

COMMUNICATION SCIENCES
AND
ENGINEERING

IX. SPEECH COMMUNICATION*

Academic and Research Staff

Prof. K. N. Stevens	Dr. Margaret Bullowa	Dr. D. H. Klatt
Prof. M. Halle	Dr. H. Funkenstein	Dr. Paula Menyuk
Prof. W. L. Henke	Dr. A. W. F. Huggins	Dr. J. S. Perkell
Prof. A. V. Oppenheim	Dr. Emily F. Kirstein	A. R. Kessler

Graduate Students

T. Baer	R. M. Mersereau	M. R. Sambur
R. E. Crochiere	B. Mezrich	J. S. Siegel
D. E. Dudgeon		V. W. Zue

A. SPEECH SOUND CATEGORIZATION WITHIN SETS BY CHILDREN

1. Problem

Children appear to make systematic substitutions in their production of speech sounds during the developmental period from two to seven years. These substitutions seem to occur most often among members of what has been termed "manner" speech sound sets (that is, sounds that are produced in the same manner, but in a different place such as /p/, /t/, /k/).¹ If this is true, then knowledge of how children divide the acoustic and articulatory space within these sets might provide evidence concerning the mode of speech sound differentiation used for speech sound categorization. We can determine whether or not there is a marked difference between the identification and reproduction of the differences between members of speech sound sets and the direction these differences take. We can also ask whether identification performance is more accurate than reproduction performance, or vice versa, for all speech sound sets and at all stages of development, or whether there are developmental shifts. Finally, we can explore the nature of the distinctions made by examining which features of the speech sound distinctions are confused and which are preserved in both tasks.

2. Subjects and Procedure

In the experiments reported here,² two were run to examine children's categorization of the following sound sets: (i) /p/, /t/, /k/; (ii) /b/, /d/, /g/; (iii) /f/, /s/, /ʃ/; and (iv) /w/, /l/, /r/. An adult female speaker recorded words with these sounds in initial position. "Pot", "tot", and "cot" were used in Set 1, "beer", "deer", and "gear" in Set 2, "feet", "seat", and "sheet" in Set 3, and "white", "light", and "write" in Set 4. There were 36 stimulus words in each set, and each word appeared 12 times in the set in a

*This work was supported by the U.S. Air Force Cambridge Research Laboratories under Contract F19628-72-C-0181, the National Institutes of Health (Grant 5 RO1 NS04332-09), the Joint Services Electronics Programs (U.S. Army, U.S. Navy, and U.S. Air Force) under contract DAAB07-71-C-0300, and by M.I.T. Lincoln Laboratory Purchase Order CC-570.

(IX. SPEECH COMMUNICATION)

random arrangement. Pictures of each word were in a horizontal display on a response box. A practice period preceded the presentation of the experimental stimuli of each set. The children were provided with the names of the three pictures. They were asked to identify the pictures, and then to tell the experimenter the names of the pictures. They were then told what they were going to hear on tape (to show the picture or say the word) and were asked to carry out both tasks.

In Experiment 1 children heard on tape "show me _____" and were asked to press the button under the picture of the word that they heard. After they completed the "show" task with all 4 sets, they heard on tape "say the word _____" and were asked to repeat the word. Alternate children received the "say" task first and then the "show" task. In Experiment 2 they either heard the "say the word _____" tape and were asked to say the word and push the button immediately or, alternatively, heard the "show me _____" tape and were asked to push the button and say the word immediately. The children in both experiments were attending a private nursery school. There were 19 children in Experiment 1, ranging in age from 3.3 to 5.3 years, with a mean age of 4.2 years. There were 14 children in Experiment 2, and their ages ranged from 3.1 years to 4.1 years with a mean age of 3.6 years. Thus, the experiments differed with respect not only to the required task but also the age of the participants.

3. Results

Table IX-1 shows the percentage of errors in each set for both tasks and for both groups of children. Those in Experiment 1 are labeled Group I and those in Experiment 2

Table IX-1. Percentage of error with sets for "show" and "say" by group.

Sets	Groups	Task	
		Show Errors (%)	Say Errors (%)
p t k	I	21.63	4.09
	II	26.92	9.82
b d g	I	11.69	9.50
	II	19.24	17.85
f s ſ	I	2.33	3.80
	II	7.34	9.52
w l r	I	14.91	10.96
	II	18.25	18.45

are labeled Group II. In general, the percentage of error was small. There were no significant differences in the percentage of error between groups for each set in both tasks, although the younger children (Group II) who also had to carry out both tasks immediately sequentially, tended to make more errors. The only significant difference in percentage of error that occurred between the identification and reproduction tasks was for the voiceless stop set (/p/, /t/, /k/). In this set both groups made significantly more errors in the "show" task than in the "say" task. In Group I the only significant difference in percentage of error between sets occurred in the "show" task. Significantly fewer errors occurred with voiceless stridents /f/, /s/, /s̥/ than with all other sets. These differences were significant above the .05 level (with /b/, /d/, /g/) or above the .01 level (with /w/, /l/, /r/ and /p/, /t/, /k/). In Group II significantly fewer errors occurred with /f/, /s/, /s̥/ in the "show" task than with all other sets. These differences were significant above the .02 level (with /p/, /t/, /k/) or above the .05 level (with /b/, /d/, /g/ and /w/, /l/, /r/). In the "say" task there were no significant differences between sets for Group I, but in Group II there was a tendency for more errors to occur in the "say" task for sets /b/, /d/, /g/ and /w/, /l/, /r/ than for the other two sets ($p < .20$). The fact that in the "show" task there was a significantly greater percentage of error with the voiceless stops (/p/, /t/, /k/) compared with all other sets is not a function of the particular stimulus words used nor the fact that it was the first set in the experiment. Halfway through the experiment with Group I the stimuli were changed to "pop", "top", and "cop" and the set was presented second, and the same kind of results were still obtained. Furthermore, the order of presentation could not account for the fact that the percentage of error in the "say" task was significantly lower than in the "show" task, since half of the children in Group I were asked to "say" before they were asked to "show."

The rank order of difficulty with sets in a task was largely the same for both groups, but it was somewhat different for the two tasks. We averaged the percentage of error with each set in each task for both groups (see Table IX-2).

Table IX-2. Rank order of percentage of error with sets for both groups.

Sets	Task		Sets	Errors (%)
	Show	Say		
	Errors (%)			
p t k	24.38	w l r		14.71
w l r	16.58	b d g		13.68
b d g	15.47	p t k		6.96
f s s̥	4.84	f s s̥		6.66

(IX. SPEECH COMMUNICATION)

The rank order of difficulty is somewhat surprising, given previous observations about which sounds are difficult for children to produce during the developmental period.³ The least difficult set to "show" and to "say" is the set of stridents. This set contains speech sounds which children are presumed to distinguish comparatively late, both perceptually and productively. The unvoiced stops were the most difficult to "show," whereas they were next to the least difficult to "say." These sounds are presumably early acquisitions; therefore, the reason for the remarkable difficulty in identifying differences between /p/, /t/, /k/ in both groups is far from clear. It does not appear to be a function of the experimental design or the phonological context (that is, unvoiced stops began and ended the stimulus words), since errors were sharply reduced when the task was "say the word." The fact that both showing and saying the set of liquids was comparatively difficult coincides with observations that the ability to make distinctions among members of this set of sounds is one of the latest to be acquired by children. The comparative difficulty in showing and saying the voiced stops is also surprising.

The overall results indicate that, except for the unvoiced stops, children perform almost equally well in identifying and reproducing members of these speech sound sets when they appear in initial position in words. They do not perform equally well across sets, however, either when identifying or reproducing. There were significant differences between the unvoiced stridents and all other sets in the "show" task and, despite the fact that there were no significant differences between sets in the "say" task, a trend could be observed. The liquids and voiced stops were equally difficult to reproduce, and more difficult to reproduce than the unvoiced stridents and stops. The data are now being analyzed to examine (i) developmental trends because a range in age of over 2 years is represented in the sample population, and (ii) the direction of substitutions in both tasks within sets.

The present study is preliminary. Since particular stimulus words were used which present particular CVC contexts, it may not be possible to generalize the results. It is important to replicate the study with other CVC contexts. A contemplated study of the determination of the speech sound boundaries within these sets for both tasks with the use of synthetic speech stimuli should add information. Such a study has already been carried out with the liquids.⁴

One other observation that can be made from these data is interesting and provocative. The children who were asked to produce the stimulus word at approximately the same time when they were asked to identify it (Group II) did not do better than the children who carried out each task separately (Group I). They did worse. Actually producing the word did not improve the identification performance. This result brings into question some hypotheses about the effect of verbal mediation on performance,⁵ and also the hypothesis that reference to a motor gesture may be used in identification. It is not clear, however, that such mediation or reference interfered with performance, since

the children participating in Experiment 2 were younger than those in Experiment 1. It is another question for further study.

Paula Menyuk

References

1. M. H. Powers, "Functional Disorders of Articulation -- Symptomatology and Etiology," in L. E. Travis (Ed.), Handbook of Speech Pathology (Appleton-Century-Crofts, Inc., New York, 1957).
2. These experiments were carried out with the cooperation of Patricia Looney of the Hearing and Speech Division of the Children's Hospital Medical Center, Boston, Massachusetts. The experimental tasks have been extended to children with hearing losses and language disorders.
3. D. McCarthy, "Research in Language Development: Retrospect and Prospect," Monograph No. 24 of the Society for Research in Child Development, 1959, pp. 3-24.
4. P. Menyuk and S. Anderson, "Children's Identification and Reproduction of /w/, /r/ and /l/," *J. Speech Hearing Res.* 12, 39-52 (1969).
5. J. H. Flavell, D. R. Beach, and J. M. Chinsky, "Spontaneous Verbal Rehearsal in a Memory Task as a Function of Age," *Child Develop.* 37, 283-300 (1966).

B. MITSYN: A GRAPHICAL NOTATION AND SYSTEM FOR THE INTERACTIVE COMPUTER-AIDED SPECIFICATION AND REALIZATION OF MUSIC AND SPEECH SYNTHESIS AND OTHER (DIGITAL) SIGNAL PROCESSING

A graphical notation (programming language) has been designed for the specification of signal processing and synthesis, and a computer-based system has been implemented to record and interpret that notation. The language and interpreter are known as the MITSYN system (Multiple or Musical Interactive Tone SYNthesis). MITSYN provides for both additive and subtractive types of synthesis, and also for the use of natural (external) signals as input for processing or control (a practice known as Musique Concrete). To facilitate and promote user celerity, both the notation and the syntax of the man-machine interaction during notation editing are highly graphical in nature, a form which most humans assimilate more rapidly than character string-oriented notations. MITSYN is used in an interactive manner wherein the user edits the notation, monitors the resultant synthesized or processed signal, and then may return to notation editing and repeat the cycle. Because of its nature as an easily programmed general (digital) signal processor and synthesizer, MITSYN is also useful in fields such as research in speech and psychoacoustics.

1. Notation Components

The specification of signal processing comprises two major components of notation -- "configurations" and "scores" -- and other supplementary components. Configurations

(IX. SPEECH COMMUNICATION)

specify in block diagram format the interconnection of elemental signal-processing primitives or "blocks" such as oscillators and filters; that is, configurations represent processing networks. Scores specify the variation of "externally controlled" parameters of configurations as functions of time. Coupling between scores and configurations is achieved by the use of "score reading" blocks in configurations.

Scores are a time-ordered sequence of data, of which the principal items are "parameter tracks" and "notes."

Parameter tracks are conceptually "continuous" functions of time, and may be edited in much the same way that a single-valued function is drawn on graph paper. Parameter levels are accessed from configurations by PAR blocks which output the specified level as a function of (synthesis) time.

"Notes" are an event or discrete type of notation, as opposed to continuous parameter tracks. Each note has 4 attribute values -- time origin, level, duration or length, and class identifier. Notes are used by configurations through NLEV blocks which output the level of the most recently read note of a given class, and through NDUR blocks which output a Boolean (truthvalue) signal which is TRUE during the scope of a note of a given class and FALSE otherwise.

The inclusion of both a continuous notation and an event notation in a single system is an important feature of the language. For many applications there is need for both types, and a system architecture oriented toward a single type may strongly (albeit perhaps unconsciously) influence users to "think its way." Published examples of related single-type systems are the event or "note-oriented" MUSIC programs of Mathews¹ and the continuous type of notation GROOVE program of Mathews and Moore.² Since parameter tracks and note levels both supply level inputs to a configuration, the selection for a specific use is determined primarily by stylistic criteria and convenience. Traditional "note-oriented" music would be typically instrumented by using note levels for pitch levels, with note durations driving envelope generators which, in turn, control note-amplitude envelopes, and with either note levels or parameter tracks for general dynamic levels. Some contemporary "continuum" musical styles would be more oriented toward parameter tracks. Formant (terminal analog) speech synthesis typically uses parameter tracks for the control of formant frequencies, fundamental frequency, and voicing and frication amplitudes.

There are two types of temporal signals in the system: analog (internal representation is digital) and Boolean (truthvalue). An important feature is the homogeneity of analog signal types, with no distinction between "audio" and "control" signals. External signals, from either score reading blocks or an "audio" input block, are thus completely interchangeable with internally generated signals. For example, the frequency controlling input of an oscillator might come directly from a parameter track or note level in a score, but might also come from the output of another oscillator (which would

then effect frequency modulation).

Another component of notation is a class of arbitrary "functions" over a finite (limited scope one-dimensional domain. Functions are used to specify the shape of each cycle of function (waveshape) repeating oscillators, and are used to specify the shape of the "attack" portion of the output of some types of envelope generators.

The last component of notation is a representation of an analog time signal. These are called "audio files." Configurations generate audio files, and may also use them as input for processing. Audio files may be converted to real-time audio by digital-to-analog conversion (playing), and may be created from real-time audio by analog-to-digital conversion (recording). We use the term "audio" as a general term for analog time signals, although all such signals need not be acoustic, for example, an electroencephalographic signal.

2. Notation Editing

Associated with each of the components is an "editor" that displays members of its class and facilitates their manual creation and modification. Hence, there are configuration, score, function, and audio editors. The act of using the MITSYN system is essentially one of communicating with the various editors. Each member of each class exists as a "file" in a file system, and thus the editing operation consists in either opening an existing file or creating a new one, making the desired additions and/or modifications, and then closing the file.

The manual editors are designed to allow the user to specify editing down to low-level detail if he chooses. Some include additional constraints which may be optionally invoked to expedite editing by imposing more structure or "default conditions." Of course, additional programs or special-purpose editors may also be written in the implementation language to generate MITSYN notation components from higher level notations. An example of this would be a program of terminal analog speech synthesis by rule strategy, which would input phonetic symbols and generate MITSYN scores for a speech-synthesizer configuration.

The MITSYN User's Manual³ defines the "semantics" of the notation and gives details of how to use the editors. In this report we shall present the general features of the notation, and some of the interaction techniques employed by the editors.

Most notation-editing operations are specified graphically by drawing with a tablet pen and observing a display. Figure IX-1 shows a typical computer terminal suitable for use with a MITSYN implementation.

3. Block Semantics and the Configuration Editor

Figures IX-2, IX-3, and IX-4 show displays of the configuration editor when editing an illustrative configuration. The basic operation of creating a configuration is to select

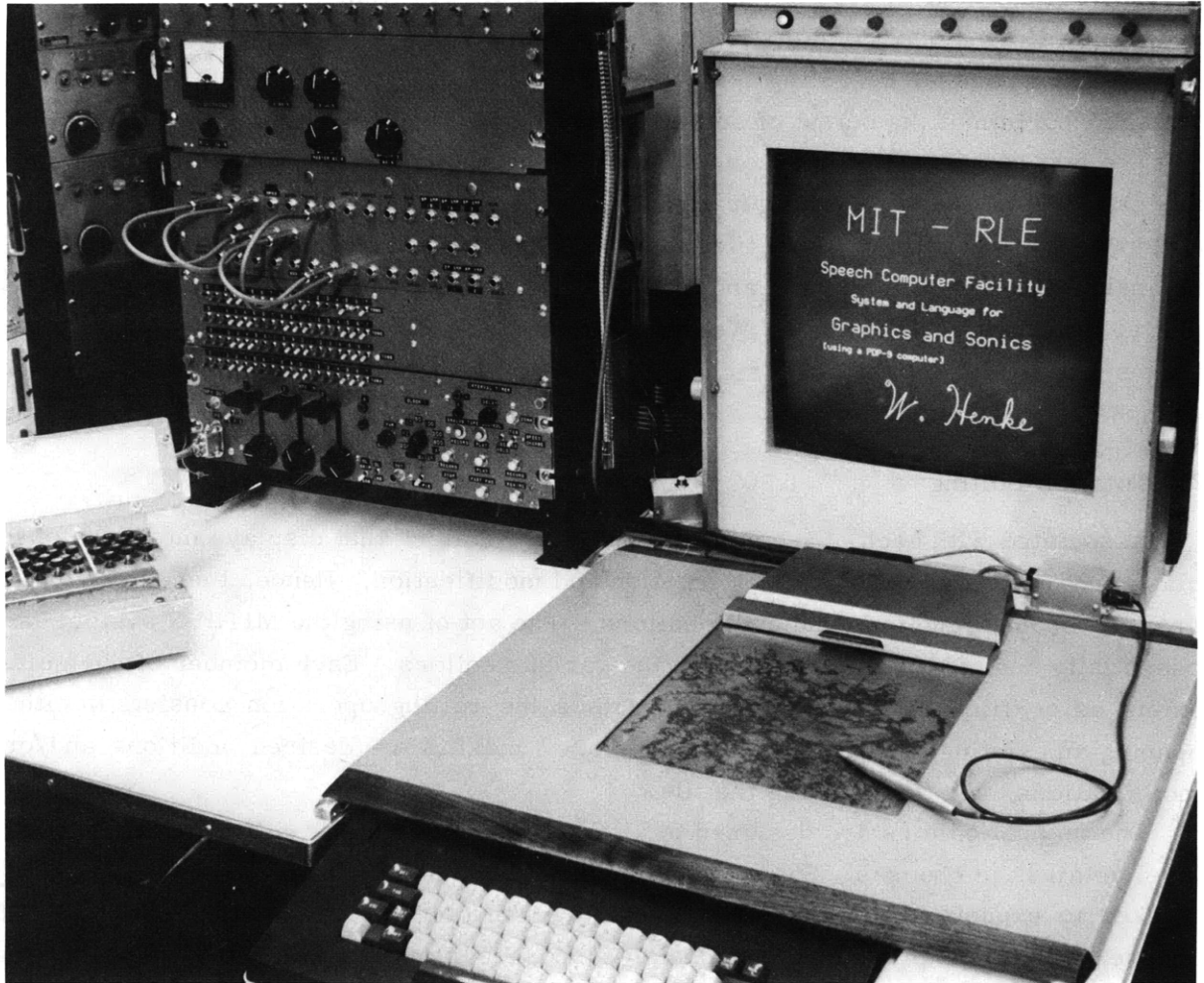


Fig. IX-1. Typical user console for a MITSYN implementation. Immediately below the display screen is a data tablet, on top of which is resting the tablet pen. The pen is the primary input device for both system control and data for the various MITSYN notations. When in use, the pen position on the tablet is usually echoed by one of several forms of a cursor in the display. In front of the tablet is a text keyboard for textual command and data input. To the left is a "function button box" occasionally used for system control, although the most frequently used functions are invoked by selecting the desired option from display "command menus" with the tablet pen. Also shown is an analog signal patch panel to make connections within the studio; speaker and microphone gain controls; and audio tape recorder controls.

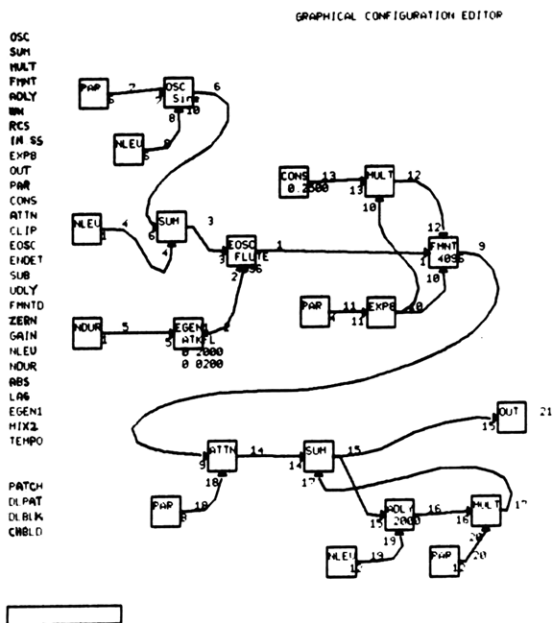


Fig. IX-2.

Display of the configuration (block diagram) editor showing one relatively small illustrative configuration. Along the left edge of the display are "command menus." The upper menu is used for block type selection when a block is to be added to a configuration. The lower menu is used to initiate other editing actions which are (in order): PATCH - interconnect a block output to a block input with a "patch"; DLPAT - delete a patch; DLBLK - delete a block; and CHBLD - change block data.

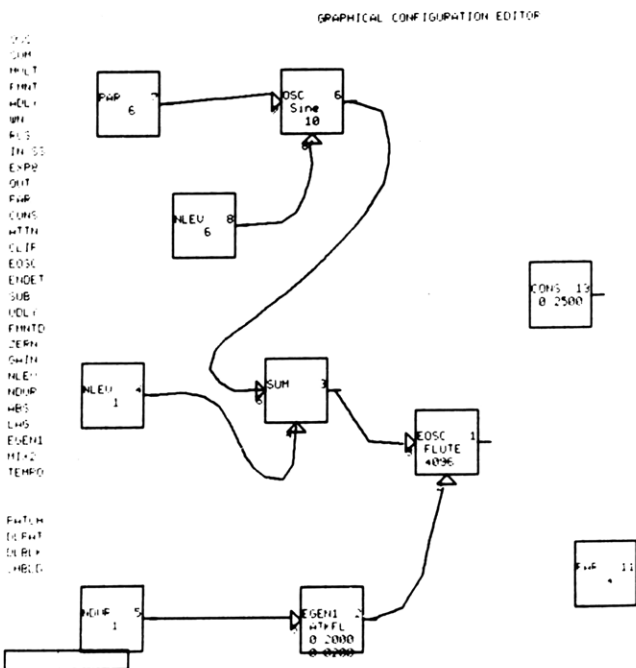


Fig. IX-3.

Display of same configuration as in Fig. IX-2 using a window of smaller scope so that a smaller section of a configuration is enlarged.

(IX. SPEECH COMMUNICATION)

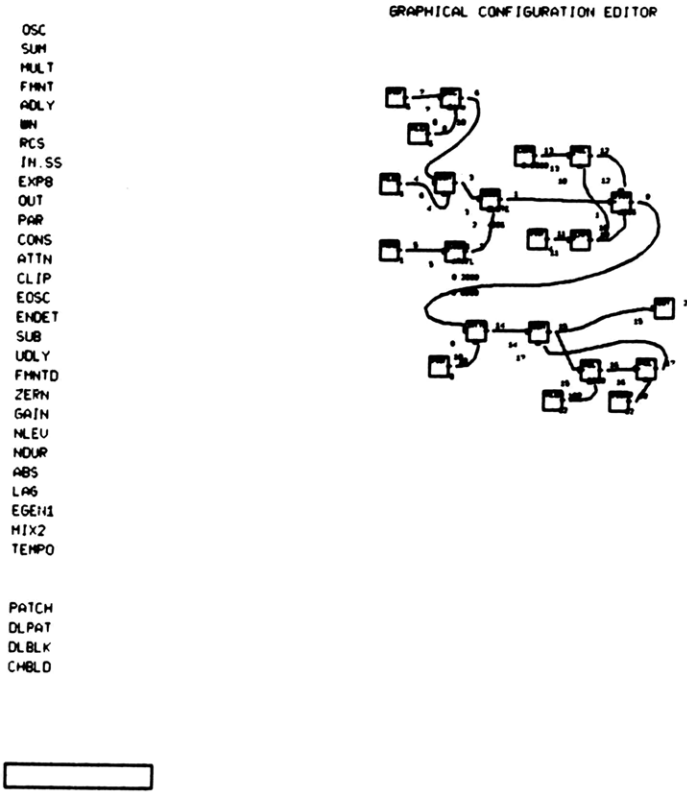


Fig. IX-4. Display of same configuration as in Fig. IX-2 using a window of larger scope showing that all of larger configurations may be simultaneously displayed (with concomitant diminution of size). Such large window scopes are not often used, however, as large configurations are usually edited by moving a smaller scope window around over subsystems.

block types and position them with the pen, and then interconnect them with "patches" by drawing lines from block outputs to block inputs.

The signal-processing primitives or blocks are represented by squares with up to three inputs, one each at the left, bottom, and top edges, and an output at the right edge. Only one patch may be connected to any input, but an output can drive any number of patches. Some block types also require auxiliary constant data such as waveform names or time constants, and these data are displayed below the block type name.

We shall use the particular configuration displayed in Fig. IX-2 to illustrate the characteristics of some of the block types, and to show how they are assembled into subsystems and a total configuration. For purposes of exposition we shall identify specific blocks using the block type name and the system-supplied integer block identifier which is displayed at the right of each block type name.

In the upper left corner there is a grouping of blocks serving as a frequency modulated source. The signal oscillator (EOSC 1) has exponential frequency dependence,

playing (repeating at the desired frequency a function of name "FLUTE" with an upper reference frequency of 4096 Hz. The amplitude of the oscillator (bottom input) is controlled by the output of an envelope generator (EGEN 1,2) using for the attack a waveform of name "ATKFL" with temporal scaling attribute of 0.2 second, and a decay time constant of 0.02 second. The envelope generator is controlled from the duration of score notes (NDUR 5) of class "1" ("1-notes"). The frequency of the oscillator (left input) is controlled by the sum (SUM 3) of "1-note" levels (NLEV 4), which are used to denote the center frequency, and the output of an oscillator (OSC 6) with a 10-Hz reference frequency playing a function of name "Sine." The frequency of modulation is thus denoted by parameter track 6 (PAR 7), and the depth of modulation with the level of "6-notes" (NLEV 8).

The output of the FM oscillator then passes through a formant filter (complex pole pair) (FMNT 9) to realize varying timbre control. The center frequency of the filter (bottom input) is driven from an exponentiated (EXP8 10) parameter track 4 (PAR 11). The bandwidth (top input) is driven from a constant (CONS 13) fraction (MULT 12) of the center frequency control, and thus this network realizes a constant Q filter.

The output of the formant filter is then attenuated (ATTN 14) by parameter track 8 (PAR 18) for dynamic level control.

The attenuation is followed by a variable reverberation (recursive) network where the feedback signal (bottom input of SUM 17) is delayed (ADLY 16) by a duration denoted by the level of "12-notes" (NLEV 19). The amount of reverberation is controlled by multiplication (MULT 17) with the level of parameter track 12 (PAR 20).

A brief description of some of the other available block types and their characteristics follows.

WN – white noise source.

RCS – random control source.

IN.SS – input from signal store, used when a previously recorded or synthesized signal is to be used as input to a configuration.

CLIP – clipper with controllable clip levels.

ENDET – envelope detector.

SUB – subtracter, an adder with polarity inversion of one input.

UDLY – unit delay, useful for building digital filter types that are not currently provided as primitives.

FMNTD – formant filter, denominator of transfer function only (i. e., not gain-normalized).

ZERN – complex pair of zeros, numerator of transfer function only.

ABS – absolute value.

LAG – a first-order recursive filter with modifications to eliminate the deadband effect.

It has an impulse response that is exponential for large input-output signal

(IX. SPEECH COMMUNICATION)

differences and straight line for smaller differences. It is often used as a smoothing filter for "control" types of signals.

MIX2 – adjustable ratio-controlled two-signal input mixer. It is useful for mixing oscillator outputs for intranote timbre variation, and for mixing envelope generator outputs for internote attack variation.

TEMPO – can be used to adjust the time scale of a score to control the tempo. It does not affect nonscore temporal attributes such as attack rates and filter time constants. For types of synthesis which are to have a metrical temporal structure, scoring is more convenient if round units of time are used in conjunction with subsequent rate adjustment to the desired rate with a TEMPO block.

4. Score Editor

Figure IX-5 is a display of the score editor showing a section of a score that includes several notes and 4 parameter tracks. Parameter tracks are represented as a sequence of dots, and notes are represented by graphic notation that depicts the four attribute values of each note: time origin, duration, level, and note class (integer).

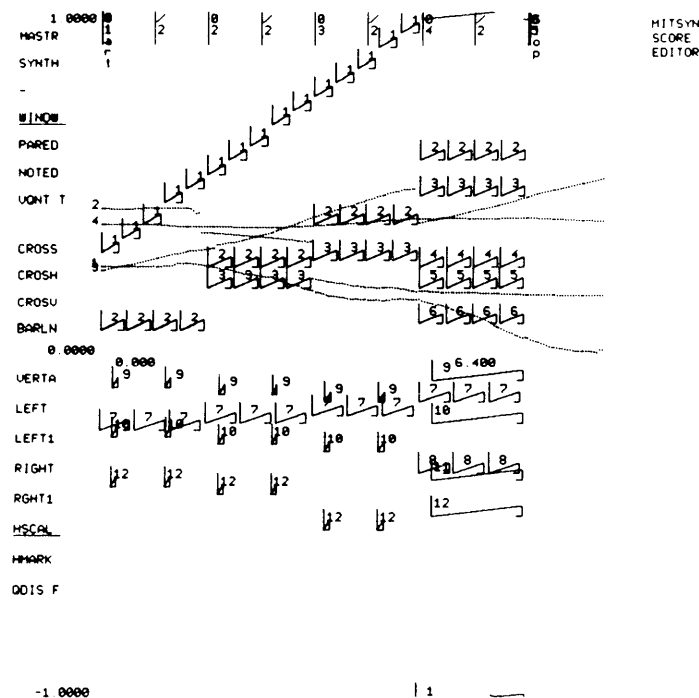


Fig. IX-5. Display of the score editor showing a section of a score and some of the editing menus. The window scale here is quite small so that a large scope of the score is presented, although the concomitant denseness makes observation of individual detail difficult. For editing, the window scale is normally increased so as to enlarge a smaller scope of a score.

Along the left edge of the display there are command menus. The upper "master" menu selects one of three major editor modes, window scope adjustment, parameter edit, and note edit, and also flips a vertical input quantization toggle (VQNT). Each of the three modes has its own "mode menu" at the lower left; the window adjustment menu is shown in Fig. IX-5. A typical editing action involves selecting (with the pen) some command from a menu, and then drawing a line on the figure or positioning a cursor with the pen.

Several other graphics and cursors may be added to the display at appropriate times to aid in editing. These include quantization levels, bar lines, bar subdivisions, and a cross-hair cursor. There are also optional "pen strategy" editing aids that may be invoked when desired. Examples include level quantization, a straight-line mode when drawing parameter tracks, and "proximity" note deletion and note attribute value adoption during note editing.

5. Function Editor

The shapes of particular "functions" (single-valued functions over a limited scope) can be specified with the function editor. Varieties of some built-in "analytic" shapes may be requested. Weighted sums of harmonically related sinusoids are often used by oscillators that are to be played at audio frequencies. The weighting of the harmonics determines the timbre. Square, sawtooth, and triangular waves may also be requested. In addition, the user may make freehand modifications to any wave shape by using the tablet pen. Straight-line and freehand-drawn wave shapes are often used for attack shapes; however, since they are usually not bandlimited (in the Fourier sense), their use with oscillators tends to generate "aliasing" or "foldover" noise, which may not be desired.

6. Audio Editor

The audio editor has facilities for audio transput (playing and recording) which permit conversion between analog signals and their digital representations at a "sampling rate." Analog signals should normally be lowpass-filtered at one-half the sampling rate to prevent conversion aliasing. The audio editor also includes (digitized) analog signal time and frequency-domain display and analysis features. These analysis features have another use: the techniques of observing a signal at various locations in a configuration and of comparing synthesized signals with signals from other sources are often helpful when we attempt to design specific characteristics into a signal.

7. Implementation

The present MITSYN implementation is written in an extended FORTRAN IV programming language. The extensions are primarily in the directions of graphics, dynamic

(IX. SPEECH COMMUNICATION)

storage allocation, list processing, and sonics (audio processing and transport [I/O]). MITSYN operates within the environment of a file-oriented Disk Operating System, and is segmented into a system of overlays for use with limited main-memory computers. MITSYN is implemented for use on the modest graphics and sonics facilities of the Research Laboratory of Electronics Speech Communication Group's computer system⁴ whose central processor is a DEC PDP-9, a small-sized computer. This implementation of MITSYN is an example of how a fairly sophisticated level of interactive computer graphics, programmed in a higher level language, can be realized on a computer system that is smaller than those which at present are normally associated with such endeavors. The implementation has been designed to be relatively machine-independent, and thus can be easily transported to other systems. [Another implementation is being planned at Ohio State University on a larger system (DEC PDP-10).]

The synthesizer segment processes a digital representation of an audio signal. Inputs are notation files created by the editor segments of the system. Synthesis is realized by a compilation phase, followed by an execution phase. A configuration file is first read and "compiled" into a signal-flow network. [A synthesizer configuration is conceptually a parallel signal processor, that is, all blocks run concurrently. For execution on a sequential machine (i. e., a contemporary general-purpose digital computer) an internal representation must be created to allow the blocks to be executed serially. The sorting of all signal-processing blocks into an appropriate execution order is a primary facet of the compilation phase.] After the signal-processing network is built, those waveform files which are requested by blocks such as oscillators and envelope generators are accessed. Then, following such compilation of a synthesizer configuration, signal-processing execution begins. The score file is read sequentially and the indicated signal-processing operations are effected.

The present host computer is an 18-bit word-length machine without floating-point arithmetic hardware. In order to achieve a reasonable synthesis time scale (synthesis compute time)/(signal real time), processing must be done by using single-word fractional fixed-point arithmetic. Synthesis speed is naturally a function of the complexity of the synthesizer configuration. For the present implementation, a simple configuration (two oscillators and one summer) runs with a time scale not much greater than unity. For more complex configurations the time scale may become significantly greater. Speed could be increased by several times on a (still small) processor with a modern instruction repertoire, and much more by using a more powerful processor.

The short word length somewhat limits the dynamic range, and the signal-to-noise ratio may deteriorate for complex configurations which require that signals pass through many blocks between their point of origin and eventual output. To minimize such deleterious effects, signal levels should be kept as high as possible throughout the system without incurring overflow (a signal exceeding unity magnitude). The first overflow that

occurs during a synthesis run is reported with the block identifier showing where it happened. If this occurs, signal levels in the vicinity of the offending block should be reduced by modifying the configuration and/or inputs to a level just under that which causes overflow. The only synthesizer processing blocks in which overflow can occur are addition-type blocks and filters.

One can easily envision achieving real-time synthesis by doing the final signal processing in hardware, either analog (for example, controlling Moog or Buchla modules) or digital (which could be designed to be much more flexible and would be inherently more stable). The cost of digital hardware continues to decrease rapidly, which suggests that a digital hardware synthesizer will soon be economically feasible. At the present time we are making a modest effort to design and construct such digital hardware. With a real-time synthesizer additional real-time (at "performance" time) control inputs would become possible. The distinction between notation inputs and real-time inputs would be somewhat analogous to the distinction between a composer and a performer or conductor.

Richard E. Albright's thesis research⁵ contributed significantly to the evolution of MITSYN. Many conversations and work sessions with Robert P. Ceely, a composer who has been MITSYN's most demanding user,⁶ have also stimulated the work.

W. L. Henke

References

1. M. V. Mathews, The Technology of Computer Music (The M. I. T. Press, Cambridge, Mass., 1969).
2. M. V. Mathews and F. R. Moore, "GROOVE - A Program to Compose, Store, and Edit Functions of Time," *Communs. ACM* 13, 715-721 (1970).
3. W. L. Henke, MITSYN User's Manual (unpublished, available from the author, Research Laboratory of Electronics, M. I. T., Cambridge, Mass. 02139).
4. W. L. Henke, "An Interactive Computer Graphic and Audio System Sans Large Budgets and Great Fuss," *Quarterly Progress Report No. 98*, Research Laboratory of Electronics, M. I. T., July 15, 1970, pp. 126-133.
5. R. E. Albright, "Methods of Tone Synthesis in Computer Music," S. M. Thesis, Department of Electrical Engineering, M. I. T., September 1970.
6. R. P. Ceely, "A Composer's View of MITSYN," Preprint No. 811 (M-1), Audio Engineering Society Convention, October 1971.

C. SPEECH ANALYSIS BY LINEAR PREDICTION

1. Introduction

This report describes the development of a speech analysis system based on linear prediction of the speech wave. The analysis is achieved by representing the speech wave

(IX. SPEECH COMMUNICATION)

in terms of a set of parameters closely related to the glottal excitation function and the vocal-tract transfer function.

The system has been implemented by utilizing the computer facilities of Group 24 at Lincoln Laboratory, M. I. T. These facilities include the Univac 1219 computer, which is a medium-sized general-purpose computer; the Fast Digital Processor, which is a fast programmable signal processor attached to the Univac 1219; and peripheries, such as A/D and D/A converters and various display facilities. The system is capable of performing real-time spectrum analysis when both spectral cross-section and spectrographic displays are possible. Effort is now directed toward evaluation of its performance in extracting such acoustic parameters as formants and fundamental frequency of voicing. An initial attempt at formant tracking on spectra derived from linear prediction has given promising results.

We shall review briefly the theory of linear prediction, describe the implemented system, and give some preliminary results of speech analysis using this system.

2. Theory

Detailed treatments of the theory of linear prediction and its variations have been reported.¹⁻⁴ Our analysis is based on the speech-production model shown in Fig. IX-6. The all-pole digital filter $H(z)$ represents the combined effect of the glottal source, the

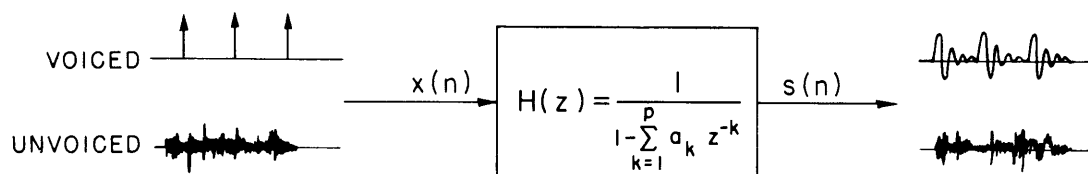


Fig. IX-6. Model of speech production.

vocal tract, and radiation losses. In this idealized model the filter is excited either by a periodic impulse train for voiced speech or random noise for unvoiced speech.

The speech production model can be equivalently characterized by the difference equation

$$s(n) = \sum_{k=1}^p a_k s(n-k) + x(n), \quad (1)$$

where $s(n)$ and $x(n)$ are the n^{th} samples of the output speech wave and the input, respectively. The a_k 's are the coefficients characterizing the filter $H(z)$, and henceforth will be referred to as the predictive coefficients.

From Eq. 1 it is clear that we can determine the a_k 's if we know the input and $2p$ consecutive values of $s(n)$. The first p of these values serves as initial conditions. We shall restrict ourselves here to voiced speech in which the input is a periodic impulse train. In this case the a_k 's can be determined with knowledge of only $2p$ consecutive values of $s(n)$ and the position of the impulses. For this idealized model, we can define the predicted value of $s(n)$ as

$$\hat{s}(n) = \sum_{k=1}^p a_k s(n-k). \quad (2)$$

The difference between $s(n)$ and $\hat{s}(n)$ will be zero except for one sample at the beginning of each period.

In reality, however, $s(n)$ is not produced by this highly idealized model and therefore prediction of $s(n)$ based on Eq. 2 will introduce error. If we are to approximate $s(n)$ by $\hat{s}(n)$ as defined by Eq. 2, the a_k 's can only be determined with the specification of an error criterion.

We can choose to determine the a_k 's by minimizing the mean-squared difference between $s(n)$ and $\hat{s}(n)$, that is, by minimizing

$$E = \sum_{n=0}^{N-1} [s(n) - \hat{s}(n)]^2. \quad (3)$$

Note that the squared difference is summed over all samples except one at the beginning of each period, and we have assumed that the minimization is to be carried out over a section of $s(n)$ of length N . It is also important to note that p more values of $s(n)$ are needed for proper boundary conditions.

The minimum mean-squared error criterion is chosen instead of other error criteria because the determination of the a_k 's now reduces to the solution of the following set of linear equations.

$$\sum_{k=1}^p a_k \phi_{jk} = \phi_{j0} \quad j = 1, 2, 3, \dots, p, \quad (4)$$

where

$$\phi_{jk} = \sum_{n=0}^{N-1} s(n-j) s(n-k) \quad k = 0, 1, 2, \dots, p. \quad (5)$$

Note that the sum in Eq. 5 excludes one point at the beginning of each period.

Equation 4 can be written in matrix form as

(IX. SPEECH COMMUNICATION)

$$\Phi \underline{a} = \underline{\psi}, \tag{6}$$

where Φ is a $p \times p$ matrix with typical element ϕ_{jk} ; \underline{a} and $\underline{\psi}$ are p -dimensional vectors with the j^{th} component given by a_j and ψ_j , respectively. The solution of this matrix equation is greatly simplified by the fact that the matrix is symmetric and hence recursive procedures are applicable.

It is of interest to compare the analysis procedure outlined above for two different cases. If the fundamental frequency of voicing is known in advance, the analysis can be carried out directly, in the sense that Eq. 5 can be evaluated exactly. In practice, however, it is highly desirable to carry out the analysis without a priori knowledge of pitch. In this case an approximation has to be made and additional error is introduced. We shall illustrate this point by a simple example, but the argument can easily be generalized to include more complicated situations.

Let us assume that there is only one pitch pulse in the data and it occurs at $n = m$. If m is known, then Eq. 5 can be evaluated as

$$\phi_{jk} = \sum_{\substack{n=0 \\ n \neq m}}^{N-1} s(n-j) s(n-k). \tag{7}$$

Equation 5 can not be evaluated explicitly, however, if m is unknown.

Let us now approximate ϕ_{jk} by

$$\hat{\phi}_{jk} = \sum_{n=0}^{N-1} s(n-j) s(n-k). \tag{8}$$

Comparing Eqs. 7 and 8, we find that the error in ϕ_{jk} is given by

$$\epsilon_{jk} = \hat{\phi}_{jk} - \phi_{jk} = s(m-j) s(m-k). \tag{9}$$

By the nature of the speech wave, $s(m-j)$ and $s(m-k)$ are small compared with samples at the beginning of each period. Therefore the error ϵ_{jk} is small compared with ϕ_{jk} for any reasonable N . Results of comparing the two analysis procedures will be presented.

The theory of linear prediction has also been formulated in a slightly different way.^{3, 4} Let $e(n)$ denote the output of the inverse filter $H^{-1}(z)$ when it is excited by $s(n)$. If we choose to determine the a_k 's by minimizing the total energy in $e(n)$, the set of equations obtained can be shown to be almost identical to Eqs. 4 and 5. The only difference between the two formulations is that, since $e(n)$ is of length $N+p$, the matrix Φ in the second formulation is of Toeplitz form,

$$\phi_{jk} = \phi_{|j-k|, 0} \tag{10}$$

Although the theory developed thus far is for voiced speech, we have used the same procedure to determine the predictive coefficients for unvoiced speech.

3. Speech System

Figure IX-7 is a block diagram of the analysis system. At present, only that part of the system enclosed in the dashed lines has been implemented. Input data are first pre-emphasized (10 dB/octave), bandlimited to 5 kHz, and sampled at 10 kHz. The computation of the $\hat{\phi}_{jk}$, as defined by Eq. 8, can be greatly reduced by noting that

$$\hat{\phi}_{j+1, k+1} = \hat{\phi}_{jk} + s(-1-i) s(-1-j) - s(N-1-i) s(N-1-j). \quad (11)$$

Therefore only $\hat{\phi}_{j0}$ for $j = 0, 1, 2, \dots, p$ need be computed directly. These are the first $p+1$ points of the short-time autocorrelation function of $s(n)$. The rest of the matrix elements are obtained recursively from Eq. 11. The last two terms on the right-hand side of Eq. 11 can vanish to result in a Toeplitz matrix, depending on how the problem is formulated. After the elements of the matrix are formed, Eq. 6 is solved by the method of square-rooting.⁵

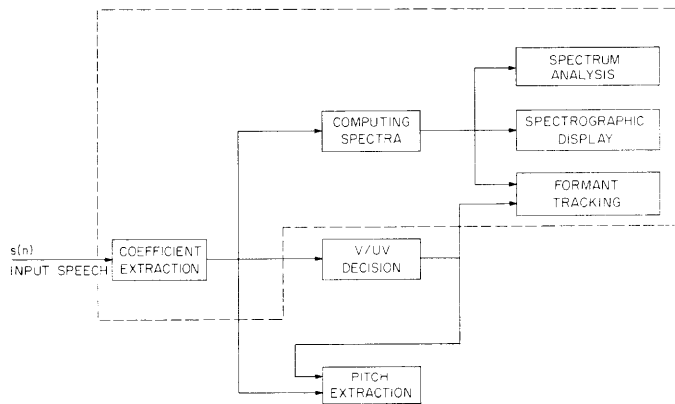


Fig. IX-7. Analysis system.

From the predictive coefficients, the approximated spectral envelope of $s(n)$ can then be computed as $|H(e^{j\omega})|$. Note that the unit-sample response of the inverse filter $H^{-1}(z)$ is given by

$$\hat{h}(n) = \begin{cases} 1 & \text{for } n = 0 \\ a_n & \text{for } n = 1, 2, 3, \dots, p \\ 0 & \text{otherwise} \end{cases}$$

(IX. SPEECH COMMUNICATION)

Therefore $\hat{|H(e^{j\omega})|}$ can be obtained efficiently by computing the discrete Fourier transform of $\hat{h}(n)$ with a fast Fourier transform algorithm, and then inverting the result. Each spectral cross section is multiplied by the rms value of the input data to provide gain normalization.

Both the input data length N and the order of the filter p are variables; the choice of these variables has been discussed elsewhere.^{1, 3} Unless otherwise specified, all results presented are obtained with $n = 256$ and $p = 12$. The coefficients are recomputed every 6.4 ms.

4. Preliminary Results

In Fig. IX-8 spectra of a synthetic vowel /a/ obtained by using various techniques are compared: (a) and (b) by windowing (with different window widths) and Fourier-transforming the waveform, (c) by cepstral smoothing,⁶ and (d) by linear prediction. In Fig. IX-8a the effect of glottal periodicities can be seen as the ripples superimposed on the spectral envelope. These ripples are greatly reduced in Fig. IX-8b because of spectral smearing of the wider frequency window. In Fig. IX-8c the effect of glottal periodicities is removed by a homomorphic technique. This effect is also removed in Fig. IX-8d. But, since the analysis is based on a specific model and thus limits the number of spectral peaks, there are no extraneous peaks in Fig. IX-8d. If we compare the locations of the spectral peaks with the actual values of the five formants, it is clear that, for this example, the spectrum derived from linear prediction provides accurate formant locations.

Figure IX-9 shows the spectrum of the same vowel obtained by linear prediction, except that in this case the analysis is carried out pitch-synchronously. Comparing Figs. IX-8d and IX-9, except for the bandwidth of the second spectral peak, we find that the qualitative difference between the two spectra is quite small.

It should be noted that we have chosen to use a lot of synthetic speech material in our study. This is because parameters of synthetic speech are known exactly. Therefore the use of synthetic speech can provide us with a more objective evaluation of the analysis system.

Figure IX-10 is a spectrographic display of a sentence generated from a synthesis-by-rule program developed by D. H. Klatt. Some observations can be made concerning Fig. IX-10. First of all, the smooth and continuous formant trajectories are clearly visible for all non-nasal sonorants. Second, the analysis is able to separate closely spaced formants very well, as in the case of /r/. The analysis also worked well for fricatives, nasals, and stops, in the sense that spectra obtained during frication, nasalization, and aspiration contain the important features characterizing these phonemes. For example, spectra derived from linear prediction for nasals all have a low-frequency peak, followed by a relative absence of energy in the 500 ~ 1500 Hz

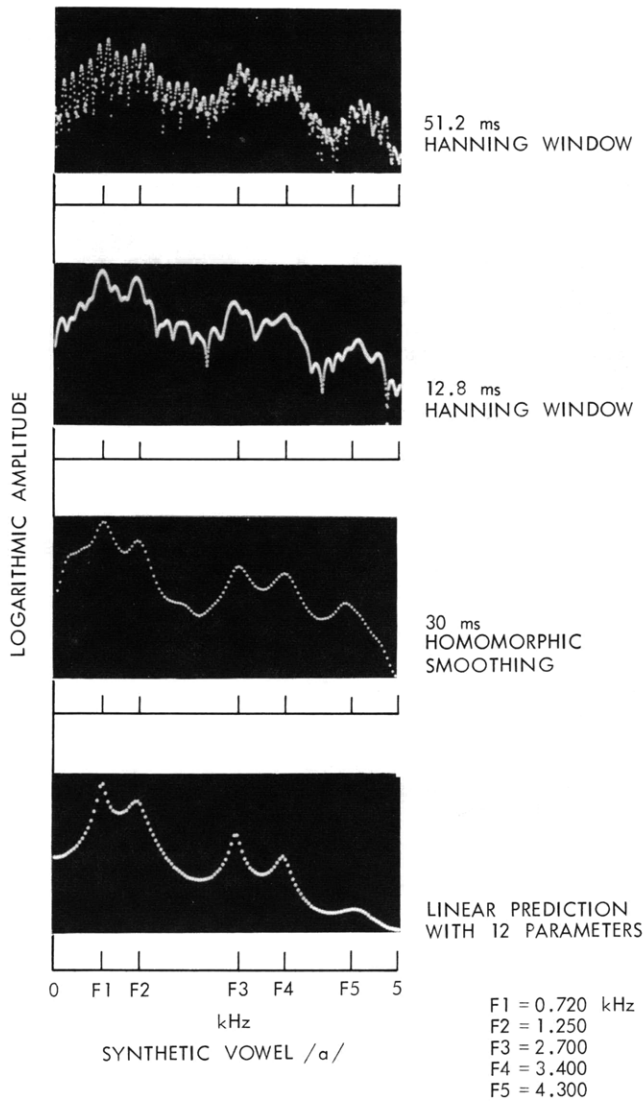


Fig. IX-8.

Spectra of synthetic /a/: (a) by windowing (51.2 ms) and Fourier transforming, (b) by windowing (12.8 ms) and Fourier transforming, (c) by cepstral smoothing, (d) by linear prediction (N=128).

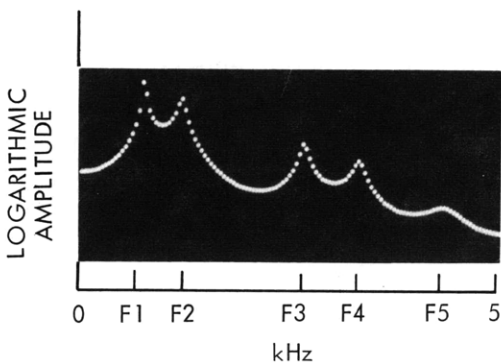


Fig. IX-9.

Spectrum of synthetic /a/ from pitch-synchronous linear prediction analysis (N=128).

(IX. SPEECH COMMUNICATION)

region. These are some of the important spectral attributes of the nasals.

Figure IX-11 is a spectrographic display of a sentence spoken by a male subject and has features similar to those discussed above.

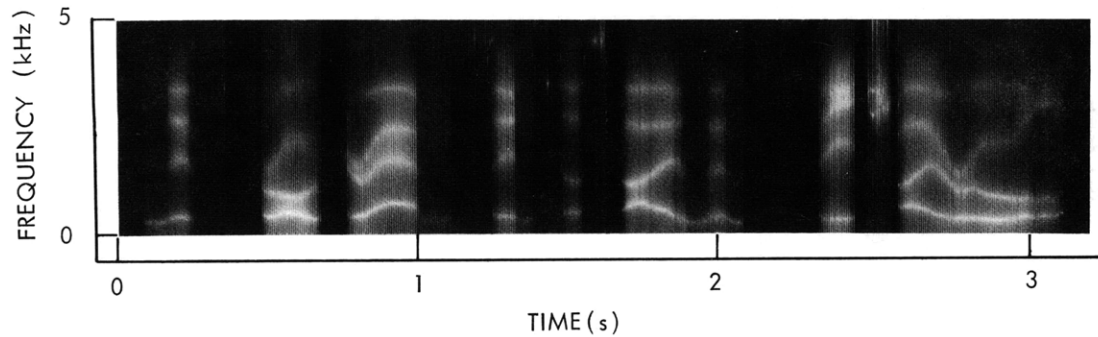


Fig. IX-10. Spectrographic display of the sentence "This program synthesizes speech by rule." (Synthetic speech.)

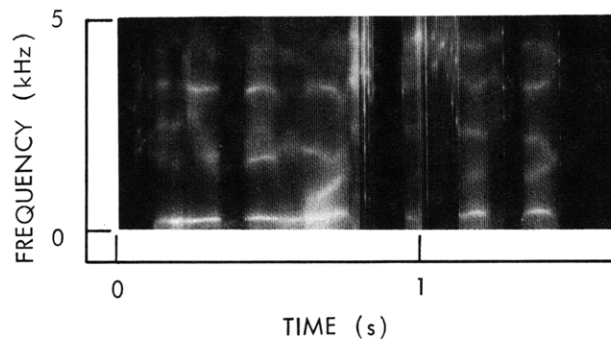


Fig. IX-11. Spectrographic display of the sentence "Can you be more specific?" spoken by a male subject.

From Figs. IX-10 and IX-11 it is clear that during the voiced portion of speech the formants are sharply defined and their trajectories are smooth and continuous. It is therefore reasonable to expect formant tracking by a simple peak-picking algorithm to give good results. Although results of this are not included in this report because a voiced-unvoiced decision has not yet been implemented, formant tracking by a simple peak-picking algorithm worked well in a few examples that were tried.

The system provides highly interactive analysis and display and is capable of real-time processing. Figure IX-12 is another example to illustrate the highly interactive display capabilities of this system. The sentence is spoken by a male subject. By setting the pointer to a specific instant of time on the spectrographic display, we can display and examine the next twelve cross sections on the other oscilloscope.

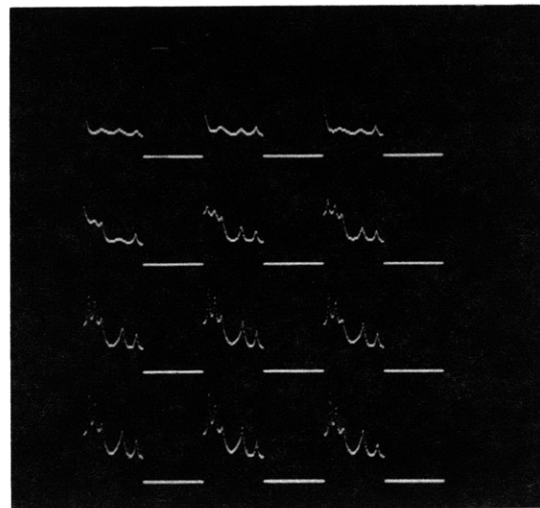
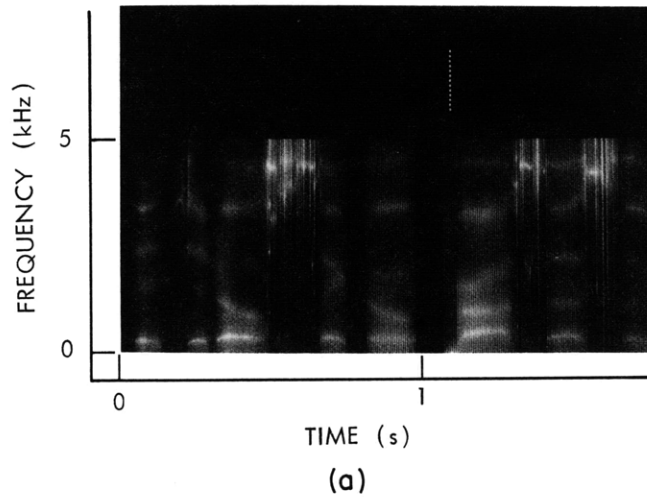


Fig. IX-12. (a) Spectrographic display of the phrase "Digital signal processing" spoken by a male subject; (b) 12 cross sections starting from the pointer in (a).

5. Summary

We have partially implemented a speech-analysis system based on linear prediction of the speech wave. The analysis technique differs from all other techniques, in that it is closely tied to a speech-production model. Our limited experience with the system indicates that it is well suited to spectrum analysis and is potentially very useful for formant tracking. The voiced-unvoiced decision and fundamental-frequency extraction parts of the system are now being implemented.

(IX. SPEECH COMMUNICATION)

The fact that the analysis is based on a specific speech production model also imposes limitations on the technique. It is well known that during the production of nasals and fricatives there exist zeros as well as poles in the vocal-tract transfer function. It can be argued that we can always approximate these zeros by multiple poles and that the important features characterizing these phonemes are generally contained in the overall shape, not in the specific pole-zero locations, of the spectrum. There are other unsettled issues, such as whether the input speech should be windowed, which of the two formulations should be chosen for actual implementation, and so forth. We are now evaluating the system with synthetic-speech material, with all parameters such as formants and bandwidths known exactly. We believe that this evaluation, together with speech synthesis based on linear prediction, will help us resolve some of these issues.

We hope that this system can serve as the acoustic parameter extraction stage of a speech-recognition system. Although it is premature to speculate on its performance for acoustic parameter extraction, the highly interactive analysis and display facilities now developed have proved to be useful in studying the spectral characteristics of phonemes and the spectral changes from coarticulations.

Programming consultation with Mrs. Stephanie McCandless is gratefully acknowledged.

V. W. Zue

References

1. B. S. Atal and Suzanne L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.* 50, 637-665 (1971).
2. C. J. Weinstein and A. V. Oppenheim, "Predictive Coding in a Homomorphic Vocoder," *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-19, pp. 243-248, 1971.
3. J. Markel, "Formant Trajectory Estimation from a Linear Least-Squares Inverse Filter Formulation," Monograph No. 7, Speech Communications Research Laboratory, Inc., Santa Barbara, California, 1971.
4. John Makhoul, "Aspects of Linear Prediction in the Spectral Analysis of Speech," paper to be presented at IEEE-AFCRL 1972 International Conference on Speech Communication and Processing, Boston, Mass., April 24-26, 1972.
5. V. N. Faddeeva, "Computational Methods of Linear Algebra," English Translation by C. D. Benster (Dover Publications, Inc., New York, 1959), pp. 81-85.
6. A. V. Oppenheim, "Speech Analysis-Synthesis System Based on Homomorphic Filtering," *J. Acoust. Soc. Am.* 45, 459-462 (1969).