source of knowledge can be seen in the handling of the "king knight" word boundary problem mentioned earlier. A long sonorant segment is subdivided into two segments to indicate the presence of two different syllables. The syllable juncture is determined in this case by the presence of a significant local minimum in an overall intensity plot (line labeled U on Figure 4).

The Recognition Process

The Hearsay system, at present, has three cooperating independent processes which help in the decoding of the utterances. These represent acoustic, syntactic, and semantic sources of knowledge:

1. The acoustic-phonetic domain, which we refer to as just **acoustics**, deals with the sounds of the language and how they relate to the speech signal produced by the speaker. This domain of knowledge has traditionally been the only one used in most previous attempts at speech recognition.
2. The **syntax** domain deals with the ordering of words in the utterance according to the grammar of the input language.
3. The **semantic** domain considers the meaning of the utterances of the language, in the context of the task.

The actual number and nature of these sources of knowledge is somewhat arbitrary. What is important to notice is that there can be several cooperating independent processes.

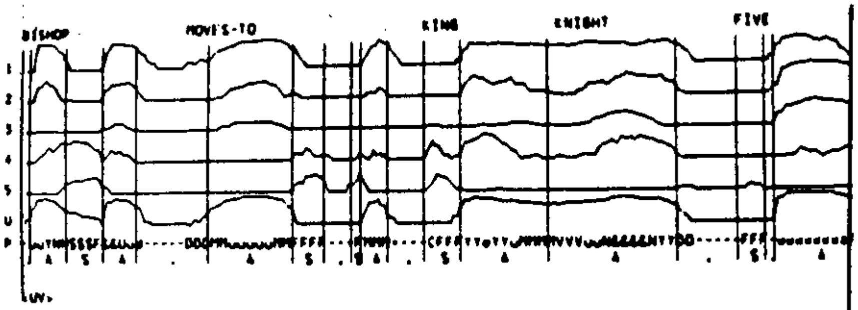
These processes cooperate by means of a hypothesize-and-test paradigm. This paradigm consists of one or more sources of knowledge looking at the unrecognized portion of the utterance and generating an ordered list of hypotheses. These hypotheses may then be verified by one or more of the sources of knowledge; the verification may accept, reject, or re-order the hypotheses. The same source of knowledge may be used in different ways both to generate hypotheses and to verify (or reject) hypotheses.

We will illustrate this recognition process by following through various stages of recognition for the utterance given in Figures 3 and 4. Figures 5 through 12 illustrate several of these stages of the recognition. In each figure, we have four kinds of information in addition to what was shown in Figure 4: the current sentence hypothesis (immediately below the P and segmentation rows), the processes acting on the current sentence hypothesis and their effect (e.g., SYN HYPOTHESIZED..., ACO REJECTED...), the acceptable option words with their ratings and word boundaries (e.g., ↑...↑ 500 Rook's), and the four best sentence hypotheses which result by adding the possible option words to the current best sentence hypothesis. When there are more than eight option words, only the best eight are shown. When there are more than four sentence hypotheses, only the best four are shown. The symbol <UV> within the current sentence hypothesis gives the location of the set of new words being hypothesized and verified. The "↑...↑" arrows indicate the possible beginning and ending for each option word.

Figure 5 shows the first cycle of the recognition process. At this point none of the words in the sentence have been recognized and the processing begins left to right. The Syntax module chooses to hypothesize and generates 13 possible words, implying that the sentence can begin with "rook's", "rook", "queen's", etc. Of these, the Acoustics module absolutely rejects the word "bishop's" as being severely inconsistent with the acoustic-phonetic evidence. The Semantics module rejects "castle" and "castles" as being illegal in this board position. The remaining 10 words are rated by each of the sources of knowledge. The composite rating and the word beginning and ending markers for the eight best words are shown in Figure 5. The words "rook", "rook's", "queen's" and "queen" all get a rating of 500. "Bishop", the correct word, gets a rating of 513. These words are then used to form the beginning sentence hypotheses, the top four of which are shown at the bottom of Figure 5.

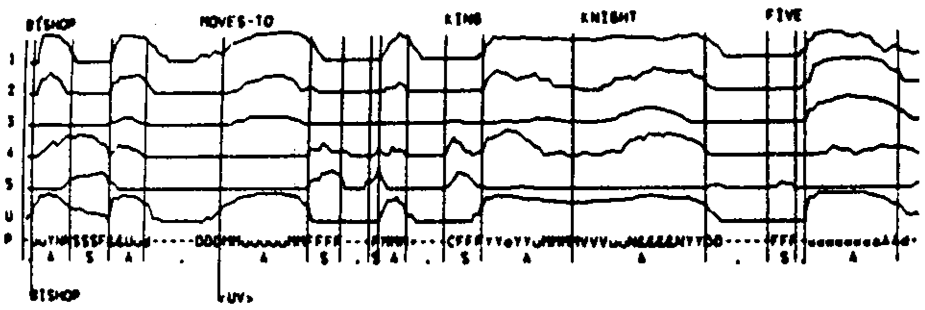
Figure 6 shows the second cycle of the recognition process. The top sentence hypothesis is "bishop ---". An attempt is being made to recognize the word following "bishop". Again Syntax generates the hypotheses. Given that "bishop" is the preceding word, the syntactic source of knowledge proposes only 7 options out of the possible 31 words in the lexicon -- a reduction in search space by a factor of 4. Of these possible 7 words, Acoustics rejects "captures" and Semantics rejects none. The remaining six words are rated by each of the sources of knowledge and a composite rating along with word boundaries is shown in Figure 6 for each of the acceptable words ("to" has a rating of 443, etc.). The correct word, "moves-to", happens to get the highest rating of 525. The new top sentence hypothesis is "bishop moves-to ---", with a composite sentence rating of 547.

Figure 7 shows the third cycle of the recognition process.



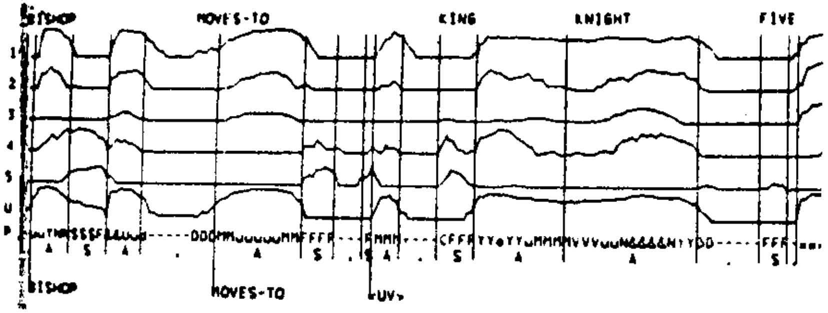
```
  ↑........↑ 500 ROOK'S
  ↑........↑ 500 ROOK
 ↑.........↑ 500 QUEEN'S
 ↑....↑ 500 QUEEN
 ↑....↑ 423 PAWN
 ↑....↑ 463 KING
  ↑ 513 BISHOP.........↑
 ↑.........↑ 463 KING'S
SYN HYPOTHESIZED    13 OW'S: ROOK'S,ROOK,QUEEN'S,QUEEN,PAWN,KNIGHT'S,KNIGHT,KING'S,KING,CASTLES
ACO REJECTED         1 OW'S: BISHOP'S
SEM REJECTED         2 OW'S: CASTLES,CASTLE

THE TOP 4 SENTENCE HYPOTHESES ARE:
( 533)  BISHOP <FILLER>
( 520)  ROOK'S <FILLER>
( 520)  ROOK <FILLER>
( 520)  QUEEN'S <FILLER>
```

Figure 5: First stage of the recognition process.



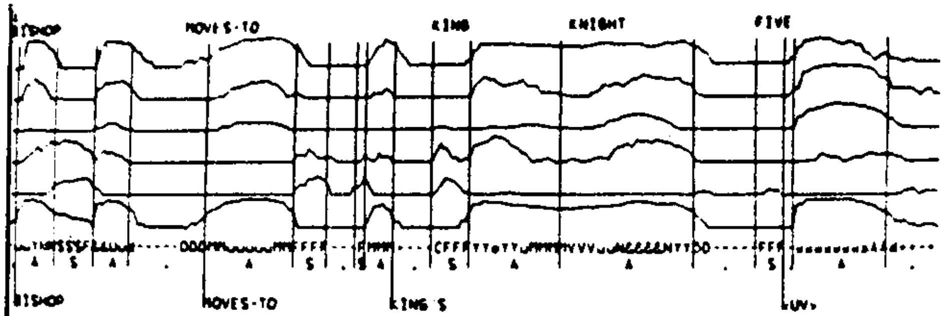
```
        ↑ 443 TO.........↑
        ↑ 345 TAKES..........↑
        ↑ 455 PAWN.......↑
                 ↑ 435 ON.↑
                 ↑ 525 MOVES-TO...↑
                 ↑ 463 GOES-TO...↑
                 ↓   0 CAPTURES ↓
SYN HYPOTHESIZED    7 OW'S: TO,TAKES,PAWN,ON,MOVES-TO,GOES-TO,CAPTURES
ACO REJECTED        1 OW'S: CAPTURES

THE TOP 4 SENTENCE HYPOTHESES ARE:
( 547)  BISHOP MOVES-TO <FILLER>
( 533)  BISHOP <FILLER>
( 526)  BISHOP GOES-TO <FILLER>
( 523)  BISHOP PAWN <FILLER>
```

Figure 6: Second stage of the recognition process.



```
                                ↑......↑ 450 ROOK
                                  ↑ 503 QUEEN'S...............................↑
                                  ↑ 503 QUEEN......↑
                                ↑......↑ 540 KNIGHT
                                  ↑ 513 KING'S...............................↑
                                  ↑ 513 KING.......↑
                                ↑ 423 BISHOP.............................↑
SYN HYPOTHESIZED    7 OW'S: ROOK,QUEEN'S,QUEEN,KNIGHT,KING'S,KING,BISHOP

THE TOP 4 SENTENCE HYPOTHESES ARE:
( 552)  BISHOP MOVES-TO KING'S <FILLER>
( 552)  BISHOP MOVES-TO KING <FILLER>
( 547)  BISHOP MOVES-TO <FILLER>
( 547)  BISHOP MOVES-TO QUEEN'S <FILLER>
```

Figure 7: Third stage of the recognition process.



```
                                                              ↑ 360 ROOK.....↑
                                                              ↑ 370 KNIGHT...↑
                                                             ↓   0 BISHOP ↓
SYN HYPOTHESIZED    3 OW'S: ROOK,KNIGHT,BISHOP
ACO REJECTED        1 OW'S: BISHOP

THE TOP 4 SENTENCE HYPOTHESES ARE:
( 552)  BISHOP MOVES-TO KING'S <FILLER>
( 552)  BISHOP MOVES-TO KING <FILLER>
( 547)  BISHOP MOVES-TO <FILLER>
( 547)  BISHOP MOVES-TO QUEEN'S <FILLFR>
```

Figure 8: Fourth stage of the recognition process.

Given the top sentence hypothesis "bishop moves-to ---", the Syntax module hypothesizes 7 option words. None of these were rejected by Acoustics or Semantics. "King" and "king's" both get the highest score of 513. The first error in the recognition process occurs at this point. As new sentence hypotheses are created based on the ratings of individual words, both "bishop moves-to king's ---" and "bishop moves-to king ---" have the same rating, with the former appearing at the top of the list. At this point it is instructive to see why the error was made in the first place. The phonemic description of "king's" causes a search for a stop followed by a vowel-like segment followed by a stop and fricative. This sequence of segments occurs in "king knight five" as can be seen from Figure 4 (improvements currently being made to the system will result in "king's" getting a much lower score). The important thing to observe is how the system recovers from errors of this type.

Figure 8 shows the system attempting to associate a meaningful word to the unverified part of the utterance, i.e., the /alv/ part of the word "five" in the original utterance. Syntax proposes 3 possible option words (out of a possible 31, giving a factor of 10 reduction). One is rejected and the other two get very low ratings. The corresponding sentence hypotheses also get low composite ratings and end up at the bottom of the stack (not visible in Figure 8).

Now we see an interesting feature of the system. In the preceding cycle (Figure 8) Syntax generated the hypotheses. It is possible that that source of knowledge is incomplete and did not generate the correct word as a possible hypothesis. Therefore, in this cycle (Figure 9), the Semantic module is given a chance to hypothesize. It hypothesizes 9 option words (a reduction of search by a factor of 3) all of which are rejected by Syntax and Acoustics. When both attempts to make a meaningful completion of the utterance fail, this particular sentence hypothesis, "bishop moves to king's--", is removed from the candidate list.

Now the top sentence hypothesis is "bishop moves-to king--" (Figure 10). Syntax hypothesizes 11 option words. Acoustics rejects six of them and Semantics rejects two. Of the remaining words, the correct word, "knight", gets the second best rating after "bishop". Again there is an errorful path, because the top sentence hypothesis now happens to be "bishop moves-to king bishop ---". This sentence hypothesis is rejected immediately in the next cycle because there is no more utterance to be recognized and "bishop moves-to king bishop" is not a legal move. Note that the correct sentence hypothesis is not at the top of the stack. Its rating of 550 is not as good as "bishop moves-to king ---" (see Figure 10).

The processing in the next cycle is illustrated in Figure 11. Note that in Figure 10, this same sentence hypothesis was used when the Syntax module hypothesized. Now Semantics is given an option to hypothesize and proposes 3 words. All of these are rejected by Syntax and Acoustics.

Finally, the correct partial sentence hypothesis, "bishop moves-to king knight ---", gets to the top (Figure 12). Syntax hypothesizes 17 option words. Of these Semantics rejects 16 as being incorrect, leaving only "five" as a possibility. This results in the correct complete sentence hypothesis of "bishop moves-to king knight five". But the composite rating for this sentence is only 545 and there are other partial sentence hypotheses with higher ratings. At this point, the system cycles eight more times before rejecting all of them and accepting the correct sentence hypothesis.

Figure 13 shows the accuracy of the system in recognizing some typical sentences. An attempt was made to estimate the effect of syntax and semantics. Using Syntax only, the average number of words analyzed was reduced to 9.4 out of the possible 31 words in the lexicon -- a reduction in search space by a factor of 3. Using Semantics only, the reduction of search space was about the same. Using both knowledge sources results in a reduction in the search space by a factor of 5.

```
    SPOKEN
        /RECOGNIZED (if not compeletely correct)

    pawn to queen four

    pawn to queen bishop four

    pawn to king four

    knight to queen bishop three

    bishop takes pawn

    queen takes queen on queen four
        (gave up after 48 seconds of computation)

    bishop to queen knight three

    bishop to king three
        bishop to king five

    castles queen side
        castles queen's side (understood correctly)

    pawn to bishop three

    pawn takes knight

    knight to queen five

    knight takes knight

    bishop to king rook six

    rook to queen three

    knight to rook three

    rook on rook one to queen one

    rook on queen one takes rook on queen three
        rook on queen one to king rook one check

    knight's pawn takes bishop

19 utterances tried:
15 recognized correctly, 16 understood correctly.  1 conceded.
Mean computation time per utterance:  10.1 sec. (PDP10 - KI10)
```

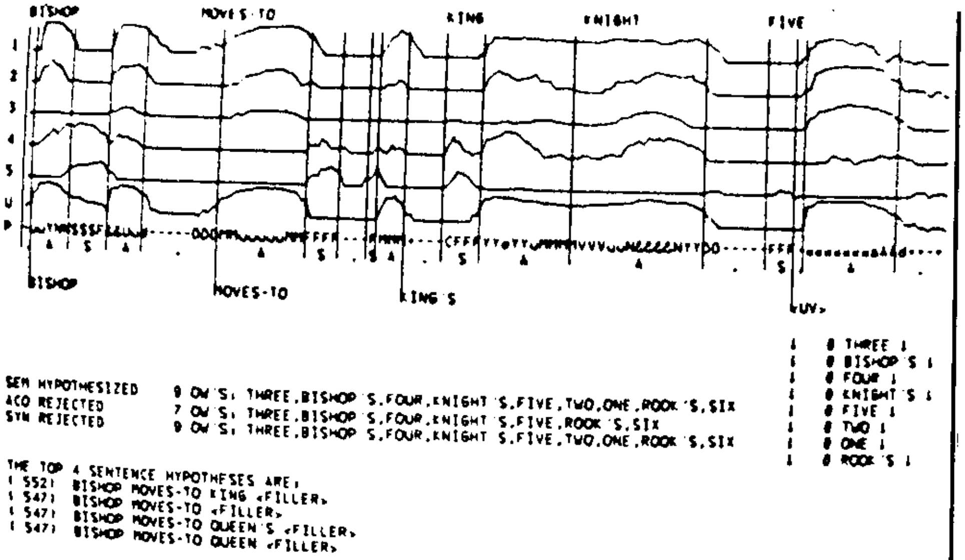
Figure 13: Examples of results for one run.

SOURCES OF KNOWLEDGE:
Their Representation and Use in the Hearsay System

Several sources of knowledge are used in the Hearsay system at present: speaker- and environment-dependent knowledge, acoustic-phonetic rules, vocabulary restrictions, and syntactic and semantic knowledge. The knowledge used at present represents only a small part of all the available knowledge. We expect to be adding to the knowledge base of the system for many years to come. The difficulties in representation and use of knowledge within the system are manifold. Even when rules exist which express pertinent knowledge, their applicability seems very limited and the effort involved to make effective use of them within the system is very large. Rules that exist are scattered in the literature. Many have not been written down and exist only in the heads of some scientists, and many are yet to be discovered. In this section, we will restrict ourselves to the discussion of the knowledge that is incorporated into the present Hearsay system.

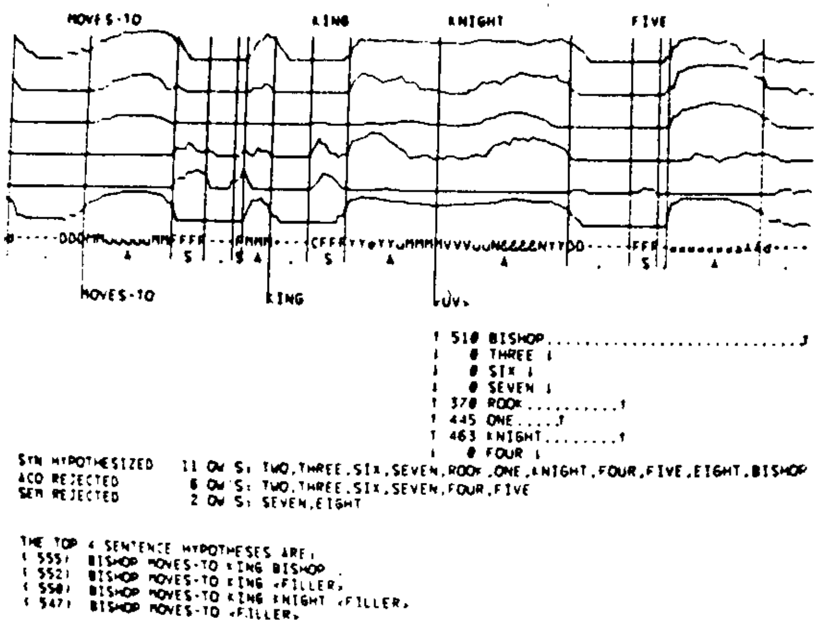
Speaker and Environment Dependent Knowledge

The characteristics of speech vary, depending on the speaker, age, sex, and physical condition. In addition, the



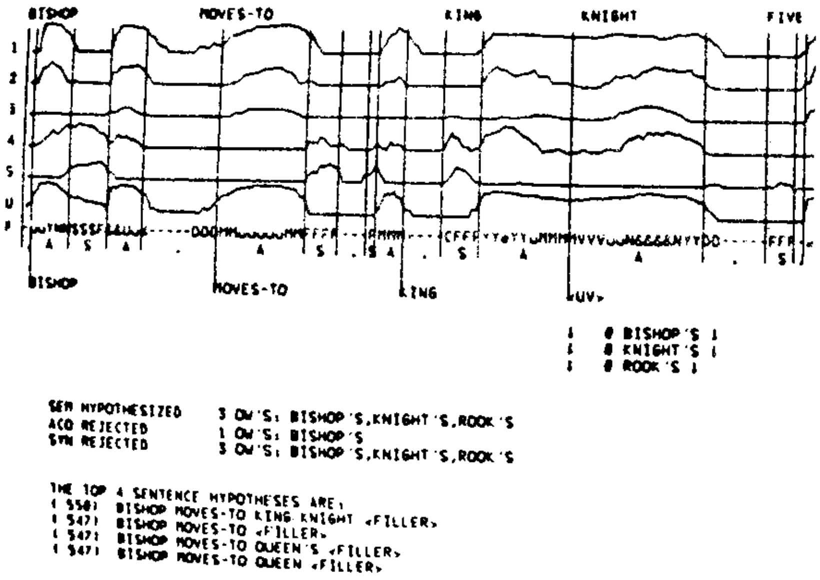
```
                                                              ↓   Ø THREE ↓
                                                              ↓   Ø BISHOP'S ↓
                                                              ↓   Ø FOUR ↓
SEM HYPOTHESIZED    9 OW'S: THREE,BISHOP'S,FOUR,KNIGHT'S,FIVE,TWO,ONE,ROOK'S,SIX   ↓   Ø KNIGHT'S ↓
ACO REJECTED        7 OW'S: THREE,BISHOP'S,FOUR,KNIGHT'S,FIVE,ROOK'S,SIX           ↓   Ø FIVE ↓
SYN REJECTED        9 OW'S: THREE,BISHOP'S,FOUR,KNIGHT'S,FIVE,TWO,ONE,ROOK'S,SIX   ↓   Ø TWO ↓
                                                              ↓   Ø ONE ↓
                                                              ↓   Ø ROOK'S ↓

THE TOP 4 SENTENCE HYPOTHESES ARE:
( 552)  BISHOP MOVES-TO KING <FILLER>
( 547)  BISHOP MOVES-TO <FILLER>
( 547)  BISHOP MOVES-TO QUEEN'S <FILLER>
( 547)  BISHOP MOVES-TO QUEEN <FILLER>
```

Figure 9: Fifth stage of the recognition process.



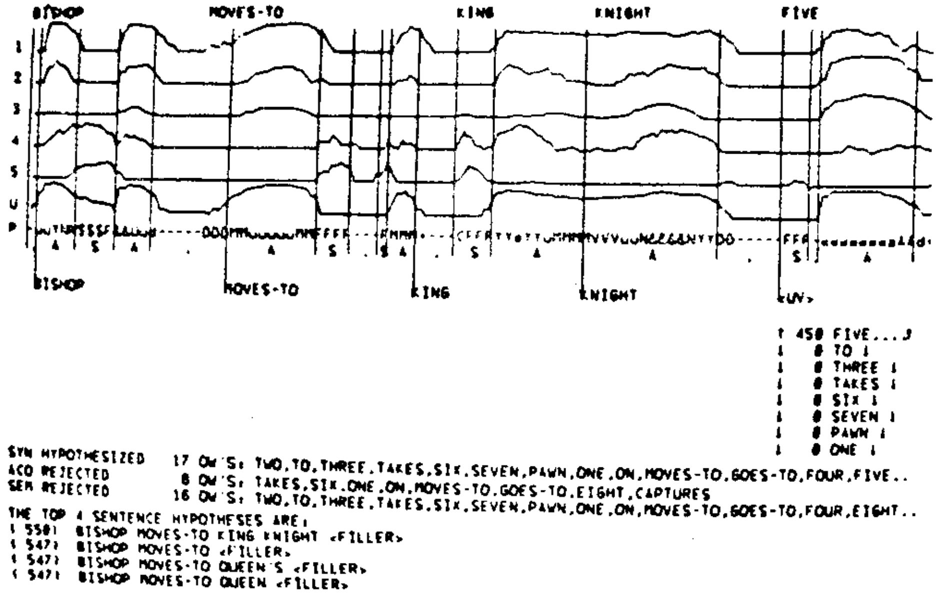
```
                                          ↑ 51Ø BISHOP..........................J
                                          ↓   Ø THREE ↓
                                          ↓   Ø SIX ↓
                                          ↓   Ø SEVEN ↓
                                          ↑ 37Ø ROOK..........↑
                                          ↑ 445 ONE.....↑
                                          ↑ 463 KNIGHT........↑
                                          ↓   Ø FOUR ↓
SYN HYPOTHESIZED    11 OW'S: TWO,THREE,SIX,SEVEN,ROOK,ONE,KNIGHT,FOUR,FIVE,EIGHT,BISHOP
ACO REJECTED         6 OW'S: TWO,THREE,SIX,SEVEN,FOUR,FIVE
SEM REJECTED         2 OW'S: SEVEN,EIGHT

THE TOP 4 SENTENCE HYPOTHESES ARE:
( 555)  BISHOP MOVES-TO KING BISHOP .
( 552)  BISHOP MOVES-TO KING <FILLER>
( 550)  BISHOP MOVES-TO KING KNIGHT <FILLER>
( 547)  BISHOP MOVES-TO <FILLER>
```

Figure 10: Sixth stage of the recognition process.



```
                                                   ↓   Ø BISHOP'S ↓
                                                   ↓   Ø KNIGHT'S ↓
                                                   ↓   Ø ROOK'S ↓

SEM HYPOTHESIZED     3 OW'S: BISHOP'S,KNIGHT'S,ROOK'S
ACO REJECTED         1 OW'S: BISHOP'S
SYN REJECTED         3 OW'S: BISHOP'S,KNIGHT'S,ROOK'S

THE TOP 4 SENTENCE HYPOTHESES ARE:
( 550)  BISHOP MOVES-TO KING KNIGHT <FILLER>
( 547)  BISHOP MOVES-TO <FILLER>
( 547)  BISHOP MOVES-TO QUEEN'S <FILLER>
( 547)  BISHOP MOVES-TO QUEEN <FILLER>
```

Figure 11: Seventh stage of the recognition process.



```
                                                              ↑ 45Ø FIVE...J
                                                              ↓   Ø TO ↓
                                                              ↓   Ø THREE ↓
                                                              ↓   Ø TAKES ↓
                                                              ↓   Ø SIX ↓
                                                              ↓   Ø SEVEN ↓
                                                              ↓   Ø PAWN ↓
                                                              ↓   Ø ONE ↓
SYN HYPOTHESIZED    17 OW'S: TWO,TO,THREE,TAKES,SIX,SEVEN,PAWN,ONE,ON,MOVES-TO,GOES-TO,FOUR,FIVE..
ACO REJECTED         8 OW'S: TAKES,SIX,ONE,ON,MOVES-TO,GOES-TO,EIGHT,CAPTURES
SEM REJECTED        16 OW'S: TWO,TO,THREE,TAKES,SIX,SEVEN,PAWN,ONE,ON,MOVES-TO,GOES-TO,FOUR,EIGHT..
THE TOP 4 SENTENCE HYPOTHESES ARE:
( 550)  BISHOP MOVES-TO KING KNIGHT <FILLER>
( 547)  BISHOP MOVES-TO <FILLER>
( 547)  BISHOP MOVES-TO QUEEN'S <FILLER>
( 547)  BISHOP MOVES-TO QUEEN <FILLER>
```

Figure 12: Eighth stage of the recognition process.

characteristics of the environment (such as background noise) and the characteristics of the transducer (such as the frequency response characteristics of the microphone) also cause variability in speech characteristics.

In the Hearsay system an attempt is made to correct for these variables through the use of a PP table. This table contains a standard set of parameters for various phones uttered by the speaker in a neutral phonetic context. This set of parameters also accounts for the characteristics of the room noise and the characteristics of the microphone in that the neutral phones were uttered in the very same environment. A complete list of the clusters used and the details of the speaker and environment normalization are given in Erman (1973).

Acoustic-Phonetic Knowledge

This knowledge is used in several places within the system to perform different functions. Knowledge related to syllabic structure is used in the segmentation. For each segment, knowledge related to voicing, frication, and syllable junction (a local minimum of energy) is used to assign labels to each segment. An example of segmentation and labeling obtained by this type of knowledge is given in Figure 4.

The acoustic-phonetic knowledge is used in the recognition process in two ways: to generate hypotheses about possible words that may be present in the incoming utterance; and to reject, accept, or re-order the hypotheses generated by other sources of knowledge.

The hypothesization is based on the fact that certain sounds within an utterance, e.g., stressed vowels, sibilants, and unvoiced stops, can usually be uniquely recognized. These features of the incoming utterance can then be used as an acoustic-phonetic filter on the lexicon to hypothesize only those words that are appropriate in this acoustic context.

When the acoustic-phonetic knowledge is used to verify hypotheses, it performs a more thorough analysis. Given a hypothesized word, its phonetic description is located in the lexicon. This description is used to guide the search for the word by means of phoneme procedures. That is, the expected characteristics of a given phoneme in various contexts are represented as a procedure; this procedure is activated to see if the expected features are present, and to provide a confidence rating based on the acoustic evidence. There are several increasingly more sophisticated verification procedures that can be used to verify proposed hypotheses. These sophisticated procedures are only invoked if word ambiguity exists at the preceding level.

Syntactic and Semantic Knowledge

Conventional parsing techniques are not very useful to direct the search within a speech understanding system. The recognizer must be capable of processing errorful strings containing spurious and repeated words. This implies that the parser must be capable of starting in the middle of the utterance where a word might be recognized uniquely and parse both forwards and backwards. The goal of parsing is not so much to generate a parse tree, but to predict what terminal symbol might appear to the left or to the right of a given context.

The predictive parsing for hypothesization is achieved in the Hearsay system by the use of anti-productions. Anti-productions act as a concordance for the grammar giving all the contexts for every symbol appearing in the grammar; they are generated from a BNF description of the language to be recognized. The anti-productions are used to predict words that are likely to occur following or preceding a word using only a limited context. Examples of anti-productions and their use are given by Neely (1973). The role of the syntactic verifier is to accept or discard hypotheses by using syntactic consistency checks based on the partial parse of the utterance. While the knowledge used for hypothesization and verification are the same, the representation and the mechanisms used in the hypothesization and verification are different. Figures 5 and 6 give examples of constraints provided by the syntactic knowledge during hypothesization. Figure 9 illustrates its use in verification.

The semantic source of knowledge for Voice-Chess is based on the semantics of the task, the current board position, and the likelihood of the move. This knowledge is used to predict likely legal moves; these moves are then used in conjunction with the partially-recognized utterance to predict a word that might appear in the utterance. The same knowledge is also used to verify hypotheses generated by other sources of knowledge. Figure 9 illustrates the use of semantic knowledge to generate hypotheses. In the context of "bishop moves-to king", Semantics hypothesizes nine possible words. It hypothesizes all the words that might appear in the utterance in positions allowed by the semantic knowledge, given the partial recognition. Figure 12 shows the use of Semantics in the verification. Syntax hypothesizes 17 possible words. The semantic knowledge, given the partially recognized utterance "bishop moves to king knight", indicates that only "five" is legal in that context by rejecting all others.

SUMMARY

This paper reports on research in progress on the Hearsay speech understanding system. The system has been operational since June, 1972. At present we are attempting to improve the accuracy and performance of the system by adding to and improving the knowledge base. This is being done by an analysis of errors made by the system on seven sets of data from five male speakers in four different task domains. This process of modification and improvement is expected to continue for several years, using increasingly complex vocabularies, syntax, and task environments. The Hearsay system will be used primarily as a research tool to evaluate the contributions of various sources of knowledge, as well as serving as an information processing model of speech perception.

ACKNOWLEDGMENT

We wish particularly to acknowledge the efforts of Bruce Lowerre who has increased the acoustic-phonetic knowledge base in Hearsay, thereby greatly improving the system's performance.

BIBLIOGRAPHY

1. Erman, L.D. (1973, in preparation), An Environment and System for Machine Recognition of Connected Speech, Ph.D. Thesis, Comp. Sci. Dept., Stanford Univ., to appear as a Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pa.

2. Fant, G. (1964), Auditory Patterns of Speech", in W. Wathen-Dunn (ed), **Models for the Perception of Speech and Visual Form**, MIT Press.

3. Gillogly, J.J. (1972), The TECHNOLOGY Chess Program, **Artificial Intelligence**, 3, 145-163.

4. Halle, M., and K. Stevens (1962), "Speech Recognition: A Model and a Program for Research", **IRE Trans. Inform. Theory**, IT-8, 155-159.

5. Liberman, A.M., F.S. Cooper, K.S. Harris, and P.F. MacNeilage (1962), "A Motor Theory of Speech Perception", Proc. of Speech Comm. Seminar, 2, KTH, Stockholm.

6. Neely, R.B. (1973), On the Use of Syntax and Semantics in a Speech Understanding System, Ph.D. Thesis, Stanford Univ., to appear as a Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pa.

7. Newell, A., J. Barnett, J. Forgie, C. Green, D. Klatt, J.C.R. Licklider, J. Munson, R. Reddy, and W. Woods (1971), Final Report of a Study Group on Speech Understanding Systems, North Holland (1973).

8. Reddy, D.R., L.D. Erman, and R.B. Neely (1970), The C-MU Speech Recognition Project, Proc. IEEE System Sciences and Cybernetics Conf., Pittsburgh, Pa.

9. Reddy, D.R. (1971), Speech Recognition: Prospects for the Seventies, Proc. IFIP 1971, Ljubljana, Yugoslavia, Invited paper section, pp. I-5 to I-13.

10. Reddy, D.R., L.D. Erman, and R.B. Neely, et al. (1972), Working Papers in Speech Recognition, Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pa.

11. Reddy, D.R., L.D. Erman, and R.B. Neely (1972a), A Model and A System for Machine Recognition of Speech, (to be published in **IEEE Trans. on Audio and Electro-acoustics**, 1973).

SYSTEM ORGANIZATIONS FOR SPEECH UNDERSTANDING:
Implications of Network and Multiprocessor Computer Architectures for AI

by

L.D. Erman, R.D. Fennell, V.R. Lesser, and D.R. Reddy

Computer Science Department*
Carnegie-Mellon University
Pittsburgh, Pa. 15213

ABSTRACT

This paper considers various factors affecting system organization for speech understanding research. The structure of the Hearsay system based on a set of cooperating, independent processes using the hypothesize-and-test paradigm is presented. Design considerations for the effective use of multiprocessor and network architectures in speech understanding systems are presented: control of processes, interprocess communication and data sharing, resource allocation, and debugging are discussed.

Keywords: **speech recognition, speech understanding, system organization, networks, multiprocessors, parallel processing, real-time systems, hardware for AI, software for AI.**

INTRODUCTION

System organizations for speech understanding systems must address many problems: effective use of multiple sources of knowledge, anticipation and goal-direction in the analysis of the incoming utterance, real-time response, continuous monitoring of input device(s), errorful nature of the recognition process, exponential increase of processing requirements with the increase of desired accuracy, and so on. A particular model of speech perception (Reddy et al., 1973) which attempts to solve the above problems involves the use of cooperating independent processes using a hypothesize-and-test paradigm. This paper examines the effect of the problem constraints and the model on system organizations, presents the structure of a system currently operational on a PDP-10 computer, and discusses the implications of multiprocessor and network architectures.

Unlike many other problems in artificial intelligence, speech understanding systems are characterized by the availability of diverse sources of knowledge, e.g., acoustic-phonetic rules, phonological rules, articulatory models of speech production, vocabulary and syntactic constraints, semantics of the task domain, user models, and so on. A major problem, then, is to develop paradigms which can make use of all the available sources of knowledge in the problem solution. At the same time, absence of one or more sources of knowledge should not cripple the system. Suppose each source of knowledge is represented within the system as a process. In order to remove or add sources of knowledge, each process must be independent, i.e., it must not require the presence of other processes in the system. But at the same time each process must cooperate with the other processes, i.e., it must be able to effectively use the information gathered by them about the incoming utterance. Thus, a major design step is to establish what information is to be shared among processes and how this information is to be communicated so as to maintain the independence of individual processes while still allowing for necessary process cooperation.

Knowledge available in the acoustic signal represents only one part of the total knowledge that is brought to bear in understanding a conversation. A good example of this is when one is interrupted by an appropriate response from the listener to a question that is as yet incomplete. In general, a human listener can tolerate a great deal of sloppiness and variability in speech because his knowledge base permits him to eliminate most of the possibilities even as he hears the first few words of the utterance (if not before!). We feel that this notion of anticipation, prediction, and hypothesis generation is essential for machine perception systems as well. In general, we expect every source of knowledge to be able to generate hypotheses in a given context, or verify hypotheses generated by others using different representations of knowledge, if necessary. The implication is that knowledge processes be organized within the system so as to reduce the problem of recognition and understanding to one of prediction and verification.

In tasks such as chess and theorem-proving, the human has sufficient trouble himself so as to make reasonably crude computer programs of interest. But, because humans seem to perform effortlessly (and with only modest error) in speech (and visual) perception tasks, similar performance is expected from machines, i.e., one expects an immediate response and will not tolerate any errors. To equal human performance, a speech understanding system must be able to understand trivial

* This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under contract no. F44620-70-C-0107 and monitored by the Air Force Office of Scientific Research.

questions as soon as they are uttered. This implies that various processes within the system should be allowed to operate as soon as there is sufficient incoming data, without waiting for the completion of the whole utterance. If the processes within the system are independent and unaware of the existence of each other, then the system must provide facilities for activation, termination, and resource allocation for each of the processes. Further, if a process can be deactivated before it reaches a natural termination point, provision must be made to preserve the state of the process until it is reactivated. Also, it is necessary to provide interlocks on the data that are shared among many processes.

This has several implications for system organization. The system must monitor the input device continuously to determine whether speech is present; this requires non-trivial processing. If the system is unable to process the incoming data, automatic buffering must be provided. If the system is to run on a time-sharing system, provision must be made to ensure that no data is lost because the program is swapped out for a period of time. If the speech understanding system is to consist of a set of cooperating independent processes, it is further necessary that they be able to be interrupted at unpreprogrammed points -- if the microphone monitoring program is not activated in time to process the incoming utterance, it could lead to irrevocable loss of data. These considerations lead to two additional requirements that are not commonly available on existing time-sharing systems, viz., process-generated interrupts of other processes and user servicing of interrupts.

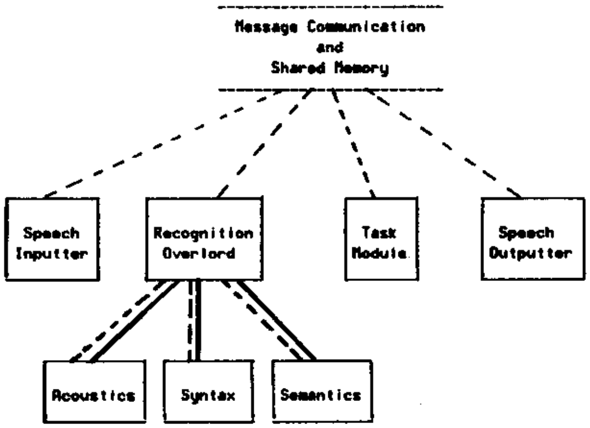
One of the characteristics of speech understanding systems is the presence of error at every level of analysis. To control such errors and permit recycling with improved definitions of the situation, one uses techniques such as feedforward, feedback, and probabalistic backtracking. If such facilities do not exist within the system, they have to be programmed explicitly.

Speech, by its nature, appears to be computer intensive. A substantially unrestricted system capable of reliably understanding connected speech of many speakers using a large vocabulary is likely to require systems of the order of a proposed AI machine (Bell, Freeman, et al., 1971a), i.e., processing power of 10 to 100 million instructions per second and memory of 100 to 1000 million bits.* To obtain such processing power, it appears necessary to consider multiprocessor architectures. Decomposition of speech processing systems to effectively use distributed processing power requires careful consideration even with primitive systems. Our model of cooperating independent processes, each representing a source of knowledge, leads to a natural decomposition of the algorithms for such machine architectures.

THE CURRENT Hearsay SYSTEM

In this section we briefly describe the Hearsay speech understanding system as it now exists at C-MU. (More detailed descriptions of the system are given in Reddy et al., 1973,1973a (this volume); Erman, 1973; and Neely, 1973.) We shall stress those aspects of its organization which are responsive to the constraints and model outlined above. This system represents a first attempt to solve those problems; thus, some of the constraints are only partially or poorly met, while others are satisfied in a more constricted way than necessary. We shall point out these limitations as they are described; later sections on closely-coupled and loosely-coupled processor network architectures describe possible corrections and improvements of the system.

The Hearsay system is implemented as a small number of parallel coroutines (see figure). Each coroutine (module) is realized as a separate job in the PDP-10 time-sharing system; thus the time-sharing monitor is the primary scheduler for the modules. In general, the modules may achieve a high degree of (pseudo-) parallel activity (through the use of shared memory and a flexible inter-process message system*), but, in practice, we limit the parallelism to a very modest amount. This limitation is imposed for two reasons: first, since the PDP-10 is a uniprocessor system, there is nothing to be gained (in the time domain) by increasing the parallelism; and, second, the greater the amount of parallelism, the more difficult it is to control and debug the programs within a time-sharing system that is not designed for cooperating processes (jobs).



Decomposition of processes in the current Hearsay system.

The model of recognition specifies that there be separate processes, each representing a different domain of knowledge. We have chosen three major domains of knowledge: acoustic-phonetics, syntax, and semantics:

1. The acoustic-phonetic domain, which we refer to as just **acoustics**, deals with the sounds of the language and how they relate to the speech signal produced by the speaker. This domain of knowledge has traditionally been the only one used in most previous attempts at speech recognition.
2. The **syntax** domain deals with the ordering of words in the utterance according to the grammar of the input language.
3. The **semantic** domain considers the meaning of the utterances of the language, in the context of the task that is specified for the speech understanding system.

These processes, according to the model, are to be independent and removable; therefore the functioning (and very existence) of each must not be necessary or crucial to the others. On the other hand, the model also requires that the processes cooperate and that the recognition should run efficiently and with

* Smaller and substantially cheaper systems can be built to perform useful but restricted speech understanding tasks.

* The facilities provided for inter-job control and communication are similar to those developed for the Stanford Hand-Eye system (Feldman and Sproull, 1971).

good error recovery; these dictates imply that there be a great deal of interaction among the processes. Thus we seem to have opposing requirements for the system. These opposing requirements led to the design of the following structure:

> Each process interfaces externally in a uniform way that is identical across processes; no process knows what or how many other recognition processes exist.
>
> A mediator, ROVER (Recognition OVERlord), handles the interface to each of the processes and thus serves as the linkage connecting the processes; the processes are called ROVER's "sons."

The interface is implemented as a global data structure which is maintained by ROVER. Each of ROVER's sons puts information into this data structure in a uniform way. Each may access information submitted by its brothers, but in a manner which leaves the source of that information anonymous. This mechanism is analogous to a bulletin board on which messages can be left by several people and for which there is a monitor who accepts the message and arranges them in appropriate places on the board for others to react.

This anonymous interface structure is appropriate only if the global data structure can be designed in such a way as to allow the processes to communicate meaningfully; i.e. there must be a common language which allows them to transmit the kind of information they need to help each other to work on the problem. We resolve this problem by using the word as the basic unit of discourse among the processes.

The basic element of the global data structure is the **word hypothesis** which represents an assertion that a particular word (of the input language lexicon) occurs in a specified position in the spoken input. A **sentence hypothesis** is an ordered linear sequence of word hypotheses; it represents an assertion that the words occur in the sentence in the order that the word hypotheses appear in the sentence hypothesis. In addition, the unique "word" FILLER may appear as a word hypothesis; this is a placeholder and represents the assertion that zero or more as yet unspecified words occur in this position in the spoken sentence. In general, there may be any number of sentence hypotheses existing at any one time.

The interactions among the source-of-knowledge processes are carried out using the hypothesize-and-test paradigm prescribed by the model. In general, any process may make a set of hypotheses about the utterance; all the processes (including the hypothesizer) may then verify (i.e. reject, accept, or re-order) these hypotheses. In particular, hypothesization occurs when a recognition process (Acoustics, Syntax, or Semantics) chooses a FILLER word from a sentence hypothesis and associates with it one or more **option words**, each of which it asserts is a candidate to replace all or part of the FILLER. Verification consists of each process examining the option words and rating them in the context of the rest of the sentence hypothesis.

Several restrictions have been placed on the implementation of this general scheme. First, at any time only one part of the shared, global data structure (i.e., one sentence hypothesis) is accessible to the processes for hypothesization and verification. Second, the processes go through the hypothesization and verification stages (and several other subsidiary stages) in a synchronized and non-interruptable manner. Finally, only one process is allowed to hypothesize at any one time. Again, these restrictions were imposed both because parallelism on a uniprocessor does not accomplish any throughput increase and because the available programming and operating systems make a more general implementation difficult to specify, debug, and instrument. These restrictions are mitigated somewhat by carefully adjusting the time grain of the processing so that each non-interruptable phase is not "excessively large."

Each sentence hypothesis has a confidence rating associated with it which is an estimate of how well it describes the spoken utterance. This rating is calculated by ROVER, based on information supplied by the recognition processes. Errors in processing become evident when the overall rating given to a sentence hypothesis begins to drop; at that point, attention is focused on some other sentence hypothesis with a higher rating. This switching of focus is the mechanism that provides the error recovery and backtracking that is necessary in any speech understanding system.

CLOSELY-COUPLED PROCESSOR SYSTEM ORGANIZATIONS

As discussed in the introduction, in order to do real-time speech understanding a substantial amount of computing power is required. Recent trends in technology indicate that this computing power can be economically obtained through a closely-coupled network of "simple" processors, where these processors can be interconnected to communicate in a variety of ways (e.g., directly with each other through a highly multiplexed switch connected to a large shared memory (Bell et al., 1971), or through a regular or irregular network of busses (Bell et al., 1973)). However, the major problem with this network approach to generating computing power is finding algorithms which have the appropriate control and data structures for exploiting the parallelism available in the network. The model for a speech understanding system as previously discussed, which is decomposed into a set of independent processes cooperating through a hypothesize-and-test paradigm, represents a natural structure for exploiting this network parallelism.

There exist three major areas for exploitation of parallelism in the structure of this speech understanding system: preprocessing, hypothesization and verification, and the processing specific to each source of knowledge. The **preprocessing task** involves the repetition of a sequence of simple transformations on the acoustic data, e.g., detection of the beginning and end of speech, amplitude normalization, a simple phoneme-like labeling, smoothing, etc. This sequence of transformations can be structured as a pipeline computation in which each transformation is a stage in the pipe. Thus, through this pipeline decomposition of the preprocessing task, a limited amount (i.e., 4) of parallel activity is generated.

The **hypothesize-and-test paradigm** for sequencing the activity of the different sources of knowledge can also be structured so as to exhibit parallelism, but the amount of parallelism is potentially much greater. This parallel activity is generated by the simultaneous processing of multiple sentence hypotheses and the simultaneous hypothesization and verification by all sources of knowledge. The simultaneous processing of multiple sentence hypotheses, rather than processing just the currently most likely candidate, can conceptually introduce unnecessary work. But in practice, because of the errorful nature of the processing, there may be a considerable amount of necessary backtracking to find the best matching sentence hypothesis. It is appropriate to quote a conjecture of Minsky and Papert (1969, Section 12.7.6) on this point:

> [While for the exact match problem] relatively small factors of redundancy in memory size yield very large increases in speed, . . . [for the best match problem] . . . for large data sets with long word lengths there are no practical alternatives to large searches that inspect large parts of the memory.

Thus, the parallel activity generated by simultaneous processing of more than one sentence hypothesis can result in a proportional speed-up of the recognition process.* Correspondingly, simultaneous hypothesization and verification by all sources of knowledge also results in a proportional speed-up of the recognition process because each source of knowledge is independent and is designed so that its knowledge contirbution is additive.

Finally, the **verification algorithm** of each source of knowledge can be decomposed into a set of parallel processes in two ways: The first kind of decomposition is based on the fact that verifications are performed on a set of option words rather than a single word at a time. Thus, for each source of knowledge there can be multiple instantiations of its verification process, each operating on a different option word. The second kind of decomposition involves the parallelizing of the verification algorithms themselves; thus, each instantiation of a verification process may itself be composed of a set of parallel processes. However, this set of instantiations may not be totally independent because the rating produced by the verification process may be dependent on the particular set of option words to be verified and also on the local data base which is common to all the instantiations. For example, the acoustic verification process is a hierarchical series of progressively more sophisticated tests. The first few levels of testing look only at the context of a single option word, while the more sophisticated tests compare one option word against another. Thus, only at the first few levels of tests can the acoustic verification algorithm be parallelized in a straightforward manner.

The parallelism generated by parallelizing the hypothesize-and-test control structure and the verification processes are multiplicative in their parallel activity (i.e. performing in parallel the updating of **n** sentence hypothesis where each hypothesis invokes **m** verification processes and each verification process operates on **o** option words leads to a potential parallelism of **n*m*o**). This parallelism, together with the pipeline parallelism of the preprocessing, leads to what appears to be a large amount of potential parallelism to be exploited by a closely-coupled network. However, it is still not clear just how much potential parallel activity exists over the entire recognition system; nor is it known how much of this potential will be dissipated because of software and hardware overhead.

In order to answer these questions, a parallel decomposition of the Hearsay speech understanding system is now being implemented on C.mmp, a closely-coupled network of PDP-11's which communicate through a large shared memory (Bell et al., 1971). The C.mmp hardware configuration can contain up to 16 PDP-11's; the highly multiplexed switch that connects processors to memory permits up to 16 simultaneous memory references if these references are not to the same memory module. Thus, if processors are referencing different memory modules, then each processor can run at full speed. In addition, C.mmp can be configured for a specific application (e.g., speech) by replacing a processor by a special purpose hardware device which directly accesses memory (e.g., a signal processor).

The HYDRA software operating system (Wulf, 1972), which is associated with C.mmp, provides an appropriate kernel set of facilities for implementing the parallel version of the speech system. These facilities permit control of real-time devices, convenient building of a tree of processes, message queues and shared data base communication among processes, user-defined scheduling strategies, arbitrary interruption of running processes, and dynamic creation of new processes. Building up from this base, a debugging system will be constructed which, in addition to the normal features, will permit the recording of all communication among processes, the tracing of all process activity, and the monitoring of global variables (including a recording of which processes have modified them). These additional capabilities are crucial for isolating errors and understanding the dynamic behavior patterns of the parallel system.

The major software problem to be investigated in this parallel implementation of the Hearsay system is how to efficiently map virtual parallelism (process activity) into actual parallelism (processor activity). This mapping problem in turn centers on three design issues, each of which relates to how processes interact:

1. the design of the interlock structure for a shared data base,
2. the choice of the smallest computational grain at which the system exhibits parallel activity, and
3. the techniques for scheduling a large number of closely-coupled processes.

The first design issue is important because in a closely-coupled process structure many processes may attempt to access a shared data base at the same time. In a uniprocessor system, the sequentialization of access to this shared data base does not significantly affect performance because there is only one process running at a time. In a multiprocessor system, however, if the interlock structure for a shared data base is not properly designed so as to permit as many non-interfering accesses as possible, then access to the shared data base becomes a significant bottleneck in the system's performance (McCredie, 1972).

The second issue relates to how closely-coupled processes can interact. If the grain of decomposition is such that the overhead involved in process communication is significant in relation to the amount of computation done by the process, then the added virtual parallelism achieved by a finer decomposition can decrease, rather than increase, the performance of the system. Thus, understanding the relationship between the grain of decomposition and the overhead of communication is an important design parameter.

The third issue relates to a phenomenon called the "control working set" (Lesser, 1972). This phenomenon predicts that the execution of a closely-coupled process structure on a multiprocessor may result in a significant amount of supervisory overhead caused by a large number of process context switches. The reason for this high number of process context switches is analogous to the reason for "thrashing" within a data working set (Denning, 1968). For example, in a uniprocessor system if two parallel processes closely interact with each other, then each time one process is waiting for a communication from the other it would have to be context switched so as to allow the other process to execute. If these two processes communicate often then there would be a large number of context switches. However, if there were two processors, each containing one of the processes, then there would be no process switching.

The implications of this phenomenon on constructing process structures are the following:

1. Processes should be formed into clusters where communication among cluster members is closely-coupled whereas communication among clusters is loosely-coupled. This process structuring paradigm has also been been suggested as a model for the operation of complex human and natural systems (Simon, 1962).
2. The size of a process cluster cannot be chosen independent of the particular hardware configuration that will be used to execute it. For example, a cluster size of 8 may be appropriate

* Simulation studies are currently being carried out on evaluating this speed-up factor. These studies are based on data generated from the current version of the Hearsay system.

for a hardware system containing 16 processors while being inappropriate for a system containing 6 processors.

3. The scheduler of a multiprocessor system should use a strategy that schedules process clusters rather than single processes. (This is analogous to the advantage of preloading the data working set rather than dynamically constructing the working set at each context swap.)
4. The use of process structures to implement inherently sequential, though complex, control structures (e.g., coroutines, etc.) may lead to inefficient scheduling of process structures on a multiprocessor system (i.e., the scheduling strategy should be able to easily differentiate those processes that can go on in parallel from those that are sequentialized).

NETWORK ORGANIZATIONS

The multiprocessor type organization described earlier implies a closely-coupled set of processes on a set of closely-coupled processors cooperating to accomplish the common goal of utterance recognition. The key idea in such a system is that both the processes and processors are closely-coupled -- that is, the cost of communication between processes or processors is relatively cheap with respect to the amount of computation to be done by any individual process. Indeed, in the multiprocess system described earlier, much interprocess communication and data sharing may be achieved by actually having shared physical address spaces. However, such a system usually also implies a certain homogeneity or physical proximity of the processors and memory.

Consider now the task of integrating the knowledge of many different research groups in various widespread geographical locations, each with its own computing facilities and each with its own areas of specialization. In an attempt to avoid unnecessary duplications of effort, one would desire a scheme whereby each group could develop pieces of a total recognition system (which pieces might represent new sources of knowledge, such as a new and improved vowel classification algorithm) using local computing resources (i.e., using an arbitrary machine configuration and program structure). Those pieces of the system would then be incorporated into a distributed "total recognition system" by appropriate (hopefully minimal) linkage and protocol conventions and their contributions to the entire system evaluated. The geographical constraints suggest the use of a computer network facility as a means by which one might assemble this total recognition system. We are currently undertaking the task of designing and implementing such a system for use on the ARPA network of computing facilities (Roberts and Wessler, 1970). The usefulness of such a network organization for a speech understanding system lies in its potential ability to combine and evaluate the various algorithms and sources of knowledge of a wide variety of research groups. In particular, the objective of the network organization is to create a research tool rather than to produce a highly efficient recognition system.

As an example, suppose a group wishes to add a new source of knowledge (a new vowel classification algorithm, for instance) to the network system. This knowledge-source is provided in the form of a process (or a set of processes) running on a local computer connected to the ARPA network. System integration is then achieved by adding linking instructions to the process (perhaps interactively) for notifying a centralized controlling process of the set of pre-conditions (e.g., conditions relating to the incoming speech wave or the current state of the recognition) that must be met in order to activate this process (Adams, 1968), as well as the required inputs and created outputs (and their formats). The central controller is then responsible for activating the new knowledge source at appropriate times, supplying the requested inputs, and updating a global data base to reflect the results of the activated process. Knowledge source processes may communicate with one another via a message service facility provided by the central controller. The marked increase of indirection with respect to communication and data sharing as compared with a closely-coupled multiprocessor approach is a result of the goal to serve a wide geographic region of users and to allow cooperation between essentially autonomous knowledge sources.

The problems that occur in this network concept are of a nature different from that of those occurring in the multiprocessor structure described previously. The many sources of knowledge are no longer necessarily closely-coupled. In fact, we might term such a network organization to be "loosely-coupled" in the sense that process communication and data base sharing must be achieved by some form of message switching scheme since the system is now operating on an indefinite number of (nonhomogeneous) computers. In particular, there is no longer the ability for all processes to share data and communicate by sharing physical address spaces. The problems of data base sharing and shipping now abound: one would like not to have multiple copies of a given data structure due to updating synchronization problems, but the message switching involved in maintaining and updating a single, centralized data structure may be overwhelmingly inefficient.

It is intended that, besides serving as a research tool for testing various recognition algorithms and combinations thereof, such a network organization will become an interesting experiment in its own right. There remains much investigation to be conducted regarding the tradeoffs involved in passing and sharing data through channels having low communication rates, as well as investigating the means of coordination of many autonomous knowledge sources. Points of interest for systems design also exist in creating the appropriate interfaces between any given group's knowledge source process and the central controlling process. Specification for data base requirements and formats (for both input and output) and specifications for determining the pre-conditions upon which a process should be activated must be easily specified for each new process to be added. In particular, the new process should not need to know the details of the global data structures it may need to access -- the linkage interface should take care of such details (Parnas, 1971,1971a).

Issues of user control over the entire system and the human interface in general are considered vital, demanding much investigation for any system organization which intends to run as a set of parallel cooperating (whether closely- or loosely-coupled) processes. The user must have the ultimate control over halting the entire recognition system or some subset of processes involved therein and interrogating (and perhaps altering) the instantaneous state of any given process. Protocols for debugging and controlling any knowledge source process should be provided via the interface linkage setup. Systems allowing the amount of user control that might be desired are not easily achievable given the current state of the art, primarily due to a general lack of experience in multiprocess environments (however, see Swinehart, 1973). Given a well-defined problem environment such as the speech understanding task, which lends itself readily to a multiple-process decomposition, investigation into the realms of multiprocess debugging and control might now be given more definite aims. Indeed, the problems involved in controlling a set of independent parallel processes that are cooperating to solve a single problem reach beyond the issues raised in the development of present multiprogramming systems (e.g., monitoring and controlling the interactions involving shared data structures and process intercommunications demand that new debugging systems and strategies be formulated).

SUMMARY

The main focus of this paper has been to illustrate the issues of system organization that arise when one attempts to build a general speech understanding system which can equal human performance. In practice, however, one can finesse a large number of these issues by working with pre-recorded data and relaxing other requirements, such as real-time response. However, unless the system is organized with the eventual goals firmly in mind, one is likely to end up with dead-end systems, necessitating a complete reformulation of the problem solution. The complexity of the hardware and software problems raised by real-time requirements explains why there are very few systems which can accept or attempt recognition of live connected speech.

Usually the term "parallel processing" is used as if it will resolve all of one's problems. The intent here is mainly to indicate that speech understanding systems naturally decompose into a set of cooperating, independent processes. Whether one uses a single processor (as we now do) or many processors (as we propose to do), the program structure and organization tends to be similar. The main question, then, is how much computational power is available on the system to attempt real-time recognition of connected speech. The multiprocessor and network organizations provide an opportunity to study and evaluate relative merits of various computer architectures in this context.

Finally, we believe that the issues of system organization raised here are relevant to a large class of current problems in AI, e.g., vision, robotics, chess, chemistry, etc., where performance is the main criterion for acceptability and where many sources of knowledge are available. In particular, the notions of hypothesize-and-test and cooperating independent processes seem equally applicable to these areas as well.

ACKNOWLEDGMENTS

We would like to thank our colleagues C.G. Bell, R. Neely, A. Newell, D. Parnas, and W. Wulf for many interesting discussions on this subject.

BIBLIOGRAPHY

Adams, D.A. (1968), "A Computation Model with Data Flow Sequencing", Tech. Rep. CS-117 (Ph.D. Thesis), Comp. Sci. Dept., Stanford Univ.

Bell, C.G., W. Broadley, W. Wulf, A. Newell, et al. (1971), "C.mmp: The CMU Multi-mini-processor Computer", Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ.

Bell, G., P. Freeman, et al. (1971a), "C.ai: A Computing Environment for AI Research", Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ.

Bell, C.G., R.C. Chen, S.H. Fuller, J. Grason, S. Rege, and and D.P. Siewiorek (1973), "The Architecture and Application of Computer Modules: A Set of Components for Digital Systems Design", COMPCON 73, San Francisco, Ca.

Denning, P.J. (1968), "The Working Set Model for Program Behavior", Comm. ACM, 11, 5, 323-333.

Erman, L.D. (1973, in preparation), "An Environment and System for Machine Recognition of Connected Speech", (Ph.D. Thesis), Stanford Univ., to appear as Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ.

Feldman, J.A. and R.F. Sproull (1971), "System Support for the Stanford Hand-Eye System", Second Inter. Joint Conf. on Artificial Intelligence, 183-189.

Lesser, V.R. (1972), "Dynamic Control Structures and Their Use in Emulation", (Ph.D. Thesis), Tech. Rep. CS-309, Comp. Sci. Dept., Stanford Univ.

McCredie, J.W. (1972), "Analytic Models of Time-Shared Computing Systems: New Results, Validations and Uses", (Ph.D. Thesis), Comp. Sci. Dept., Carnegie-Mellon Univ., Chapter 5.

Minsky, M. and S. Papert (1969), **Perceptrons**, MIT Press, Cambridge, Mass.

Neely, R.B. (1973), "On the Use of Syntax and Semantics in a Speech Understanding System", (Ph.D. Thesis), Stanford Univ., to appear as a Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ.

Parnas, D.L. (1971), "Information Distribution Aspects of Design Methodology", Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ.

Parnas, D.L. (1971a), "On the Criteria to be Used in Decomposing Systems into Modules", Tech. Rep., Comp. Sci. Dept., Carnegie-Mellon Univ.

Roberts, L.G. and B.D. Wessler (1970), "Computer Network Development to Achieve Resource Sharing", Proc. SJCC, 36.

Reddy, D.R., L.D. Erman, and R.B. Neely (1973), "A Model and System for Machine Recognition of Speech", **IEEE Trans. Audio and Electro-acoustics**, in press.

Reddy, D.R., L.D. Erman, R.D. Fennell, and R.B. Neely (1973a), "The Hearsay Speech Understanding System: An Example of the Recognition Process", Third International Joint Conference on Artificial Intelligience (this volume).

Simon, H.A. (1962), "The Architecture of Complexity", Proc. Am. Phil. Soc. 106.

Swinehart, D.C. (1973, in preparation), "A Multiple-Process Approach to Interactive Programming Systems", (Ph.D. Thesis), Comp. Sci. Dept., Stanford Univ.

Wulf, W.A. (1972), "C.mmp: A Multi-Mini-Processor", Proc. FJCC.

A New Time-Domain Analysis

of

Human Speech

Janet Mac Iver Baker

Computer Science Department

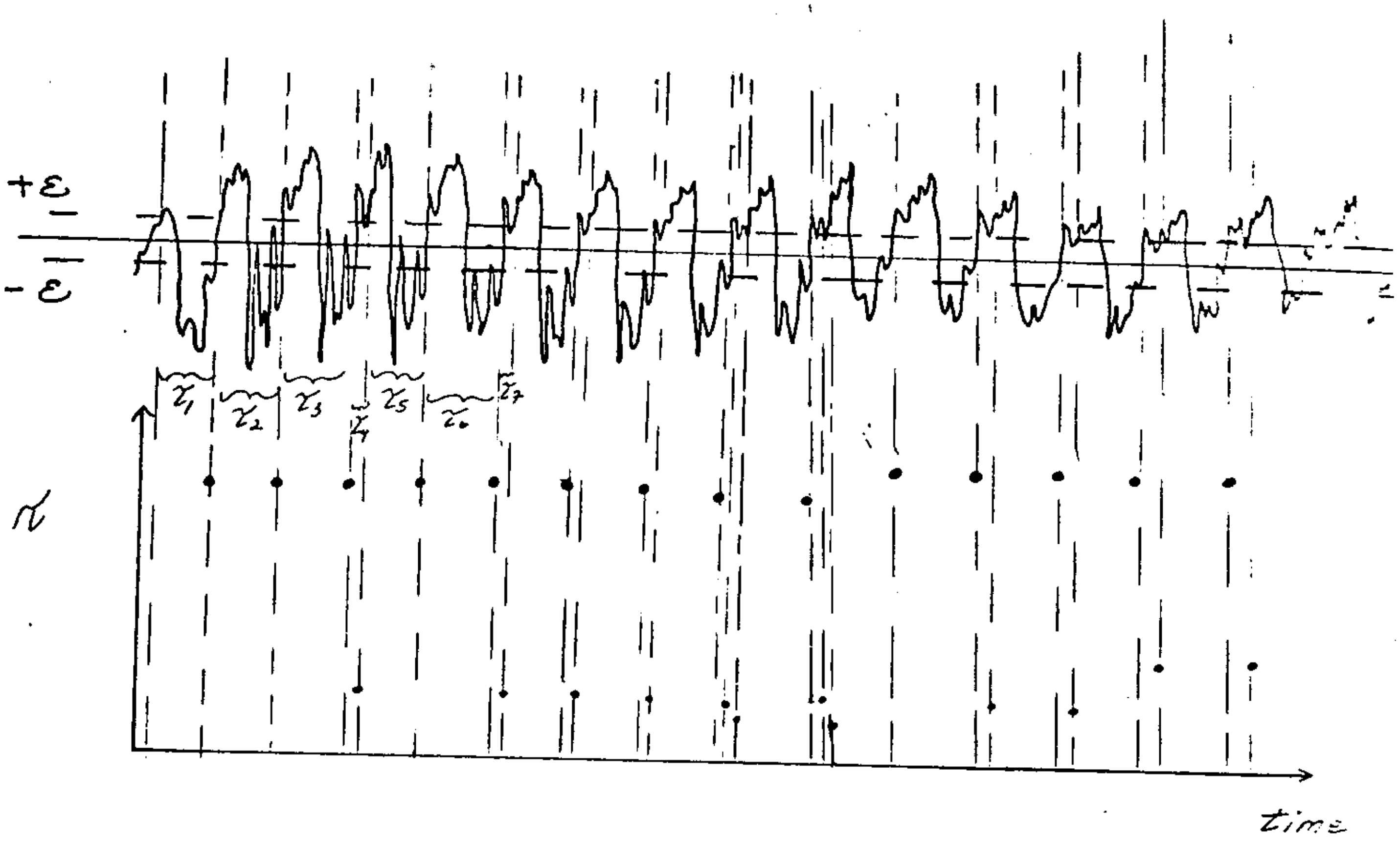
Carnegie-Mellon University

April, 1973

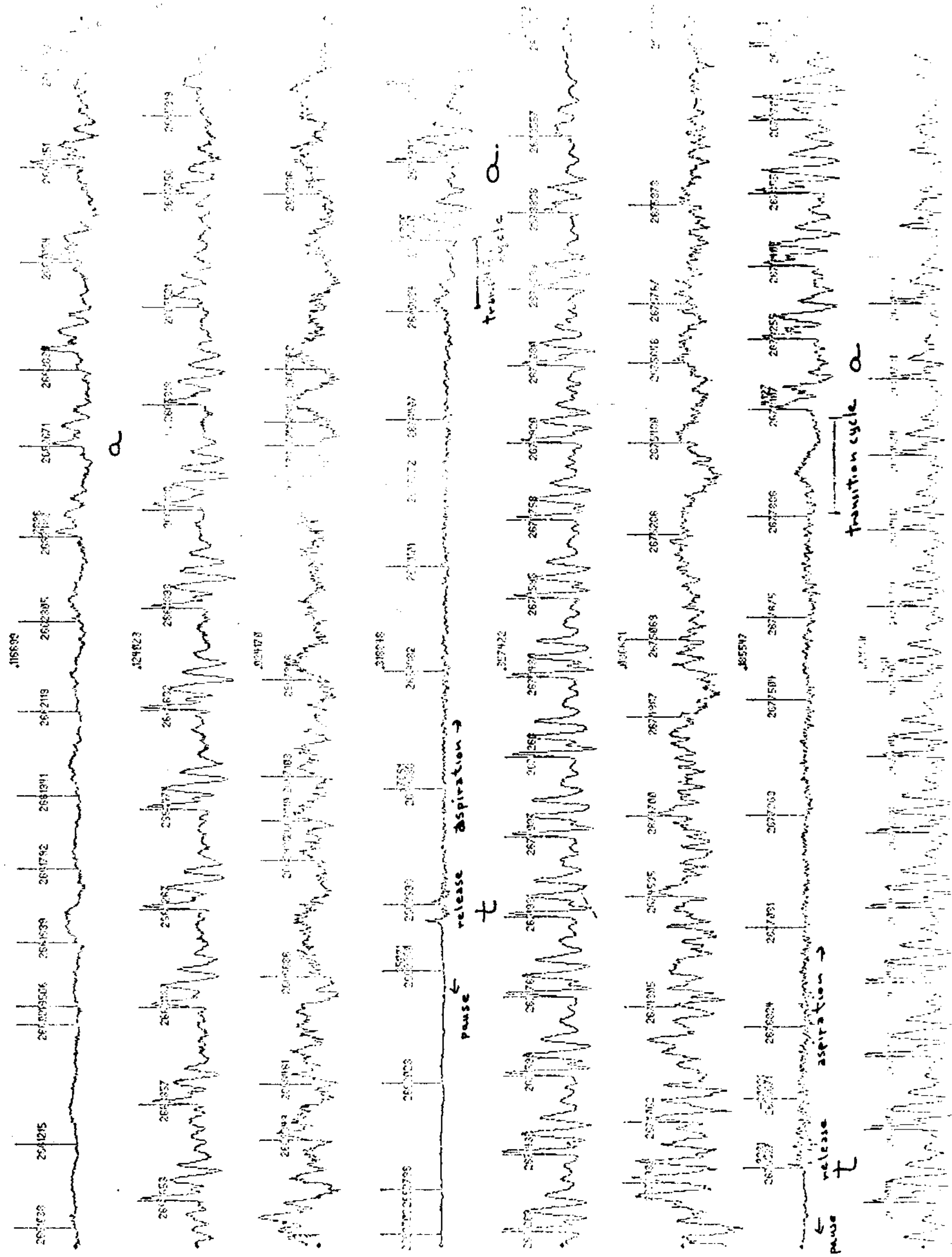
This paper concerns the application of a new time-domain technique to the analysis of complex acoustic signals such as human speech. The chief advantage of this method is its precise temporal resolution allowing exact timing of articulatory events within a sample of speech; that is, no bandwidth limitation is present. This temporal resolution is most significant for characterizing fast transitional regions such as occur at vowel-consonant and consonant-vowel boundaries and within stop consonants. We generate visual displays of waveform up-crossings in time, derived directly from the acoustic waveform itself.

The impetus for our work comes from two sources: 1)First are the studies by Licklider and his colleagues who 25 years ago demonstrated the intelligibility of infinitely clipped speech. This showed that sufficient acoustic speech information is encoded in the zero-crossings of the waveform itself. Given the redundancy of speech such information is most probably encoded by other aspects of the waveform. As it happens though, zero-crossings or up-crossings are easy to see and extract from the waveform. 2)The second motivation for this work comes from neurophysiological research on the auditory information processing of the ear itself. Basically the ear processes an incoming signal in at least two widely recognized manners. The first is analysis in the frequency-domain and is analgous to a kind of filter bank where different neurons along the basilar membrane respond to different frequency ranges; that is, a given neuron fires if it detects a signal of sufficient intensity within a particular frequency range. Neurons also code information in the time-domain in a manner known as phase-locking. Given a waveform, a phase-locking neuron responds by firing once, phase consistently, for each cycle or integer number of cycles within the waveform. The technique we are using is directly analagous to this latter time-domain coding technique.

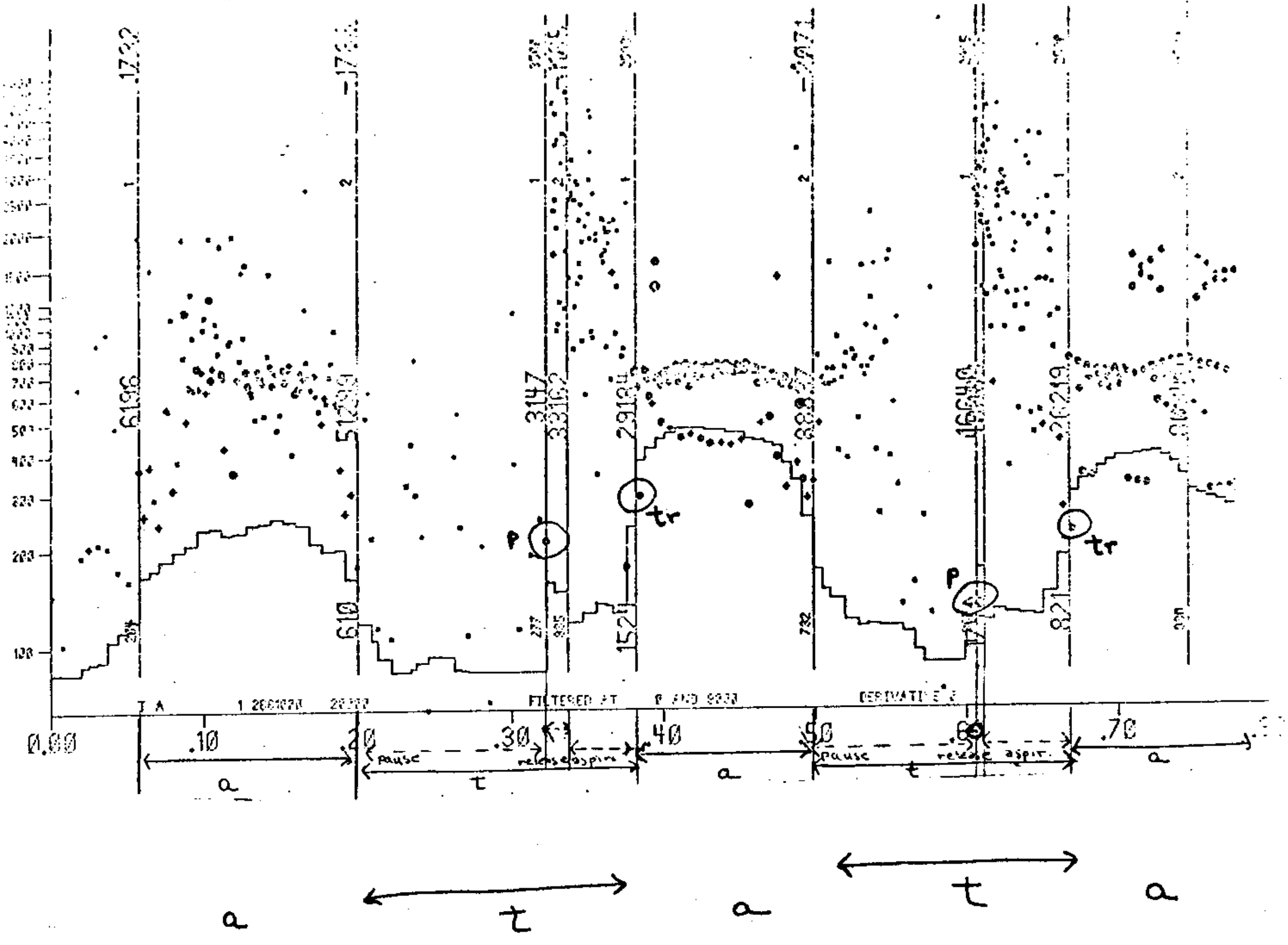
We generate our visual displays as follows: A zero-axis is drawn horizontally through the center of the acoustic waveform. We note the exact time when the waveform crosses this axis in an upward direction. In actuality, we usually record only those up-crossings which exceed some threshold amplitude, epsilon, set slightly above the horizontal zero-axis. This threshold tends to preclude low amplitude background noise. We measure each interval between successive up-crossings and plot these as a function of time in our displays. Therefore each up-crossing in the acoustic waveform is represented by a discrete dot in our displays. In fact, we actually plot on a log scale, the inverse of the interval between successive up-crossings along the vertical Y-axis and time along the horizontal X-axis. This yields a display which superficially resembles a kind of spectrographic display. (N.B. For those readers familiar with neurophysiological studies of single unit responses, this display is directly analagous to an "instantaneous frequency" plot and functionally analagous to a phase-locking phenomenon.) We also display a rough intensity measure by means of a Z-axis modulation. That is, the size of a dot representing a given cycle is proportionate to the log of the greatest intensity achieved during that cycle. This dot size intensity measure in our up-crossing displays is analagous to the intensity measure expressed in spectrograms.



The idea of looking at zero-crossing measures per se is not in itself conceptually new. However, in contrast to most other investigators who have used zero-crossing measures to analyze speech, we do not average our up-crossings over a fixed interval of time. Reasons for this will be discussed shortly. First of all it is important to be aware that the chief motivation for many zero-crossing studies has been in searching for an inexpensive way to find frequency domain acoustic features, such as formants. This method avoids the computations required for Fourier transforms, for example. In order to decrease the expense and variability in examining individual cycles,it was easy to to compute an average cycle length by simply counting the number of zero-crossings occurring during a given time interval. This procedure has two major consequences: 1)the perfect time resolution inherent in the time-domain is lost when crossings are averaged; that is, a bandwidth limitation is introduced, 2) the conventional acoustic features extracted are usually less precise and more variable than the same acoustic features extracted directly with a frequency-domain analysis. Our reason for not averaging up-crossings is that in the speech waveform itself there are significant acoustic features which only last for one or a few cycles in duration. If cycles are averaged, this information is irrevocably lost. Such transient events frequently occur at vowel-consonant and consonant-vowel boundaries as well as between other acoustically distinct regions, within stop consonants for example. In the waveform shown here of the nonsense word "ă tăt' ă" (stress on the second syllable), some of these short duration features can be seen. For example, one such feature often occurs at the transition from a stop or fricative to a following vowel. We find there exists a relatively long and intense cycle between the consonant and vowel. Sometimes there are several such cycles before the vowel. On our displays this phenomenon appears as a relatively low frequency large dot, or sometimes several, immediately preceding the vowel. The occurrence of this transition cycle(s) coincides with the upswing in energy from the consonant to the vowel. In our up-crossing display of the same utterance we have circled these transition cycles and labeled them "tr".



(each line of the above waveform has been individually amplitude normalized)


0.00
.10
.20
.30
.40
.50
.60
.70
6196
3147
1524
16648
26219
821
610
FILTERED AT
DERIVATIVE 2
P
P
tr
tr
pause
release aspir.
pause
release aspir.
a
t
a
t
a
a
t
a
t
a

Another area where consistent time-domain features can be seen is during the course of stop consonants. Both "t's" shown in this example consist of three distinct regions: the initial pause, a release, and aspiration. The pause is characterized in the waveform as a region of very low energy, irregular activity which is terminated abruptly by the release characterized by many greater amplitude, high frequency cycles. In the up-crossing displays, the initial pause appears as either one or a few outstandingly low frequency dots immediately preceding the release activity. In our display, these dots are circled and labeled "p", for "pause dot". The precise duration of any unusual cycle or sequence thereof may be trivially determined by noting the corresponding dot's(s') height(s) on the vertical axis. We have seen these pause and transition dots in literally thousands of our displays oı utterances spoken by both men and women.

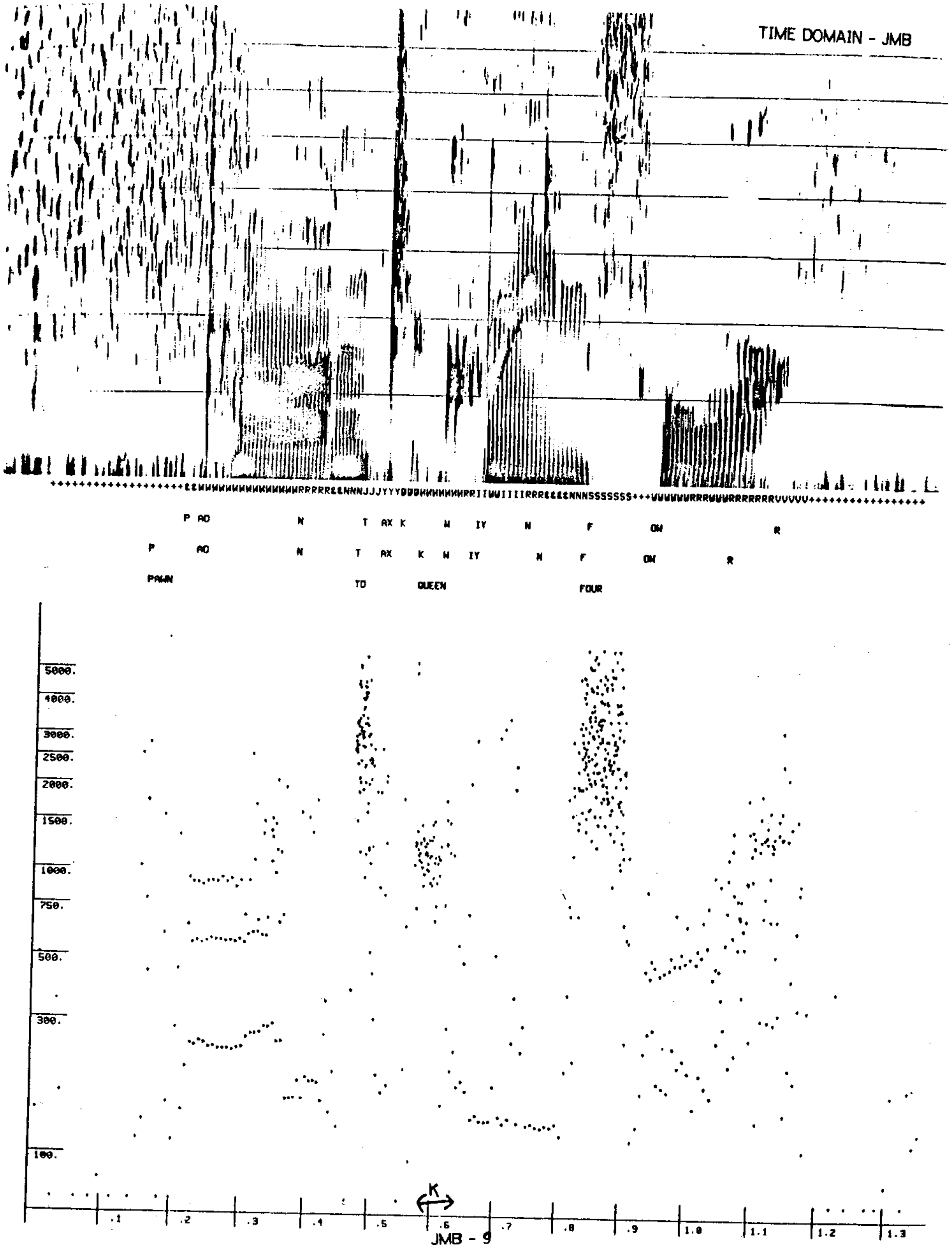
In the up-crossing display here, there is also an example of an automatic boundary segmentation as evidenced by the vertical lines drawn through the display*. These vertical segmentation lines were drawn automatically solely on the basis of discontinuities in the signal intensity functions. These intensity functions were computed pitch-synchronously and are represented by the line graph at the base of the plot. As easily seen, although the dot features and vertical line segmentation were independently derived, the times at which they occurred were rather close.

Another finding with this unaveraged up-crossing analysis is the presence of visually easily distinguishable patterns for fricatives and stops, e.g."p", "t", and "k" distinctions. We performed the following experiment with 10 people, most of whom had no experience with spectrograms or other speech research. First of all, we had a stack of photographs of our displays (with no segmentation lines or even any vertical or horizontal axis markings). The photographs showed displays of nonsense words all in the form of ə CVC' (stress on the CVC syllable), spoken by both male and female speakers. In a typical experiment, we would give a subject three model pictures, each of a nonsense word containing "p", "t", and "k" in the initial consonant position respectively. We would then show him where in the pictures these consonants were located. Next we handed him a stack of unsorted pictures and instructed him to sort these into four piles, one each for those that contained "p", "t", or "k" in the same position as in the model pictures, and one pile for those pictures that did not look like any of the model

*(automatic segmentation algorithm and implementation done by James K. Baker)

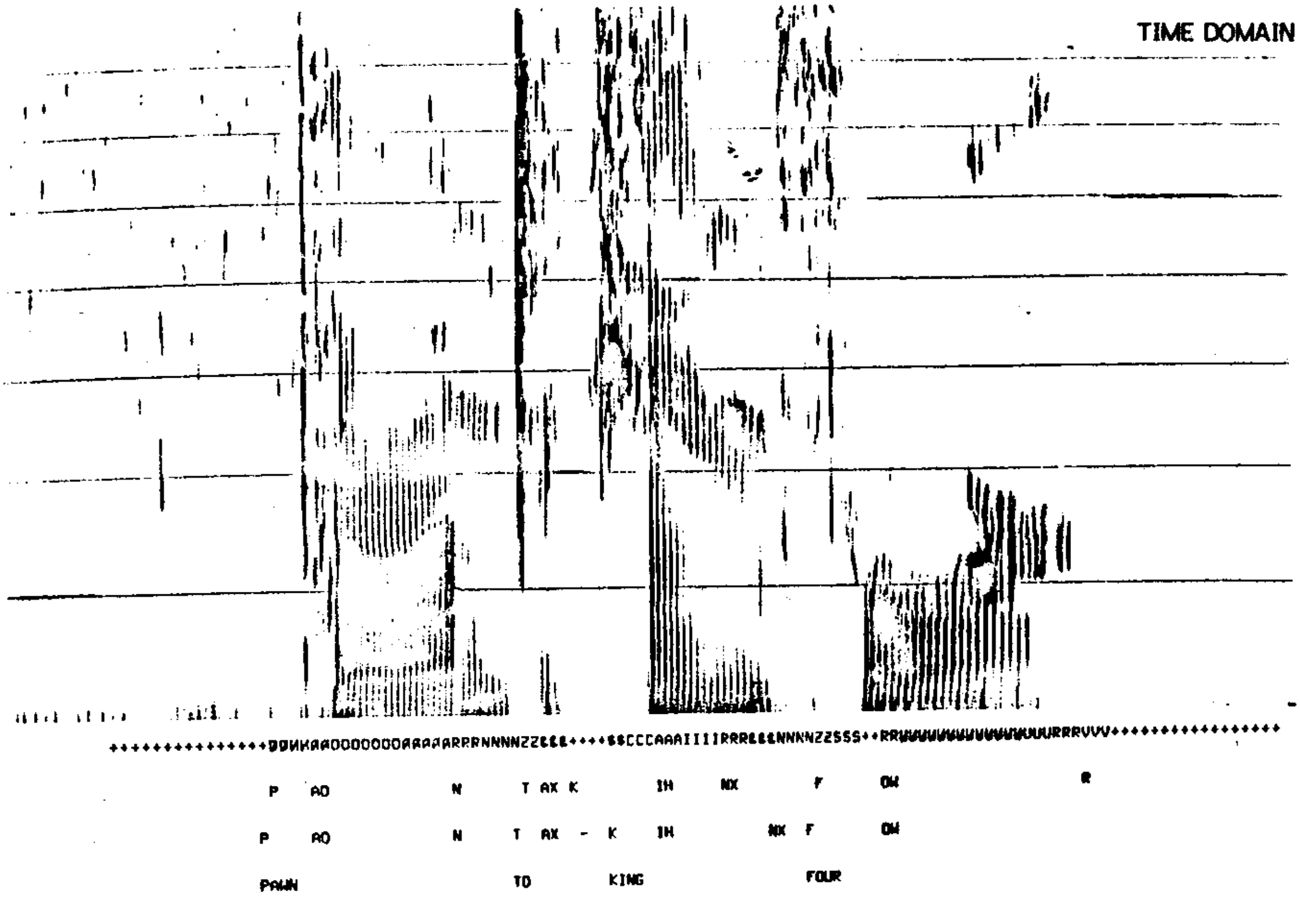
pictures. Despite speaker, allophone, and vowel differences between the model pictures and those sorted, subjects were able to distinguish "p's", "t's", and "k's" from each other about 80% correctly on a first try, regardless of the subject's familiarity with speech research. Additional practice improved scores.

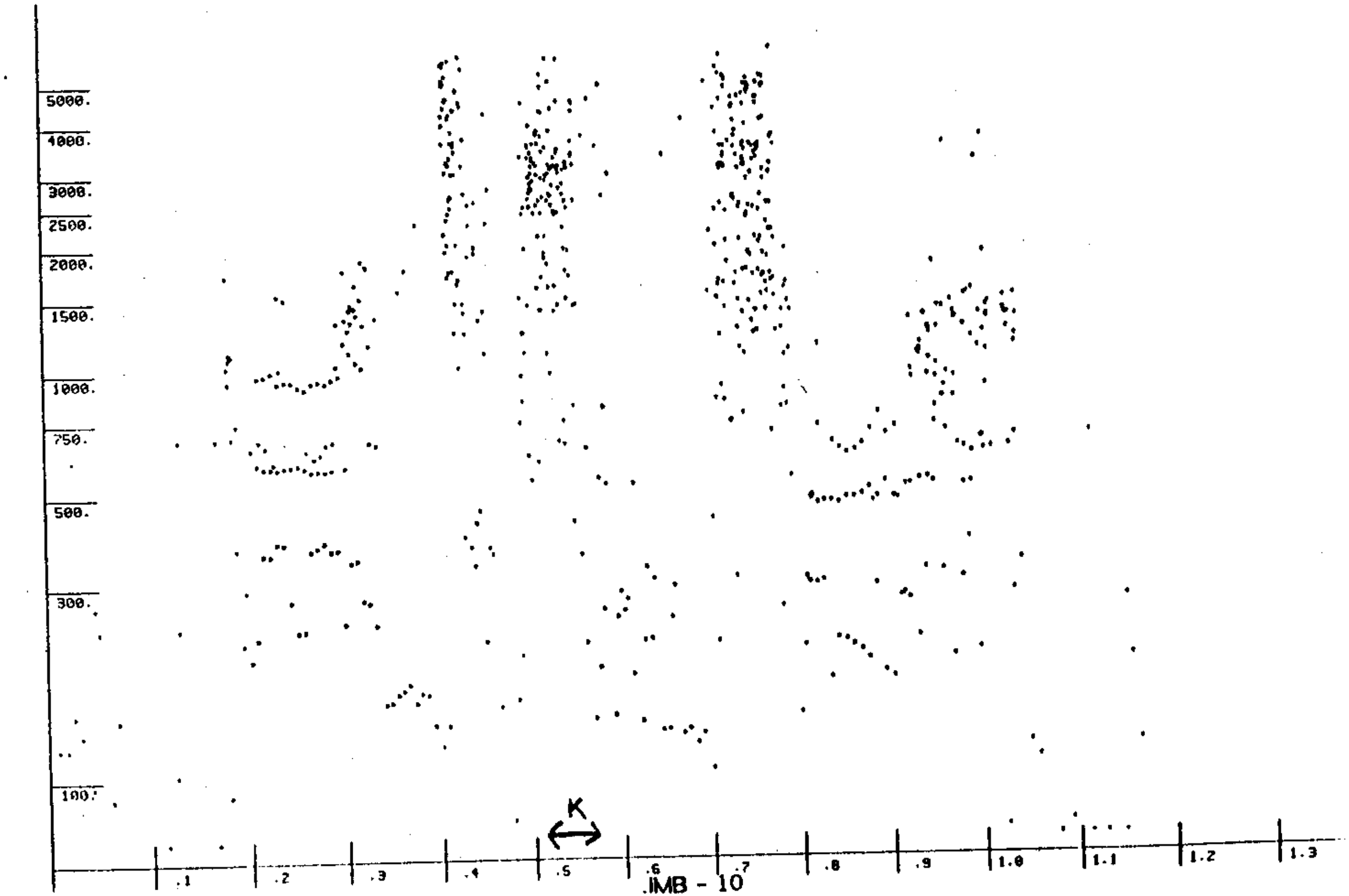
At this point two issues arise. First is the issue that the ability of humans to distinguish these phoneme patterns does not guarantee that an automatic speech recognizer can be programmed to do as well. The dot pattern itself is complex and it is not clear exactly which visual features subjects use in making their decisions. Although we do have some specific ideas about which acoustic features are most reliable for these discriminations, we have not yet subjected a large sample of data to an automatic testing program to determine which features are most reliable and when. This brings us to the second major issue, the problem of allophones and coarticulation effects. Different allophones of the same phoneme often are acoustically very different. An extreme example of this phenomenon appears in the following pictures (spectrograms and up-crossing displays) of the connected speech utterances "Pawn to king four" and "Pawn to queen four". The "k" in "king" differs radically from the "k" in "queen". The most obvious difference is the lower frequency components in the "k" of "queen", probably due to the lips' rounding, effectively lengthening the vocal tract.


TIME DOMAIN - JMB
++++++++++++++++++++££WWWWWWWWWWWWWWWWWRRRRRR££NNNJJYYYDDDWWWWWWWRRIIWWIIIIIRRRR£££NNNSSSSSSSS+++VVVVVVRRRVVVVRRRRRRRVVVVV++++++++++++++++++
P AO N T AX K W IY N F OW R
P AO N T AX K W IY N F OW R
PAWN TO QUEEN FOUR
5000.
4000.
3000.
2500.
2000.
1500.
1000.
750.
500.
300.
100.
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
1.1
1.2
1.3
K

JMB - 9

TIME DOMAIN - JMB





JMB - 10

Therefore we are starting an extensive investigation examining time-domain acoustic features of all the allophones of the stops and fricatives common in English. We will examine in detail a large sample of these utterances, about 500 each from seven or eight male and female speakers to determine the most reliable cues for stop and fricative discriminations and ascertain which allophones and coarticulation effects must be dealt with explicitly. This kind of basic research is essential for the development of automatic speech recognition systems.

In summary, we find that, due to its precise temporal resolution, this up-crossing analysis (and presumably other related time-domain analyses) is particularly well-suited to examining fast transitional regions of acoustic signals. In our displays we often find, particularly for traditionally difficult stop and fricative discriminations, visually distinct patterns, consistent across male and female speakers. In addition, this technique is generalizable to any waveform and is particularly applicable to complex waveforms characterized by rapid frequency changes. On the basis of both theoretical considerations and the empirical results of our studies, in conjunction with other studies in speech analysis, we feel that future automatic speech recognition systems may be more successful by incorporating both time and frequency domain analyses, rather than either separately. Although there is a great deal of redundancy in terms of the information yielded by both domains, frequency-domain analyses will generally be more powerful for steady or quasi-steady state phenomena, e.g. stressed vowels, while time-domain analyses will usually be most effective for studying fast transient phenomena, e.g. stop consonants.

MACHINE-AIDED LABELING OF CONNECTED SPEECH

James Baker
Computer Science Department
Carnegie-Mellon University
April, 1973

This paper presents preliminary results of a project for machine-aided segmentation and labeling of connected speech. The segmentation and labeling problem is reformulated as a problem of searching for a minimum cost path in a network. Such abstract formulation permits the construction of a system which avoids the complexities of a system built by ad hoc methods from the acoustic and phonological properties of speech. That such a simple abstract model is adequate for this problem is demonstrated by a functioning program which is described by the pair of simple formulas (10) and (11). A program which uses more sophisticated acoustic observations and more complicated matching procedures is under development, but it is also an implementation of the same abstract model.

For research in the acoustic properties of speech it is imperative to have a large data base of speech utterances which have been reliably segmented and labeled. Each important event must be found and labeled as to time of occurrence. Let's restrict our attention to finding the beginning and ending time of each phone in a given utterance. A reliable method to do this labeling is to generate an appropriate display of the acoustic parameters and then have a trained person label the phone segments. But for a large data base of thousands of utterances, such hand labeling can be very time consuming. The goal of this project is a program which can automatically assign the labels to a connected speech utterance with the need for human intervention only on special problem cases.

Assigning labels to a speech utterance to be used in a data base is a very different problem from segmentation and labeling for automatic recognition because the utterance is known. The program is given either a phonetic transcription or can approximate one from an orthographic transcription plus a phonemic dictionary. On the other hand, the labeling must be as complete and reliable as possible whereas a general recognition system should be able to tolerate incomplete labeling or even errors.

A second goal of this project is the exploration of the application of stochastic models to automatic speech analysis. A general technique has been developed for combining information from several sources when each source alone would result in a significant number of errors. Clearly there are many problems in speech recognition which fit this general framework. The unifying principle is a generative stochastic model for fitting a sequence of states to errorful data from several sources. Machine-aided segmentation and labeling has been approached as a specific application of this general technique.

To relate the phones to the acoustic observations requires knowledge of the acoustic phenomena which are expected with each phone. In line with the probabilistic approach, each phone is assumed to be associated with a stochastic process which produces acoustic parameter values for any instance of the phone. The statistical properties of the stochastic process associated with any particular phone are to be estimated from the occurrences of the phone in the part of the data base which have already been segmented and labeled. Thus a non-negligible data base must first be analyzed by hand before the machine-aided system can be started.

Each acoustic observation is to take a value from a finite set D. Assume that for each phone P there is a positive-integer-valued random variable Z_P and a family of random variables $X_P(1)$, $X_P(2)$, ... , $X_P(Z_P)$ with values in D. Let $f_{P,n}$ be the conditional probability function

$$(1)\ f_{P,n}(x_1, x_2, \ldots, x_n)$$

$$= \mathrm{PROB}(X_P(1)=x_1, X_P(2)=x_2, \ldots, X_P(n)=x_n \mid Z_P=n)$$

Let $g_P(n) = \mathrm{Prob}(Z_P=n)$. The interpretation is to be that Z_P is the duration of an instance of phone P and $X_P(1)$, $X_P(2)$, ... , $X_P(Z_P)$ are the acoustic observations made during that instance of P.

Let V(1), V(2), V(3), ... , V(T) be the sequence of observations made for the utterance being analyzed. Let P(1), P(2),

.... , P(R) be the sequence of phones in the utterance. Use the notation V[t1:t2] as an abbreviation for the sequence V(t1), V(t1+1), ... , V(t2-1), V(t2). Let U(1), U(2), ... , U(R) be a sequence of putative starting times for the phones. That is, U(1) < U(2) < ... < U(R) and for each k, P(k) is supposed to last from observation V(U(k)) to observation V(U(k+1)-1). Suppose a set of observations V[1:T] and times U[1:R] are produced by applying in succession the stochastic processes for each of the phones P(1) through P(R) and concatenating the observations, the individual processes being independent. Then the probability of producing the observed sequence is

$$(2)\ \mathrm{PROB}_{P[1:R]}(\ V[1:t],\ U[1:r]\ )$$

$$= \prod_{k=1}^{R} (f_{P(k),U(k+1)-U(k)} (V[U(k):U(k+1)-1]) g_{P(k)} (U(k+1)-U(k))$$

The segmentation and labeling problem consists of finding the correct set of values for the sequence U[1:R]. We shall use a maximum likelihood estimation scheme. Pick for U[1:R] that sequence that maximizes Prob(V[1:T], U[1:R]) for the given observations V[1:T]. The problem of finding U[1:R] is equivalent to finding the best path through a binary decision tree where each node at level t represents a decision of whether or not there is a phone boundary at time t. Subject to the constraint that there are R phones, there are

$$(3)\quad \binom{T-1}{R-1} = \frac{(T-1)!}{(R-1)!(T-R)!}$$

paths through this tree. This number is prohibitively large (if an observation is made every centisecond and the utterance lasts two seconds, then T=200), so some reduction is necessary.

Note that our model is such that given k and U[k:R] we can evaluate

$$(4)\ \mathrm{PROB}_{P[1:R]}(\ V[U(k):T],\ U[k:R]\ )$$

$$= \prod_{j=k}^{R} (f_{P(j),U(j+1)-U(j)}(V[U(j):U(j+1)-1])g_{P(j)}(U(j+1)-U(j))) \quad .$$

that is, the probability does not depend on U[1:k-1]. Also note that

$$(5)\ \mathrm{PROB}_{P[1:R]}(V[1:T],\ U[1:R]\ )$$

$$= \mathrm{PROB}_{P[1:R]}(V[1:U(k)-1],\ U[1:k])\mathrm{PROB}_{P[1:R]}(V[U(k):T],\ U[k:R])$$

Therefore if at any node of the tree corresponding to a particular k and U(k) we have evaluated Prob(V[1:U(k-1)], U[1:k]) then the subsequent analysis depends only on k and U(k). That is, for the purpose of analyzing V[U(k):T] and U[k:R] we can identify all nodes of the tree which correspond to the same pair k and U(k). Since we are only interested in the best U[1:R], we associate with this combination node the maximum of Prob(V[1:U(k+1)-1], U[1:k]) over all the nodes which are combined. This identification reduces the tree to a network whose nodes correspond to the two-dimensional set of values $(k, U(k))$, where $1 \le k \le R$, $1 \le U(k) \le T$. Procedures for finding the best path through such a network have been extensively investigated. A simple, computationally efficient, procedure is dynamic programming.

To facilitate dynamic programming, introduce the function

$$(6)\quad A(k,\ t) = \max_{\substack{U[1:k] \\ U(k)=t}} \{PROB(V[1:t-1],U[1:k])\}$$

That is, $A(k,t)$ is the probability along the best path leading up to the (k,t) node. A may be calculated by

$$(7)\quad A(k,\ t) = \max_{j}\{\ A(k-1,\ t-j) f_{P(k),j}(V[t-j:t-1]) g_{P(k-1)}(j)\}$$

Let $J(k,t)$ be the value for which this maximum is achieved. Then after A and J have been calculated for the whole network, the best path through the network is obtained by

$$(8)\qquad U(k) = U(k+1) - J(k+1,\ U(k+1))\ .$$

If we are willing to assume that $X_P(1), X_P(2), \ldots, X_P(Z_P)$ are independent and identically distributed and that

$$(9)\qquad g\ (n) = (1-a)a^{n}\ ,\ \text{for some a independent of P,}$$

then an even simpler computation is possible. It is not claimed that these additional assumptions are realistic. However, some examples will be given to show that even with these assumptions and very crude acoustic observations the model can produce reasonable segmentation and labeling.

The extra assumptions allow us to ignore the durations of the phones by factoring out a factor which is the same for all paths through the network. Reformulate the network, ignoring duration information. Let the node (k,t) correspond to the event $U(k) \leq t < U(k+1)$ with U(k) , otherwise unrestricted. Let B(k,t) be the probability along the best path leading to (k,t). Then B may be calculated by

$$(10)\quad B(k,\ t) = \{\ \mathrm{Max}\{\ B(k-1,\ t-1),\ B(k,\ t-1)\ \}\ \}\mathrm{PROB}(X_{P(k)} = V(t))$$

Then the sequence U[1:R] may be calculated by

$$(11)\ U(k) = \mathrm{Max}\{\ t \mid t<U(k+1) \text{ and } B(k-1,\ t-1)>B(k,\ t-1)\ \}$$

Since some of the simplifying assumptions are admittedly unrealistic, the model must be tested in actual use. First we must find some measurable parameter to use as the sequence of acoustic observations V[1:T]. The better the parametric representation distinguishes the phones, the more the conditional probability function $f_{P,n}$ will be concentrated in different regions for different phones, and the better the system will work. For final production runs the best parametric representation available should be used. For preliminary testing, however, there is an advantage to using a less precise parametric representation. If the system is to be of significant value it must be robust. It must be able to operate in environments in which the direct acoustic observations do not well characterize the underlying phones. Besides, if the system works with a crude parameterization, it can be used to help assemble the data base needed for finding and testing a more refined parameterization.

The parameter which has been used is the output of a crude local-pattern-match phonetic recognizer. The output of the recognizer is a label which is intended only to be an approximation to the associated phone. The conditional probabilities are given in Table 1. Each row corresponds to a given phone, and the columns are the possible labels that the recognizer might assign. This recognizer frequently confuses phones within a class, but it can generally distinguish among broad classes.

The output of the system is shown for three chess utterances. The six line graphs in each figure are the six parameters that are input to the pattern recognizer. They are intensity measures of the signal passed through each of five octave-wide band-pass filters and of the unfiltered signal. The line immediately below the graphs is the sequence of labels assigned by the recognizer. This is the sequence V[1:T]. There is one label for each centisecond. The phones as segmented and labeled by a program using formulas (10) and (11) are displayed on the second line. Each phone is printed at the position that indicates the time at which the phone begins. The hand segmentation data is given on the third line and the orthographic transcription on the fourth. The phone sequence for the program is derived from a phonemic dictionary, so it differs in places from the hand labeled sequence.

In evaluating a system of this type it is important to note the different kinds of errors and their effects. There are three important kinds of errors: (1) The sequence of phonetic labels may differ from the correct sequence. (2) A boundary position may be shifted between two phones which are otherwise correctly placed. (3) A phone may be so misplaced that its machine-labeled segment does not intersect the correct segment. The different kinds of errors have various effects in a total man-machine system.

The first type of error results from an inadequately specified phonetic input. Problems may result especially when the input sequence is derived by rule from a phonemic dictionary. The algorithm is not permitted to alter the nominal phonetic sequence which it is given. To reduce errors of this kind more sophisticated phonological rules must be combined with the phonemic dictionary, or the utterance must be transcribed by hand. Note, however, that for the purpose of collecting statistics for machine recognition pattern matching algorithms, the best labeling may in fact be that which is derived from a dictionary. Then the statistics are grouped according to the dictionary phonemic label, which is just what is needed for pattern matching statistics.

Some errors of misaligned boundaries are inevitable. In fact, the format of the output has some error built in since it assumes that the phones can be occupy non-overlapping time segments. It is especially hard for the program to accurately place the boundaries between vowels and semi-vowels or nasals. More accurate and detailed acoustic observations may help, but the output must still be checked and corrected by hand.

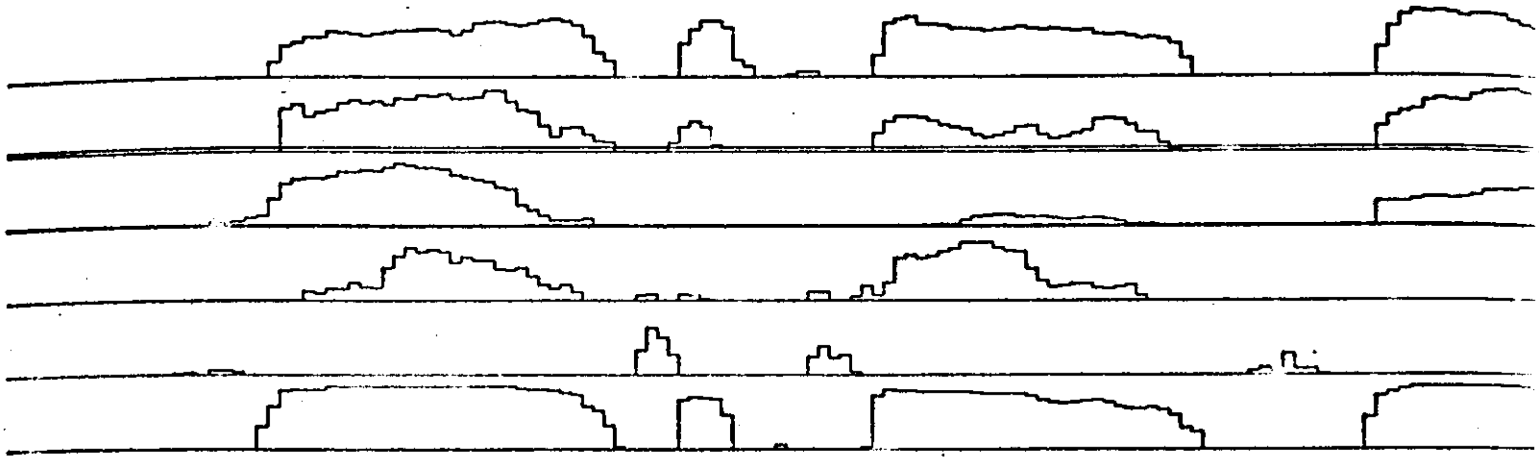
The third type of error is the most serious. It implies that several boundaries are misplaced and that the underlying sequence of states in the path through the network is not following the actual sequence of phones at all. Such errors are easy for a human checker to detect, but to correct them may require that the whole utterance be hand labeled. Unless the number of errors of this type is small, the machine-aided system is not successful.

No systematic performance evaluation has been attempted, since the program is still in a preliminary version. A file of hand segmented data must be built up to establish statistics for estimating the conditional probability distributions of the X_P's. It may be necessary to use the more complete model given by formulas

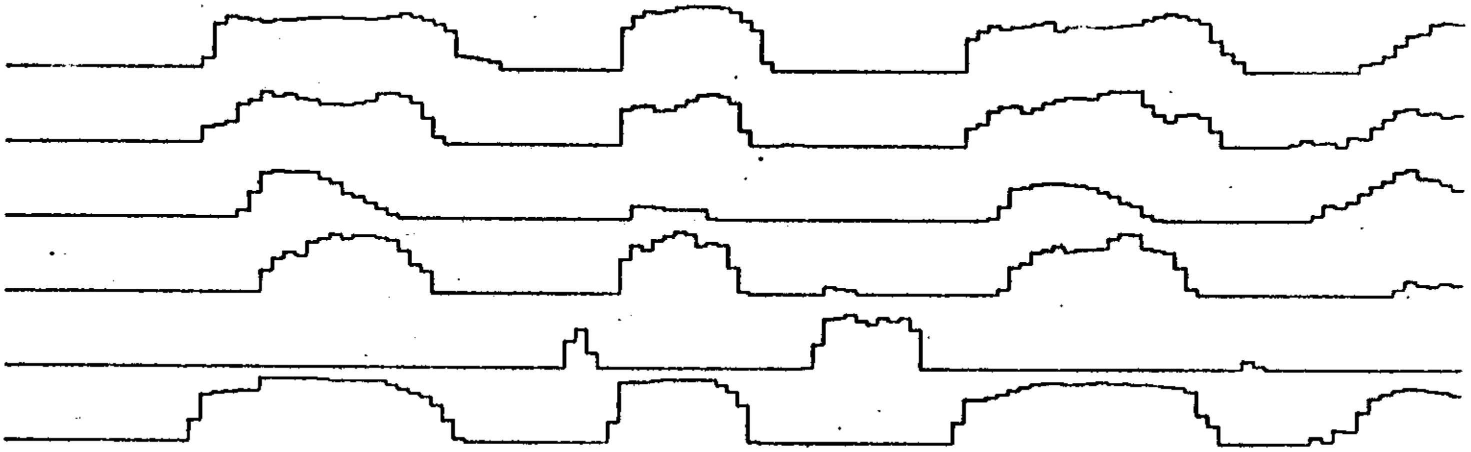
(7) and (8). Duration information is a valuable tool for preventing the type-3 errors (which still occur under certain conditions). Other parametric representations of speech must be explored, especially if the system is to work without tuning to individual speakers. The pre-processor which is being used presently is tuned to the extent of having the speaker produce one prototype version of each phone. When this crude tuning is omitted the quality of the acoustic obsevations is degraded sufficiently to introduce type-3 errors in many utterances.

	-	•	F	C	*	$	S	Ž	D	D/	J	V	W	M	U	R	L	Y	I	U/	A	£	•	N	W/	E/	E	O	A/	H/
-	90	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	2	27	2	2	2	2	2	2	2	2	2	2	2	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
B	42	7	1	1	1	1	1	1	4	1	1	1	1	13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T	10	6	53	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D	3	5	3	3	3	3	3	3	21	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
K	11	53	[illegible]	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
F	32	1	39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
V	16	27	5	2	2	2	2	2	2	2	2	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
TH	10	1	49	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DH	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	3	3	3	3
CH	3	3	3	3	3	3	9	3	3	3	3	3	3	6	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
JH	3	3	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
S	8	1	59	1	1	1	8	2	1	1	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&	2	2	17	2	2	2	9	7	2	2	2	2	2	9	2	2	2	2	2	4	2	2	2	2	2	2	2	2	2	2
SH	1	6	20	8	1	37	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ZH	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
HH	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
M	3	3	3	3	3	3	3	3	15	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	3	3
N	1	5	0	0	0	0	0	0	2	0	0	9	0	54	2	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0
NX	1	1	1	1	1	1	1	1	1	1	1	12	3	32	3	1	1	1	1	25	1	1	1	3	1	1	1	1	1	1
W	2	2	2	2	2	2	2	2	2	2	2	2	2	10	6	2	2	5	2	37	2	2	2	2	2	2	2	2	2	2
R	1	8	1	1	1	1	1	1	1	4	1	3	1	15	1	7	1	1	1	16	1	1	1	1	23	1	1	5	1	1
L	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
UW	1	1	1	1	1	1	1	1	1	1	1	4	1	18	1	1	1	1	1	41	1	1	1	1	1	1	1	1	1	1
UH	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
OW	1	1	1	1	1	1	1	1	1	1	1	1	1	6	1	1	13	1	1	24	1	1	1	1	23	1	1	1	1	1
AO	1	1	1	1	1	1	1	1	1	2	1	1	1	1	7	8	1	1	1	1	2	2	1	1	41	1	1	2	17	1
AA	2	2	2	2	2	2	2	2	2	8	2	6	2	12	8	2	2	2	2	2	2	2	2	2	14	2	2	2	2	2
AH	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
ER	3	3	3	3	3	3	3	3	3	3	3	3	3	11	3	3	3	14	3	3	3	3	3	3	3	3	3	3	3	3
AE	2	4	2	2	2	2	2	2	2	2	2	2	6	6	2	2	2	2	2	2	2	6	12	2	2	2	12	2	12	2
EH	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
IH	1	1	1	1	1	1	1	1	1	1	1	1	2	3	1	1	1	28	2	10	1	7	1	25	1	11	1	1	1	1
IY	1	1	1	1	1	1	1	1	2	1	1	1	1	16	1	1	1	20	1	2	1	1	1	18	1	22	1	1	1	1
AW	1	5	1	1	1	1	1	1	2	6	1	2	4	35	2	1	1	7	1	16	1	3	1	1	1	1	1	1	1	1
EY	1	1	1	1	1	1	1	1	1	5	1	1	1	3	1	1	1	24	1	1	1	5	1	28	1	7	1	1	1	1
AH	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
AY	0	2	0	0	0	0	0	0	2	3	0	2	1	0	4	1	0	0	2	4	7	14	0	10	14	1	6	2	14	0
OY	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
WH	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
EL	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	8	3	3	18	3	3	3	3	3	3	3	3	3	3
EM	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
EN	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
DX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Q	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
P/	60	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T/	57	20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
K/	39	17	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

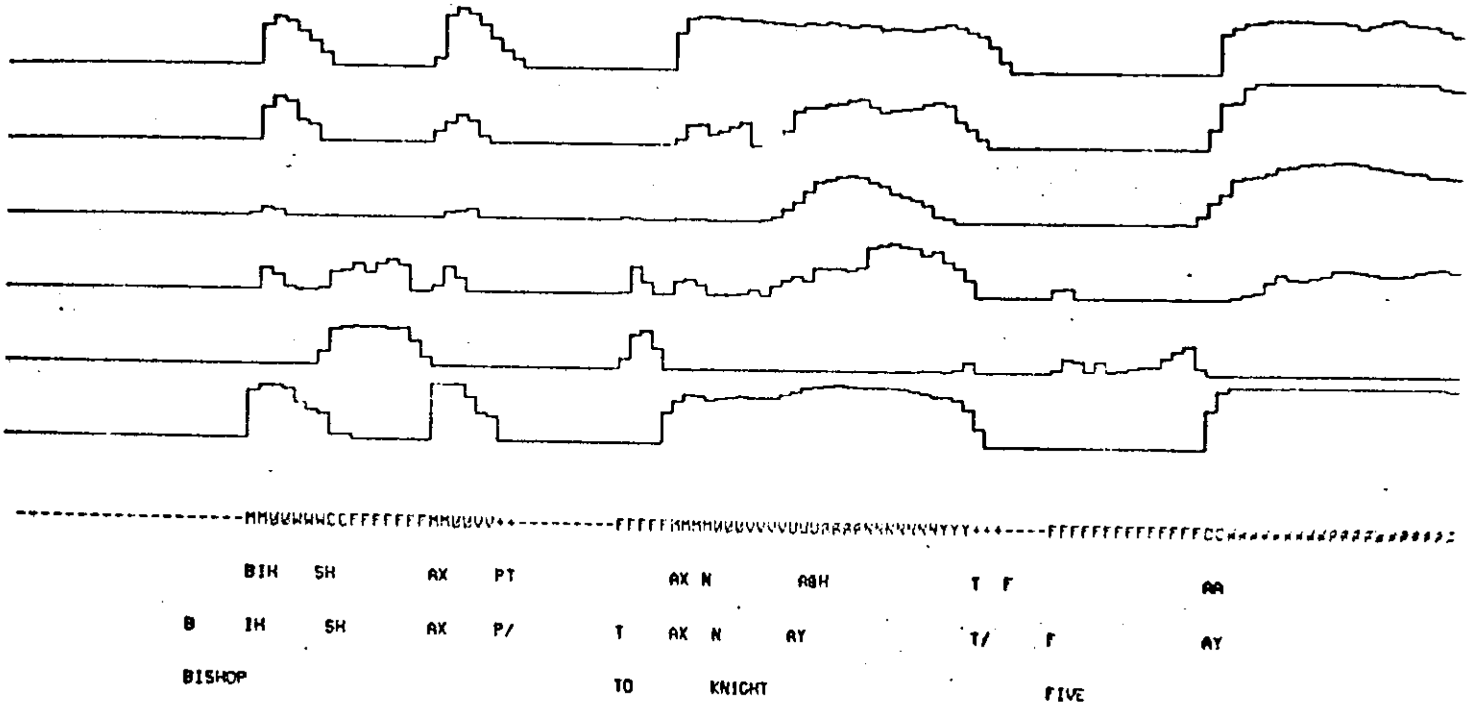
TABLE 1



```
-------------FF-+++++RRRRRRRRRRAAAPAPAAAAAARUUUMMMMMM--FFFFVWUMT++---+++FFFF++WWYYYYEEEEEENNNMMMMMMUUUMMMMM-----FFFFFFF---MMWWWLLLL##--

                P     AO                          N     T     AX   K           IH           NX            F                OW       R

              P     AO                        N         - T   AX   -       K   IH           NX          -   F         -    OW       R

              PAWN                                        TO               KING                             FOUR
```



```
--------------MMMWWAAAAAAANNNNNNNNYYYDD++++------FFF--NNNNEEENNNNDD--++FFFFFFFFFFF---MMWWWEEEEAAEEEEEEYYYYWWW++FF--+++++VVVVVVVVVV

N      ABH            T            TEBH       K    S       N    ABH             T         AA      N

N    AY                  T/    T    EY         K/   S       N    AY               T/T -     AA

KNIGHT                         TAKES                           KNIGHT                          ON
```


-------------------MMWWWWWCCFFFFFFFFMMWWVV++---------FFFFFMMMMWWVVVVVUUUAAAANNNNNNNYYY+++----FFFFFFFFFFFFFFFCC
BIH SH AX PT AX N AGH T F AA
B IH SH AX P/ T AX N AY T/ F AY
BISHOP TO KNICHT FIVE