THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 50, No. 5, May-June, 1971
Printed in U.S.A.

Computer Synthesis of Speech by Concatenation of Formant-Coded Words

By L. R. RABINER, R. W. SCHAFER and J. L. FLANAGAN

(Manuscript received January 13, 1971)

Speech signals can be described in terms of the resonances of the vocal tract. These resonances, or formants, change at rates comparable to the motions of the vocal tract. They therefore can be sampled and quantized to low bit-rates, and hence constitute an economical form for digital storage of speech information. Formant coding also permits flexible arrangement of speech elements into various contexts. This report describes a computer technique for synthesizing continuous messages by concatenating formant data for word-length utterances. The stored data for the synthesis corresponds to a bit-rate of 533 b/s. A Honeywell DDP-516 computer is used to experimentally evaluate a voice response system. In an initial application, the system is used to synthesize 7-digit telephone numbers. To assess the synthesis an interactive dialing experiment, also conducted by the computer, is described. The results show the synthesized numbers to be comparable in communicative effectiveness to naturally spoken digits.

I. INTRODUCTION

If computers could speak with sophisticated vocabularies they could provide a variety of automatic information services. Machines could be interrogated from conventional *Touch-Tone*® telephones and stored data could be accessed by voice.

Naturally spoken speech messages can of course be prerecorded and stored. However, the digital storage required for sizeable amounts of natural speech is inordinate. Further, elements of natural speech in one context cannot be realistically assembled into a different message. With individual pieces of the signal waveform there is no practical way of making natural transitions from one element to the next. In certain messages of highly limited context—notably the Automatic Intercept System—individual words are adequately abutted by having

more than one spoken version of each word. In general, however, sentence-length material cannot be satisfactorily produced in this manner.

For answer-back purposes, requiring sizeable vocabularies, an efficient means of storing and accessing speech information is required. This requirement implies low bit-rate representation of vocabulary elements *and* a flexible means for assembling the vocabulary elements into any message specified by the answer-back program. Toward this requirement, we have devised a synthesis method based upon formant-coded vocabulary elements.

Formants are the resonances, or eigenfrequencies, of the vocal tract. They change at relatively slow rates, comparable in speed to the articulatory motions. Their variations with time can consequently be sampled and quantized to low bit-rates. Furthermore, this description of the speech signal permits separation of information about vocal-tract excitation (i.e., voiced/unvoiced distinctions and voice pitch) from the resonance information. The formant description therefore provides a flexible means for smoothly assembling vocabulary elements into connected speech.

Toward this goal of low bit-rate storage and flexible assembly of computer speech, we have implemented and experimentally evaluated a formant-synthesis answer-back system. In the subsequent discussion we outline principles of the implementation and offer results of an initial application to the synthesis of telephone numbers.

II. SYNTHESIS MODEL

The model for formant synthesis of speech is shown in Fig. 1. The voiced sounds of speech (i.e., those generated by vocal-cord vibration) are produced by the upper branch of the system. An impulse generator produces a sequence of impulses whose spacing is controlled by the "pitch period" parameter, P, which corresponds to the period of vocal cord vibration. This impulse train is modulated in amplitude by a control parameter, A_V, which represents the intensity of voiced sounds. The resulting signal excites a time-varying digital filter composed of four cascaded resonators. Three of the resonators have time-varying resonant frequencies (F_1, F_2, F_3)—which correspond to the first three resonances, or formants, of the vocal tract. The output of this system is passed to a fixed, second-order digital filter which approximates the source spectrum and mouth radiation characteristics of human speech.[1]

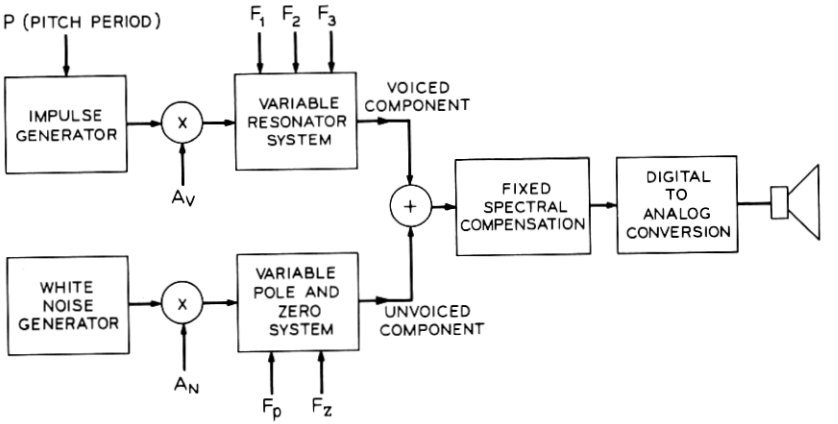

Fig. 1—Digital formant synthesizer, block diagram.

Unvoiced speech is produced by the lower branch of the system in Fig. 1. A random number generator, representative of the fricative noise source in unvoiced speech, produces samples of uniformly distributed white noise. The noise amplitude is modulated by the control parameter, A_N, which represents the intensity of unvoiced sounds. This signal excites another time-varying digital filter composed of one time-varying resonator (F_p) and one time-varying antiresonator (F_z). This pole-zero pair constitute an approximation to the formant structure of unvoiced speech sounds.[1] The output of this system also is passed to the fixed spectral-shaping filter. Digital-to-analog conversion provides an audible output.

All the parameters required by the synthesis system of Fig. 1 can be estimated automatically from natural speech by recently developed digital signal processing techniques.[2,3]

III. CONCATENATION MODEL

The Acoustics Research DDP-516 computer facility has been used to implement a complete answer-back system. A block diagram of the system used for synthesis of connected speech from a vocabulary of formant-coded words is shown in Fig. 2. Naturally spoken, isolated words (or phrases) are analyzed by a formant analyzer to give three formants (F_1, F_2, F_3) voiced and unvoiced amplitude (A_V, A_N), pitch period (P), and unvoiced pole and zero (F_p, F_z) once every 10 ms. These control parameters are smoothed by programmed digital

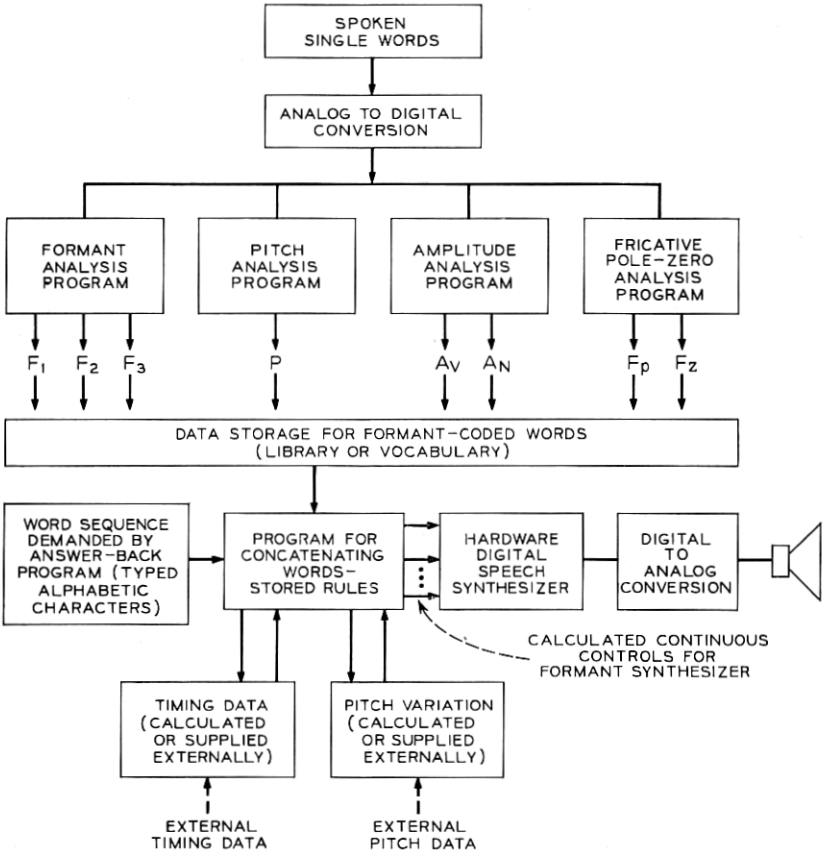

Fig. 2—Overall synthesis system, block diagram.

filters, sampled at their Nyquist rates (typically 33-1/3 s^{-1}), quantized, and stored in the word catalog as the reference library. The typical bit-rate used for storage of these data is 700 b/s when the pitch signal is saved. When pitch is not saved (the usual situation since it is normally calculated by the concatenation program) the bit-rate for the stored data is 533 b/s. Table I shows a breakdown of how these bit-rates are achieved. The data in this table were derived from experimental investigation of the effects of smoothing and quantization on the perception of the synthetic output.[4]

As shown in Table I, at every 10-ms interval the speech is classified as voiced or unvoiced (V/U) by a 1-bit signal. Thus, for each

frame, storage is required for either voiced parameters or for unvoiced parameters, but not for both. It should be noted that the control parameter frame rate (33-1/3 s^{-1}) is one-third the rate of the V/U signal. The manner in which the raw data (which are obtained at a 100 s^{-1} rate) are coded to the lower rate, consistent with the frame rate of the V/U signal, is described in Ref. 3.

Once input words and phrases are coded in terms of the formant representation, they can easily be modified for use with the synthesis program. Words can be lengthened or shortened, formants can be changed easily, and a pitch contour, different from the one originally spoken, can be superimposed on the data. Thus the vocal resonance data is available to the synthesis program in a form flexible enough to conform to the timing and pitch generated by the concatenation program.

The lower portion of Fig. 2 shows how the system assembles a synthetic message composed of words and phrases from the reference library. First, the answer-back program requests the word sequence for a specific message. The word concatenation program first determines timing data for the message (in one of several ways to be explained below) from an auxiliary program. The timing data is in the form of a word duration for each word in the output message. The concatenation program then accesses, in sequence, the control parameters for each of the words in the string. A duration modification adjustment on each word is first made, so the word duration in context matches the duration specified by the timing rules. Next the concatenation program smoothly interpolates the formant control parameters when the final part of any word and the initial part of the following word are both voiced. An interpolation algorithm designed to

TABLE I—CODING OF FORMANT PARAMETERS

Parameter	No. bits/frame	No. frames/second	No. bits/second
F_1 or F_p	3	33-1/3	100
F_2 or F_z	4	33-1/3	133-1/3
F_3	3	33-1/3	100
P	5	33-1/3	166-2/3
A_V or A_N	3	33-1/3	100
V/U	1	100	100
		Total	700
		Pitch	−166-2/3
Data rate for synthesis using calculated pitch data			533-1/3

produce physiologically realistic formant transitions is used. Finally a continuous function for pitch variation is produced for the whole message. All computed control parameters are outputted to a hardware digital speech synthesizer designed in accordance with Fig. 1. Digital-to-analog conversion produces a continuous synthetic speech output.

In the remainder of this section we will detail the way each of the above operations is carried out. In Section IV we will give an illustrative example of the use of the system for the synthesis of 7-digit telephone numbers. Further, we will describe a dialing experiment, using the synthesized numbers and the DDP-516 in an interactive manner, to estimate the communicative effectiveness of the synthetic speech.

The duration computations and the interpolation algorithm of the concatenation program depend upon a measure we call *"spectral derivative."*

3.1 *Spectral Derivative*

The control parameters are stored in the catalog at a sampling rate of 33-1/3 per second. When accessed and used in the synthesis program, however, they are interpolated to a rate of 100 per second; i.e., 10 ms between frames. For each 10-ms frame of a given word, a calculation is made of the absolute rate of change of the formant data from the previous frame. We call this calculation the spectral derivative, since it is a measure of how rapidly the spectrum is changing. The spectral derivative is used to determine where to lengthen or shorten a word, and is also used to determine at what rate a formant transition is made from one voiced interval to the next.

For each voiced 10-ms interval, the spectral derivative, SD_i, is computed as:

$$SD_i = \sum_{j=1}^{3} | F_j(i) - F_j(i-1) | \tag{1}$$

where i is the ith 10-ms interval in the word, and $F_j(i)$ is the value of the jth formant in the ith time interval. This measure of spectral change is an arbitrarily chosen one; several others could be considered. For instance a weighted sum of absolute values of formant change:

$$SD_i = \sum_{j=1}^{3} a_j | F_j(i) - F_j(i-1) | \tag{2}$$

might be a suitable replacement for equation (1) above. By adjusting the weights, a_j, the influence of changes in individual formants

can be made large or small. For example, by making a_2 much larger than a_1, or a_3, the spectral derivative is essentially the absolute change in the second formant. A more reasonable choice for the weight, a_j, might be the average value of the jth formant. The spectral derivative would then be the sum of relative changes in the formants. Although there are several possibilities for spectral derivative, the measure of equation (1) is the one we use throughout.

3.2 *Timing Calculation*

The timing calculation essentially consists of determining the duration of each of the words and phrases in the context of the message to be produced. There are several possible methods we have considered for determining these durations—ranging from fully automatic rules, which use syntactic and grammatical information, to manual insertion into the program of the desired timing sequence.

One technique, and the most accurate way of obtaining timing data, is to make measurements from a naturally spoken version of the message and manually supply these data to the program. This possibility is indicated at the bottom of Fig. 2 as the external timing data input to the timing subroutine. The timing data obtained in this manner are optimum and can be used to evaluate the efficacy of other aspects of the synthesis rules. This form of input is therefore important for evaluational purposes.

A second technique for obtaining word duration data is to make the duration of each word be some fixed percentage of its duration in isolation, independent of the message context. The motivation here is that the duration of the word in isolation is an overbound of its duration in context because of the unusually long vowels when spoken in isolation. Hence some shortened version of the word would suffice in many contextual situations. Clearly, the more limited the context of the message, the more applicable is the above approximation.

Another technique for obtaining durational data is by simple table-lookup procedures. Here the duration of every possible input, in every possible contextual position must be tabulated. For limited context messages, such as telephone number generation, this table-lookup procedure is an attractive way of generating timing information because of the limited number of situations which arise in practice. For more general situations, the amount of storage necessary would often become prohibitive.

The most sophisticated way of generating timing data is to make

calculations based on language rules. A syntactic and phonetic analysis of the printed text of the message is converted by rules into durational data about each of the phonemes in the message. For the most general cases of speech synthesis, i.e. unrestricted context, this kind of procedure is an absolute necessity to give good timing data. A computer program for such sophisticated analysis has recently been developed.[5,6] Continuing work is aimed at combining this program with the concatenation system.

3.3 *Word Duration Modification*

Once the duration of the jth word in the message has been determined by one of the methods discussed in the previous section, it is then necessary to modify the set of control signals of the reference version of the word to match the desired duration. Assume the duration of the reference version of the jth word is w_j frames and the desired duration is d_j frames where a frame is 10 ms long. If we define the symbols:

$$I_P(j) = \begin{cases} 1 & \text{if the end of the } (j-1)\text{st word is voiced, and the beginning of the } j\text{th word is voiced.} \\ 0 & \text{otherwise} \end{cases}$$

$$I_F(j) = \begin{cases} 1 & \text{if the end of the } j\text{th word is voiced, and the beginning of the } (j+1)\text{st word is voiced.} \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$b_j = w_j - d_j + \frac{t_c}{2}\cdot(I_P(j) + I_F(j)), \tag{3}$$

where

t_c = duration (in frames) over which voiced intervals are concatenated

and

b_j = number of frames to be eliminated from (if $b_j > 0$) or added to (if $b_j < 0$) the jth reference word.

The reason for the last term in equation (3) is that whenever adjacent voiced intervals occur between words they are smoothly merged

together. Hence their durations overlap and it is this last term which accounts for the overlap. Typical values of t_c are 4 to 10; i.e., 40 to 100 ms overlap between voiced words.

The manner in which the b_j frames are eliminated, or added in, is based solely on the spectral derivative. To eliminate frames, the b_j frames in the word having the smallest spectral derivatives are removed. To add frames, the region of the word having the smallest spectral derivative is located, and b_j consecutive frames are inserted in the middle of this region. The parameter values during the inserted frames are identical to those of the frame nearest the middle of the region. The rationale behind this technique is that to lengthen or shorten a word, by any significant amount, it is most desirable to do this in parts of the word where the spectrum is changing the least. Thus the dynamics of the word are always unaltered by this method. A linear compression, or expansion, of the whole word is a useful technique only when the compression or expansion ratio is close to 1.0. This is not always the case in synthesis, and so the above technique is used instead.

3.4 *Merging of Isolated Words*

Generally, the manner in which the control signals from isolated words are combined is by abutting them directly, once the timing modifications described above have been made. However when the words to be combined have a common voiced interval (i.e. the end of the one word and the beginning of the next word are both voiced), a more complicated procedure is used to merge the words. This is because merely abutting the words would often produce cases where formants on one side of the word boundary would be vastly different from formants on the other side of the boundary. If such data were merely abutted, then in synthesizing the message objectionable transients would be present at the boundary. To alleviate this problem, a merging interpolation algorithm is used. The algorithm is based on the spectral derivative, and provides smooth formant transitions from one word to the next.

The merging procedure combines data over the last t_c frames of the first word and the first t_c frames of the second word. The duration of t_c frames is called the overlap region of the words. The average spectral derivative during this region, for both words, is calculated as:

$$\overline{SD1} = \frac{1}{t_c} \sum_{i=1}^{t_c} SD1\,(i) \tag{4}$$

$$\overline{SD2} = \frac{1}{t_c} \sum_{i=1}^{t_c} SD2\,(i) \tag{5}$$

where $SD1(i)$, and $SD2(i)$ are the spectral derivatives for the two words during the t_c overlap frames. Using the notation:

$F_j(i)$ = value of the jth formant at frame i during the overlap region

$F_j^k(i)$ = value of the jth formant at frame i during the overlap region' for word k

then the interpolation function used is:

$$F_j(i) = \frac{F_j^1(i)\cdot(t_c - i - 1)\cdot\overline{SD1} + F_j^2(i)\cdot i\cdot\overline{SD2}}{(t_c - i - 1)\cdot\overline{SD1} + i\cdot\overline{SD2}}, \qquad i = 1, 2, \cdots, t_c \tag{6}$$

Figure 3 illustrates the type of interpolation performed for four simple cases. (Although all three formants are interpolated in the program, for simplicity just one formant is drawn in Fig. 3 for each word.) The interpolated curve always begins at the formant of the first word, and terminates at the formant of the second word. The rate at which the interpolated curve makes the transition from the formants of the first word, to those of the second word, is determined by the average spectral derivatives $\overline{SD1}$, and $\overline{SD2}$. For case 1, in Fig. 3, $\overline{SD1} \approx 0$ so $\overline{SD2} \gg \overline{SD1}$; hence the interpolated curve makes a rapid transition to the formant of word 2. Case 2 is the reverse of case 1: here $\overline{SD1} \gg \overline{SD2}$,

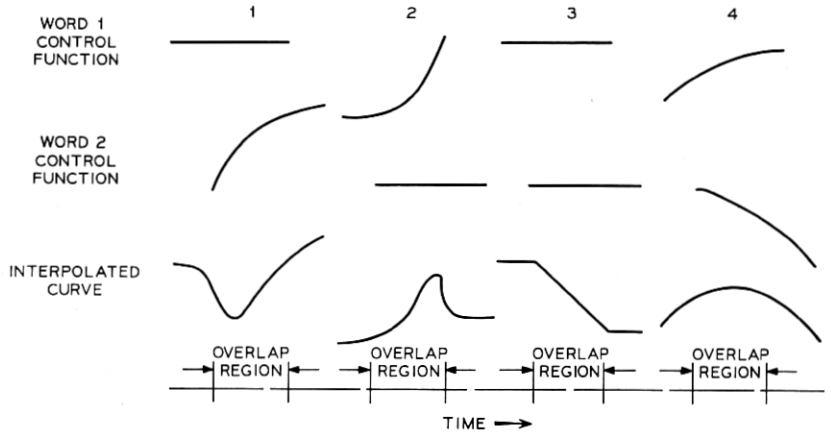

Fig. 3—Interpolation of control parameter contours for four typical cases.

so the transition does not occur until near the end of the overlap region. For case 3, both words have a small spectral derivative; hence the interpolation function degenerates into a linear transition. For case 4, both words have large spectral derivatives; hence the transition occurs about midway through the overlap region.

The data of Fig. 3 show that the interpolated formant function tends to be a smooth, continuous curve when the above technique is used. Values for t_c, the number of frames in the overlap region, have been from 4 to 10; i.e., 40 ms to 100 ms overlap.

3.5 *Pitch Calculation*

One of the most important aspects of speech synthesis is the determination of a suitable pitch variation for the message being produced. We have considered several ways of obtaining pitch information. These have included:

(i) Supplying a pitch contour extracted from a naturally spoken version of the message: These data, when used with similarly extracted timing data, give the most natural sounding messages that can be obtained with the technique. This form of input is most useful for evaluation purposes, but is not practical for an automatic system.

(ii) Using an archetypal pitch contour: For limited context applications this technique supplies a contour with realistic intonation, and hence is quite acceptable. The use of monotone pitch throughout the message is a special case of an archetypal contour, but such a contour gives an unacceptable drone to the speech, and hence would only be used in special situations.

(iii) Calculating a pitch contour by rule based on a stress analysis of the text of the message: This is a difficult task to do, but is most appropriate for an unlimited context, fully automatic system. Present research[5,6] on this topic makes it an attractive possibility for incorporating into a concatenation system.

(iv) Using the pitch variations associated with the isolated versions of the word, and concatenating them to give the overall pitch contour: This technique is unacceptable, unless several versions of each word are stored in the library, because the pitch contour of the isolated word tends to characterize the word only in isolation. The pitch usually rises sharply at the beginning of the word, and falls sharply at the end of the word. When concatenated, the words sound distinct, rather than merging into a continuous message.

3.6 *Gain Parameters*

The voiced gain parameter, A_V, is preserved on a frame-by-frame basis along with the formants. When formants are merged, the gain parameter is also merged. The unvoiced gain parameter, A_N, is also preserved on a frame-by-frame basis along with the fricative pole and zero. A_N is not required to be merged.

IV. AN ILLUSTRATIVE EXAMPLE

Figure 4 illustrates how these synthesis rules are applied in a typical case. At the top of this figure are shown the resonance data for four words spoken in isolation. The first and fourth words are entirely voiced, and the second and third words contain both voiced and unvoiced sections. The duration of each of the words spoken in isolation is shown by the w_i's in Fig. 4a. In order to form a message composed of these four words the following steps occur:

(*i*) The duration of each word in the specified context is determined.
(*ii*) Duration adjustments are made (frames removed or inserted) to match the timing of step *i*.
(*iii*) Since words 1 and 2 do not share a common voiced interval, the time adjusted control signals for word 1 are accessed.
(*iv*) Since words 2 and 3 do share a common voiced interval, all but the last t_c frames of the time adjusted control signals for word 2 are accessed and abutted to the controls from word 1.
(*v*) The last t_c frames from word 2 are interpolated with the first t_c frames from word 3, and added on to the previous control signals.
(*vi*) Since words 3 and 4 do not share a common voiced interval, the remaining control signals for word 3, and the time adjusted control signals for word 4 are added on to the previous control signals.
(*vii*) A pitch contour for the entire message is calculated.
(*viii*) The message is synthesized.

The resulting control signals and pitch contour are shown in Fig. 4b.

4.1 *Synthesis of Telephone Numbers*

For evaluation of this technique we chose the limited context situation of synthesis of the carrier phrase "The number is" followed by a 7-digit telephone number. Here, the timing was generated by a simple table-lookup procedure. The timing data we used are shown in Table

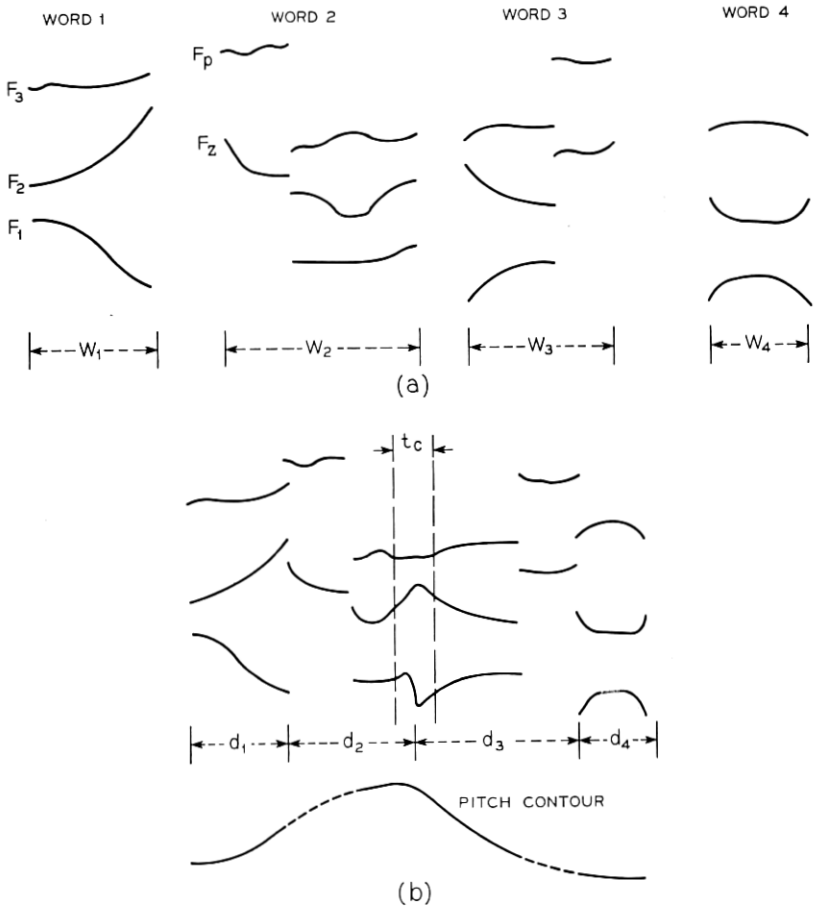

Fig. 4—Typical example of how control parameters are generated from the word library store. A message composed of four words is illustrated. All parameters are functions of time.

II. The table shows the digit duration (in milliseconds) as a function of the number of phonemes in that digit and its position in the string. The data in the table were obtained from measurements on real 7-digit numbers and, in effect, constitute first-order statistics on duration. The influence of context (position in the digit string) is easily seen in Table II. For example, any digit in the third position is from 50 to 90 percent longer than the same digit in the sixth position.

A single archetypal pitch contour was used in all cases. The arche-

TABLE II—SIMPLE TIMING RULES FOR 7-DIGIT NUMBERS

Position in Digit Sequence	Time, Milliseconds			
	Phonemes/Digit			
	1	2	3	4
1	250	330	410	490
2	280	330	390	450
3	450	500	560	610
4	260	300	340	380
5	340	370	410	440
6	230	280	340	390
7	290	380	460	550

typal form was taken to match as well as possible the general shape of the pitch contours measured in naturally spoken 7-digit numbers. This basic shape was used to calculate the pitch contour for each number string requested by the answer-back program. Informal listening suggested that this pitch contour was adequate as an initial estimate, and was a substantial improvement over the pitch information associated with individual isolated words.

The synthesis program ran on the Honeywell DDP-516 computer. The isolated digits were analyzed and stored in the computer memory at a data rate of 533-1/3 b/s. The concatenation program accepted an input sequence from the typewriter or card reader, computed the control signals for the message, smoothed them by programmed digital filters, and outputted the data to a hardware digital terminal analog synthesizer[7,8] in real time. Figure 5 shows a spectrographic comparison between a typical computer-generated 7-digit number, and a natural version of the same number. The timing and formant data of the synthesized example are seen to be reasonably good matches to those of the natural utterance.

4.2 *Dialing Experiment Using Synthetic Speech*

To evaluate the communicative effectiveness of the concatenated synthetic speech in a real dialing situation, we arranged for the DDP-516 computer to speak telephone numbers (both natural and synthetic) to a listener in a sound booth. The listener was provided a conven-

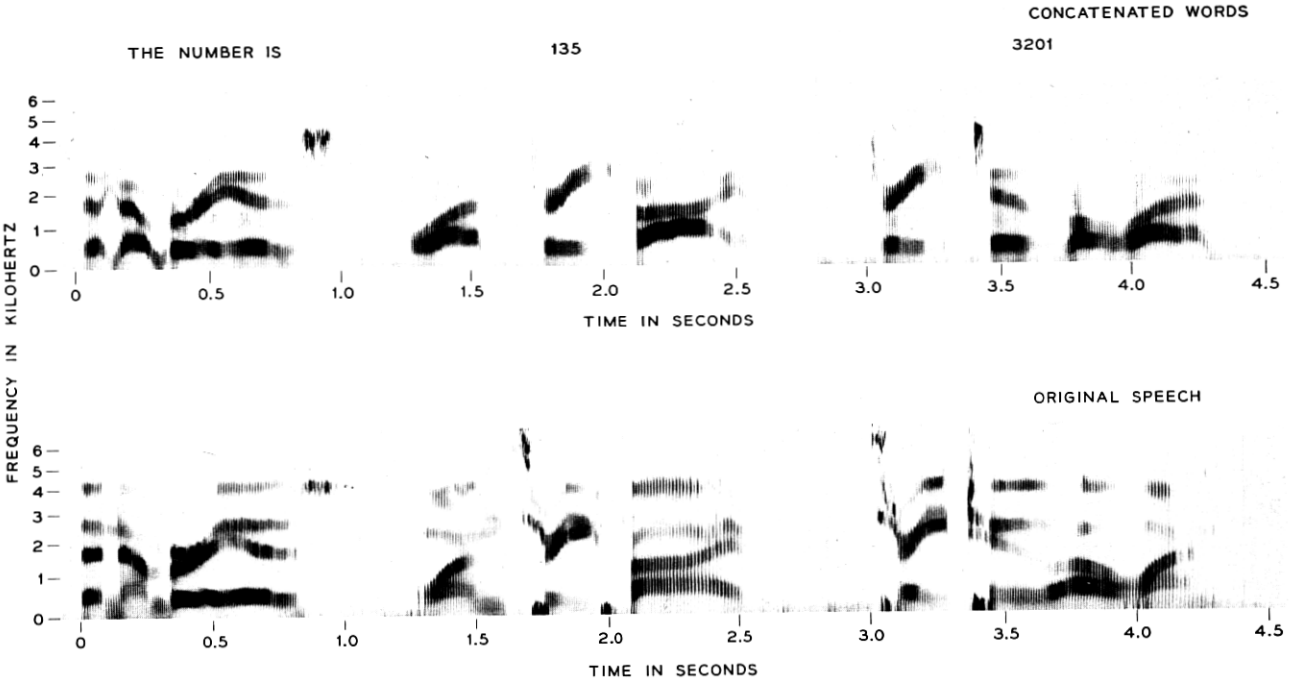

Fig. 5—Spectrogram comparison between synthetic and natural versions of a typical telephone number.

tional *Touch-Tone* telephone with which he could dial the numbers. A central office *Touch-Tone* decoder received the dial pulses, decoded them, and presented them via a data channel to the computer. The computer maintained a running analysis of the results. The experimental arrangement is shown in Fig. 6.

In the experiment we compared four types of speech. These included:

I. Naturally-spoken, 7-digit telephone numbers.
II. Naturally-spoken, isolated digits, abutted together.
III. Synthetic isolated digits, abutted together.
IV. Concatenated digits produced by the concatenation program method.

Listeners, seated in a sound booth, heard telephone numbers over the *Touch-Tone* telephone. After a prescribed delay, they were required to dial the number just heard. The DDP-516 computer generated the signal, read the number dialed, and tabulated the results.

Figure 7 shows the total number of dialing errors for 12 subjects. The dialing errors are broken down into digit errors (i.e., number of digits incorrectly dialed) and telephone number errors (i.e., number of phone numbers with one or more digit errors). The lower pair of curves shows the number of digit errors and the number of phone-number errors for 1-second delay in dialing of speech. The upper pair of curves shows the corresponding results for 5-seconds delay in dialing.

An analysis of variance of these data indicated that, at the 95 percent level of confidence, there existed no significant difference between dialing performances with the natural and the synthetic concatenated signals (i.e., between speech types I and IV). In other words, synthetic, concatenated speech is comparable to natural speech in dialing effectiveness.

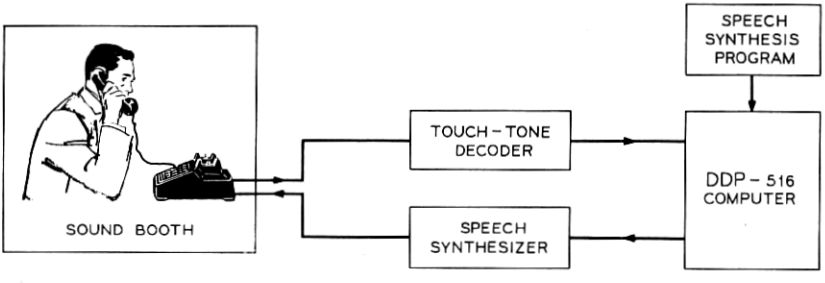

Fig. 6—Experimental arrangement used to measure the communicative effectiveness of several types of speech.

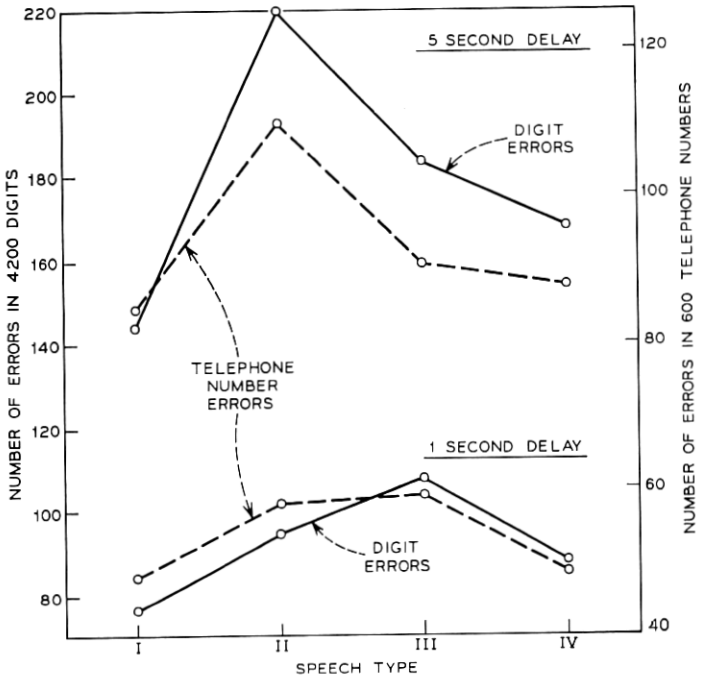

Fig. 7—Experimental results showing the total number of digit errors and telephone number errors. Four types of speech are tested: I, natural digits; II, natural, abutted digits; III, synthetic abutted digits; IV, concatenated digits. Response delays of 1 and 5 seconds are tested.

The differences between speech types I or IV and types II or III, however, was significant at the 95 percent level. That is, digital strings produced by simple abutting (II and III) led to a greater number of dialing errors. The suggestion is that the concatenation program is effective in reducing the dialing errors over that which would result from mere abutting of the digits.

Another factor of interest, of course, is the naturalness of the signal. Some preliminary informal experiments indicate that listeners rank the naturalness of these four signals in order of the "machine attributes," i.e., type I speech is ranked most natural, followed by types II, III, and IV. The synthetic concatenated signal has more machine-made features than any of the others—with pitch and duration both being calculated by machine. One might be willing to accept machine accent if the signal has attractive advantages in communicative ac-

curacy and economy of storage. Formant synthesis using the concatenation technique appears to have both.

V. ACKNOWLEDGMENT

We wish to thank J. D. Robinson for conducting the dialing experiment described here, and D. Bock for designing and implementing the *Touch-Tone* decoder/DDP-516 interface.

REFERENCES

1. Flanagan, J. L., *Speech Analysis, Synthesis and Perception,* New York: Academic Press, 1965, Chapter III.
2. Schafer, R. W., and Rabiner, L. R., "System for Automatic Analysis of Voiced Speech," J. Acoust. Soc. Amer., *47,* (February 1970), pp. 634–648.
3. Schafer, R. W., Rabiner, L. R., and Graham, N., "Formant Analysis, Synthesis and Coding for Computer Voice Response," unpublished work.
4. Rosenberg, A. E., Schafer, R. W., and Rabiner, L. R., "Effects of Smoothing and Quantization of the Parameters of Formant-Coded Speech," unpublished work.
5. Coker, C. H. and Umeda, N., "Text to Speech Conversion," IEEE International Convention Record, New York, N. Y., March 1970.
6. Flanagan, J. L., Coker, C. H., Rabiner, L. R., Schafer, R. W., and Umeda, N., "Synthetic Voices for Computers," IEEE Spectrum, *7,* (October 1970), pp. 22–45.
7. Rabiner, L. R. "Digital Formant Synthesizer for Speech-Synthesis Studies," J. Acoust. Soc. Am., *43,* (April 1968), pp. 822–828.
8. Rabiner, L. R., Jackson, L. B., Schafer, R. W., and Coker, C. H., "Digital Hardware for Speech Synthesis," Seventh International Congress on Acoustics, Budapest, August 1971.