EXPERIMENTAL METHODS FOR SPEECH SYNTHESIS BY RULE

BY

I. G. MATTINGLY

Reprinted from IEEE TRANSACTIONS
ON *AUDIO AND ELECTROACOUSTICS*
Volume AU-16, Number 2, June, 1968
pp. 198-202

Experimental Methods for Speech Synthesis by Rule

IGNATIUS G. MATTINGLY

Abstract—An experimental system for synthesis by rule has been developed, consisting of a parallel-resonance synthesizer, a laboratory computer, and two computer programs. To drive the synthesizer, values for each of its parameters must be specified at appropriate intervals. The "Rules" program (an elaboration of earlier work done with Holmes and Shearme at the Joint Speech Research Unit) accepts an input string representing an utterance phonemically, and an auxiliary input consisting of rules in tabular form for synthesis of the segmental and prosodic phonemes of a particular dialect. From these inputs, the sets of parameter values for the utterance are calculated. The "Executive" program reads the parameter value sets into memory and transmits them to the synthesizer. This program also enables the variation of the transmission rate, editing of the stored values, disk storage and retrieval of the parameter value sets for a large number of utterances, and synthesis of a sequence of utterances.

Manuscript received October 10, 1967. This work was supported by grants and contracts from the Information Systems Branch of the Office of Naval Research, the Prosthetic and Sensory Aids Service of the Veterans Administration, the National Institute of Child Health and Human Development, and the General Research Support Branch, Div. of Research Facilities and Resources, of the National Institutes of Health. This paper was presented at the 1967 Conference on Speech Communication and Processing, Cambridge, Mass.

The author is with Haskins Laboratories, New York, N. Y. 10017 and the Dept. of Linguistics, University of Connecticut, Storrs, Conn.

DURING the past year, an experimental system for speech synthesis by rule has been developed at Haskins Laboratories. The system consists of a resonance synthesizer, a DDP-224 digital computer and associated peripheral devices, and two computer programs.

The synthesizer was designed by F. S. Cooper and R. Epstein. The objectives of the design were flexibility, suitability for computer control, and reliability. For flexibility, the resonant circuits are in parallel, like those of the Joint Speech Research Unit (JSRU) synthesizer (Holmes, Mattingly, and Shearme[5]) and Lawrence's PAT,[6] rather than in series like those of OVE II (Fant, Martony, Rengman, and Risberg,[3]) and the University of Edinburgh's PAT in its present configuration. The requirement for computer control dictated the choice of a parameter set, parameter value ranges, and parameter bit assignments, such that the information in two 24-bit computer words would serve to specify the state of the synthesizer for one increment of time. Reliability has been achieved thanks to recent advances in electronics (notably the greatly reduced cost of high-quality operational amplifiers). As in the case of the synthesizer described by Coker[1] and Tomlinson,[9] these advances have made it practical to build an instrument which is believed to be more stable than any of the first-generation resonance synthesizers.

Fig. 1 shows a block diagram of the Haskins synthesizer. The first four formants of the vocal tract are represented by resonant circuits 1, 2, 3, 4. Each of the first three resonances can be varied through an appropriate range and is designed to preserve a constant bandwidth over its range. The design of the variable resonant circuits has been described by Epstein.[2] In addition to the resonant circuits, there are three high-pass filters which shape the high-frequency noise for the sounds [s], [ʃ], and [f, θ]. As yet, there is no nasal circuit, but one is planned. Vocal cord excitation is simulated by a buzz generator; turbulent excitation by a hiss generator. By setting the buzz and hiss switches appropriately, it is possible to excite the resonances with buzz, with hiss, with both, or with neither. Closing hiss switch 1 introduces low-level noise, intended to increase the naturalness of the synthesized speech by making the excitation somewhat more irregular; closing hiss switch 2 introduces higher level noise for the synthesis of sounds for which noise in the spectrum is one of the cues. Depending on the setting of the fricative selection switches, the hiss generator also excites one of the three high-pass filters, or none of them. The amplitude of the excitation of each of the resonances and filters is regulated by a modulator. After all outputs have been added in a mixer, the overall amplitude of the resulting signal is further regulated by a variable gain amplifier.

While manual control is possible, the synthesizer is normally driven dynamically from two 24-bit control registers, into which digitally coded values for the various parameters are loaded by the computer in the

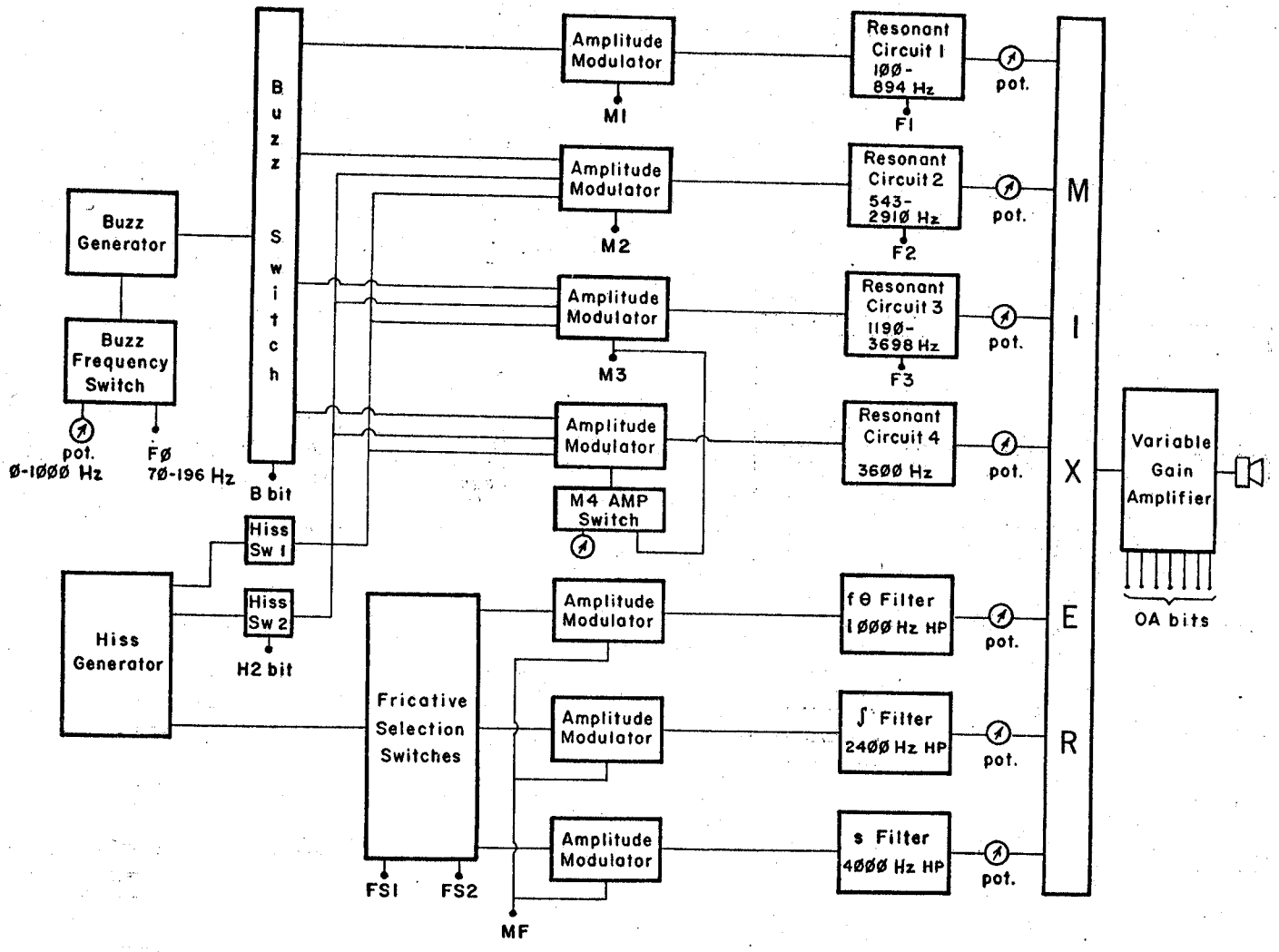

Fig. 1. Block diagram of the Haskins resonance synthesizer.

format shown in Fig. 2. The switch bits B, H2, FS1, and FS2 control the corresponding switches directly. For each of the frequency parameters Fϕ, F1, F2, F3, a voltage proportional to the control value is transmitted to the appropriate circuit by a digital-to-analog converter. For the amplitude parameters M1, M2, M3, MF, OA, it is convenient to have the digital control value in the register linear with the decibel scale. The control values are, therefore, decoded antilogarithmically. For M1, M2, M3, MF, a digital-to-analog converter transmits a control voltage proportional to the decoded value to the corresponding modulators. The decoded OA bits control the variable gain amplifier digitally. M1, M2, M3 have a 21 dB-range, MF a 20-dB range. The OA parameter adds an additional 28 dB to the available range. The relative gains of the outputs of the resonances and filters can be preset by potentiometers.

A 24-bit buffer storage unit links the synthesizer to the computer, which is operated by the experimenter himself. Using a program called the "Executive" (similar to the program described by Scott, Glace, and Mattingly[8]), he communicates with the computer by typing one of a set of mnemonic commands on the console typewriter. The command TALK calls the routine which drives the synthesizer. The routine assumes that the utterance to be synthesized is represented in memory by a series of pairs of 24-bit control words, each pair containing a set of parameter values in the format shown in Fig. 2. The first and then the second word in each pair is transferred from memory to the buffer storage unit, and from the buffer storage unit to the control registers of the synthesizer. After one pair of control words has been transmitted to the synthesizer, the computer waits a certain number of milliseconds before transmitting another pair. Time is measured by an

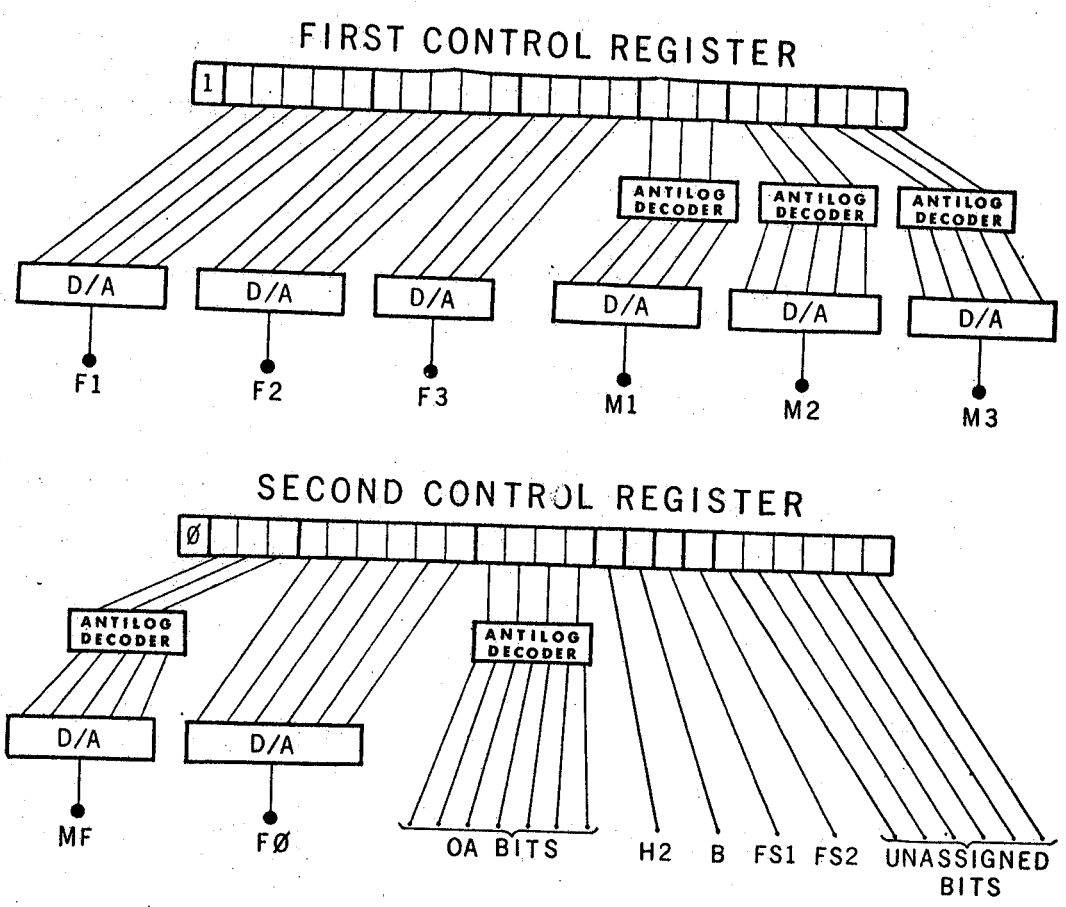

Fig. 2. Digital control for the Haskins synthesizer.

electronic clock, which is programmed to set a sense line once every millisecond. During the waiting period, the synthesizer is controlled by the parameter value set contained in the most recently transmitted pair of words. The constant in the program, which determines the length of the waiting period and, therefore, the speed of the utterance, can be changed temporarily from the computer console during synthesis, or permanently with the TIME routine. When the entire series of control words for an utterance has been transmitted, there is a wait of 1 second before synthesis recommences. The utterance is synthesized repeatedly until the experimenter interrupts the process by raising a sense switch on the computer console. The output of the synthesizer is available at a jack near the console, and can be heard through headphones, or a loudspeaker, or recorded on the tape recorder.

To load the computer memory with control words, as required for TALK, the READ routine reads the parameter values for an utterance into memory from a paper tape prepared in a decimal format. The PNCH routine punches out on paper tape some or all of the stored control words in binary form. This tape can be read back into memory using the command INBI. The PRIN routine prints out some or all of the stored control data in a columnar decimal format.

The "Executive" also includes a group of routines which enable editing of the control data stored in memory. The SERT routine inserts a control word pair following a specified pair. STUT increases the duration of part of the utterance by repetition of a selected control word pair. DELE deletes one or more control word pairs. PARA revises one or more sequential values of a selected parameter. TERP replaces part of the series of values for a parameter with new values, calculated by linear or nonlinear interpolation between two specified values. Since, unlike the linear rule, the rule for nonlinear interpolation (to be given later) does not produce the same result when applied in opposite directions, interpolation in both forward and reverse directions is possible. A number of other routines control an IBM disk file, which, for purposes of the program, is divided into 512 storage locations. At each location, control data for as much as 5 seconds of speech may be stored, assuming a 5-ms waiting period. To clear the disk of previously stored data, the experimenter uses the BEGN routine. To copy onto the disk the control data currently in memory, he uses SAVE, specifying a storage location; to retrieve the control data from a specified storage location, he uses PULL. SEQU synthesizes a long series of utterances at specified intervals from specified file locations. SEQT synthesizes a series of utterances for audio

recording, the timing being controlled by pulses on the audio tape; these pulses can be recorded at specified points on the tape using the MARK routine. With these routines, a set of utterances can be recorded in a number of different orderings without time-consuming tape dubbing and splicing.

What has been described so far is actually a general-purpose speech synthesis system and, in fact, has been used extensively by members of the Haskins staff for various purposes, notably the generation of stimuli for psycholinguistic experiments. The experimenter has the advantage of immediate playback and quick revision, just as he does with a system in which the synthesizer is controlled by a function generator; yet he can also plan and record his experiments quantitatively and keep numerical records.

The purpose of the "Rules" program is to calculate the input to the "Executive" from a phonemic transcription. This program is a near relation of two programs developed at JSRU, one for the synthesis of segmental phonemes (Holmes *et al.*[6]) and the other for the synthesis of prosodic features (Mattingly[7]). The functions of these two programs have been combined, and various refinements have been made: nonlinear formant transitions, machinery for the statement and application of allophone rules, a more elaborate procedure for the specification of excitation, and generalization of the method of calculating Fø.

The tasks of the "Rules" program are the calculation of transitions and steady states for the formant frequency and amplitude parameters F1, F2, F3, M1, M2, M3, MF, OA; the specification of the excitation and fricative parameters H2, B, FS1, FS2; and the calculation of Fø. The inputs to the program are a set of rule tables and a string of phonemic symbols, such as the following:

H EE – " T OO K – A – ' W AW K – " E V

. R EE – ' M AW R N ING .

The tables include a group of variables used in the Fø calculation; a set of phoneme tables containing the information needed to calculate the other parameters; and a set of allophonic rules, each of which contains a statement of the phoneme class and the context to which the rule applies, and the change in the phoneme table required to implement the rule. The phoneme class is specified in terms of a few elementary distinctive features. Similarly, the context is described as a combination of such elementary contextual features as "prejunctural," "prevocalic," etc. It appears that a relatively small inventory of contextual features suffices for the statement of a substantial number of rules. The allophone rules can take care of both segmental allophones (e.g., exploded and unexploded voiceless stops) and prosodic allophones (e.g., durational variation induced by stress).

As the input string is read in, each phoneme is classified according to its distinctive and contextual features. Before the calculation of the parameter values for a phoneme or its neighbors takes place, this classification is compared with each of the allophone rules to determine which rules apply, and the appropriate modifications are made in the phoneme table for that particular occurrence of the phoneme.

For each of the frequency and amplitude parameters, an initial transition, a final transition, and a steady state must be calculated, in turn, for each phoneme. This calculation depends on the information in the tables for the current phoneme and its predecessor and successor, as modified by the allophone rules. The steady state is specified in the phoneme table. To calculate the transition, the values of the parameter at the boundaries between the current phoneme and the preceding and following phoneme are first determined by a procedure essentially similar to that described in Holmes *et al.*[6] The initial and final transitions are now calculated by interpolating from the steady-state value of the phoneme backwards to the initial boundary value and forwards to the final boundary value for the prescribed durations. For the amplitude parameters, the equation

$$X_a = S + Ka$$

is solved for odd positive values of a less than $2d$, where S is the steady-state value of the parameter, d is the transition duration, and K is a constant chosen so that X_{2d} = the boundary value. For the formant frequency parameters, the procedure is the same, except that the nonlinear equation

$$X_a = S + Ka^2$$

is used, so as to produce a transition which curves more sharply as it nears the boundary.

In the buzz-hiss calculation, B and H2 are calculated together as a 2-bit number; similarly, FS1 and FS2 are calculated together. The calculation of the fricative selection and the buzz-hiss excitation has been made somewhat more elaborate than in the JSRU segmental program, in order to minimize the number of phonemes which must be treated as consisting of two or more parts with a separate table for each part. For the fricative selection, one can specify an initial and a final fricative value, and the proportion of the total duration for which each is to be used. For the B-H2 excitation, similarly, an initial value and a final value can be specified. Furthermore, the beginning of a phoneme can be devoiced by the previous phoneme. Voicing during stop closure is treated, in effect, as if it were a special kind of excitation; a special parameter value set representing its excitational and spectral characteristics is a constant of the program, and can be used for a specified proportion of the duration of the phoneme.

Fø is calculated syllable by syllable, the boundaries of a syllable being defined by an arbitrary rule. The calculation does not use the Fø coding directly, which is

linear with frequency, but rather a set of "pitch values" logarithmically related to frequency. Once the pitch values have been calculated, they are converted to Fø values by table lookup. For the part of the sentence up to the tonic syllable, values assigned to variables in the pitch table specify changes in the current value of the pitch and the slope of the pitch function for the syllable, depending on whether the syllable is stressed or, if unstressed, whether it follows a stressed syllable; and on whether a voiceless interval precedes the voiced part of the syllable. Other variables in the pitch table specify the shape of the contour (fall, rise, or fall–rise, depending on the intonation mark in the input sentence) to be imposed on the tonic vowel and any following syllables.

The system just described is not itself a set of rules for synthesizing speech, but merely a general framework within which a set of rules can be stated, tested, and improved. One set of rules, developed by Haggard for British English, is described in another paper[4] presented at this conference; another set, for General American, is being developed by the author. Experience with the system, gained in developing these rules, indicates that while both the synthesizer and the programs have certain limitations (not all speech sounds can be easily synthesized, nor all rules conveniently stated), it will be some time before the rules reach a state of sophistication at which extensive modifications of the synthesizer or the programs are essential for further progress.

Acknowledgment

The disk file routines were programmed by A. Singer.

References

[1] C. H. Coker, "Real-time formant vocoder, using a filter bank, a general-purpose digital computer, and an analog synthesizer" (abstract), *J. Acoust. Soc. Am.*, vol. 38, p. 940, November 1965.

[2] R. Epstein, "A transistorized formant-type synthesizer," Haskins Labs., New York, N. Y., Status Rept. on Speech Research SR-1, February 1965.

[3] G. Fant, J. Martony, U. Rengman, and A. Risberg, "OVE II synthesis strategy," *Proc. Speech Communications Seminar* (Speech Transmission Lab., Royal Institute of Technology, Stockholm, Sweden, 1963), paper F5.

[4] M. P. Haggard and I. G. Mattingly, "A simple program for synthesizing British English," Conf. on Speech Communication and Processing, Cambridge, Mass., November 1967. Also see *IEEE Trans. Audio and Electroacoustics*, vol. AV-16, pp. 95–99, March 1968.

[5] J. N. Holmes, I. G. Mattingly, and J. N. Shearme, "Speech synthesis by rule," *Language and Speech*, vol. 7, p. 127, July–September 1964.

[6] W. Lawrence, "The synthesis of speech from signals which have a low information rate," in *Communication Theory*, W. Jackson, Ed. London, 1953. p. 460.

[7] I. G. Mattingly, "Synthesis by rule of prosodic features," *Language and Speech*, vol. 9, p. 1, January–March 1966.

[8] R. J. Scott, D. A. Glace, and I. G. Mattingly "A computer-controlled, on-line speech synthesis system," *1966 IEEE Communications Conf., Digest of Tech. Papers*, p. 104.

[9] R. S. Tomlinson, "SPASS—an improved terminal-analog speech synthesizer" (abstract), *J. Acoust. Soc. Am.*, vol. 38, p. 940, November 1965.

Ignatius G. Mattingly, for a biography and photograph, please see page 99 of the March, 1968, issue.