Behavior Research Methods & Instrumentation
1976, Vol. 8 (2), 189-196

SESSION X
CONTRIBUTED PAPERS: STIMULUS GENERATION

DOMINIC W. MASSARO, *University of Wisconsin, Presider*

Real-time speech synthesis

MICHAEL M. COHEN and DOMINIC W. MASSARO
University of Wisconsin, Madison, Wisconsin 53706

This paper describes how a speech synthesizer can be controlled by a small computer in real time. The synthesizer allows precise control of the speech output that is necessary for experimental purposes. The control information is computed in real time during synthesis in order to reduce data storage. The properties of the synthesizer and the control program are prsented along with an example of the speech synthesis.

Real-time minicomputers are now fairly commonplace in the psychological sciences. The psychologist expects the computer to control the sequence of events in an experiment, present the stimuli, record the participant's response, and analyze the cumulative results. Although the computer appears to be well educated, he (or she) has developed language abilities in the reverse order of the scientist. Man spent many centuries speaking before writing was developed, whereas the computer writes (or at least prints) but "speak less than thou knowest" (King Lear, Act I, Scene IV). If only our small computers could speak scholarly and wisely (or at least intelligibly), they could be assigned to many additional useful tasks. This paper describes how a relatively cheap synthesizer can be controlled by a small computer in real-time.

Artificial speech has been synthesized by mechanical, electronic, and computer simulation techniques. [Coker, Denes, & Pinson, Note 1; Dudley & Tarnoczy (1950), Mattingly (1968), Holmes (1972), and Flanagan (1972) discuss the historical developments in speech synthesis.] The electronic resonance synthesizer is currently one of the most efficient and popular techniques of speech synthesis. Whereas mechanical synthesis attempted to stimulate the articulatory properties of speech production, electronic synthesis focuses on the acoustic structure of speech. The focus on acoustic structure rather than articulatory structure in synthesizing speech led to synthesizers that were terminal analogs rather than direct analogs of speech. Whereas a direct analog synthesizer would have a direct representation of each component movement or sound in the vocal tract in the synthesizer, the terminal-analog synthesizer simply attempts to duplicate the final speech output. The acoustic speech signal can be considered as a sound resulting from a two-stage process. The sound source of the first stage is modified by the time-varying filter characteristics of the vocal tract at the second stage. In this model, the sound source and the characteristics of the resonant circuits representing the vocal tract can be independently varied to produce the sound output.

The preparation of this paper as well as the purchase of the synthesizer was supported in part by U.S. Public Health Service Grant MH-19399. We would like to thank Jim Bryant for his help in preparing the demonstration tape for the conference.

The desired sound can, therefore, be produced by specifying a small number of parameters controlling the significant acoustic dimensions of the speech. The sound source can be either voice or noise. The voice source stimulates vibration of the vocal cords in real speech; it consists of a periodic quasi-sawtooth-shaped sequence of pulses. The frequency of pulsing is referred to as the fundamental (F0) and is heard as the pitch of the speaker's voice and the intonation pattern of the message. The noise source stimulates the forcing of air through some constriction in the vocal tract. It has the properties of a pseudorandom noise generator.

The sound source is fed into the resonant circuits at the second stage of synthesis. For the production of vowel sounds, the resonant circuits are set to correspond to the acoustic resonances or formants of the vocal tract. The effect of each resonator is to emphasize the energy at its set frequency and to produce additional energy at its formant of the sound to be synthesized. The resonators can be arranged in parallel or in series. Parallel synthesizers combine the output of individual resonating circuits (Mattingly, 1968). In series synthesizers, the resonating circuits are arranged so that the sound source is fed into the first resonator and the resulting output is the input of the second resonator, and so on. The series synthesizer better approximates the vocal tract in which the sound is modified in a serial fashion as it flows through the vocal tract (Fant & Martony, 1962; Flanagan, 1957).

There are a number of dimensions that must be considered in determining the most appropriate speech synthesis system for experimental use. For the synthesizer these are cost, flexibility, degree of control, and programming complexity. For the control programs one must consider the power, speed, and memory capacity of the controlling machine. We desired a synthesizer that allowed precise control over the synthesized signal since

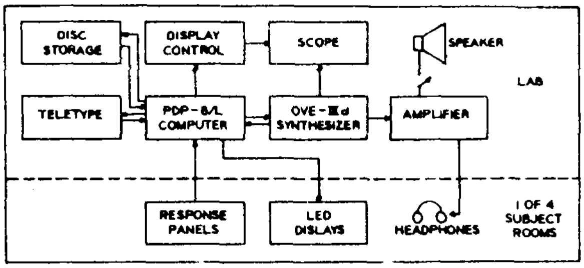

Figure 1. Hardware configuration of laboratory used for speech perception research.

our research program involves the manipulation of the microstructure of speech stimuli. Accordingly, the commercial synthesizers that have phoneme or syllable sounds hard wired into the system would not be appropriate (e.g., the VOTRAX synthesizer, Rahimi & Eulenberg, 1974). Figure 1 shows the general layout of equipment used in our laboratory for speech perception research. Given the small memory capacity (8K) of our PDP-8/L and the fact that the synthesis would have to be carried out during the experiment proper, we decided that rather than calculating the control information in advance, as done by the Haskins system (Mattingly, 1968), all interpolations would have to be carried out dynamically during synthesis. This method allows for very compact and flexible specification of the stimuli. Given that synthesis is carried out during the experiment itself, the control program must be very small. Memory storage must also be allocated to the experimental control subroutines, the main experimental program, data storage, synthesizer control specifications, and the disk system resident. Our current synthesizer program, written as an assembly language subroutine callable from a main experimental program, takes only 317 decimal locations. (A somewhat smaller program could probably be written for a machine that had hardware arithmetic.) The cost of the synthesizer and the interface to the computer had to be minimal. The actual cost came to less than $4000.

Our present synthesizer operating system is not meant to compete with large-scale speech synthesis programs. We do not presently foresee using our machine for synthesis by rule, i.e., a system that automatically takes the user from a phonemic transcription input to synthesized speech output (Holmes, Mattingly, & Shearme, 1964; Mattingly, 1968). Rather than quantity, we are striving for quality within the confines of simplicity and compactness.

THE SYNTHESIZER

OVE-IIId[1] is a formant series synthesizer stimulating the vocal tract. The synthesizer is a very compact device measuring 19 x 14 x 1.75 in. high and is rack mounted. The OVE-IIId synthesizer is the theoretical descendant of such synthesizers as OVE-II (Fant & Martony, 1972) and SPASS (Tomlinson, 1966). [For a more detailed description of the OVE-III see Liljencrants (1968).] The synthesizer incorporates three parallel branches for the synthesis of vowels, fricatives, and nasals (see Figure 2). The synthesizer is digitally controlled. Control data are received over a 10-bit bus and stored in digital registers. One 10-bit control word contains a 4-bit address code and a 6-bit logarithmic data code (see Figure 3). The frequencies are incremented in 3% steps and the amplitudes in 2-dB steps.

When a parameter word is presented to the synthesizer along with a set command, a 1-μsec control cycle starts. A demultiplexer within the synthesizer then gates the data to the appropriate register according to the address code. The data in these registers are used as coefficients by the analog circuitry in generating the pertinent waveforms. Parameter words can also be entered manually from toggle switches and a set button on the front panel.

Vowels may be synthesized by introducing the voice source to the vowel formant branch through the level control amplifier AV. The frequency of the voice source is controlled by F0. After passing through a correction network and mixer KH, the sound is directed through formant resonators corresponding to the first five formants, F1-F5 (cf. Figure 2). Only the first three formants can be controlled; formants F4 and F5 are present at 3.5 and 4.0 KHz, respectively. The center frequencies of the first three formants are controlled by F1, F2, and

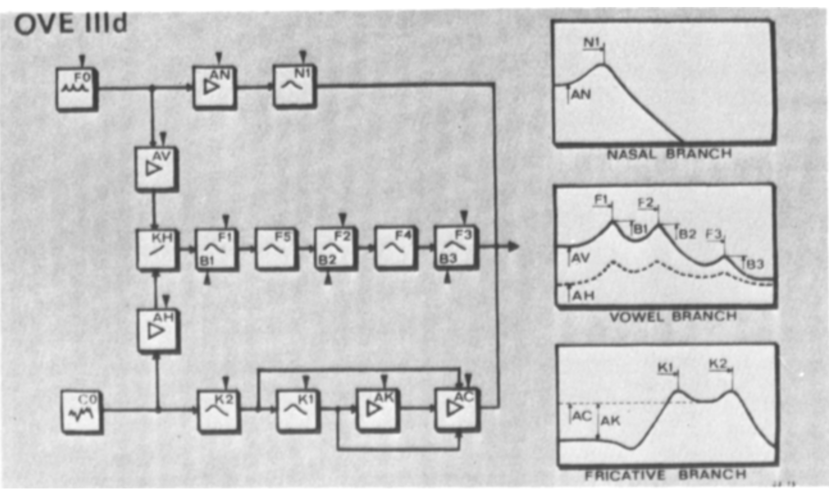

Figure 2. Block diagram of OVE-IIId speech synthesizer. See text for explanation.

PARAMETER	mne	ADDRESS dec	A3 2 1 0	DATA D5 4 3 2 1 0	RANGE	STEP
	SP	0	0 0 0 0			
Vowel excitation	AV	1	0 0 0 1	X X X X	-∞, 2 - 30 dB	2 dB
Aspirative excitation	AH	2	0 0 1 0	X X X X	-∞, 2 - 30 dB	2 dB
Nasal excitation	AN	3	0 0 1 1	X X X X	-∞, 2 - 30 dB	2 dB
Pitch	F0	4	0 1 0 0	X X X X X X	50 - 308 Hz	3 %
Vowel formant 1	F1	5	0 1 0 1	X X X X X X	200 - 1234 Hz	3 %
Vowel formant 2	F2	6	0 1 1 0	X X X X X X	504 - 3109 Hz	3 %
Vowel formant 3	F3	7	0 1 1 1	X X X X X X	800 - 4935 Hz	3 %
Nasal formant	N1	8	1 0 0 0	X X X X X X	200 - 1234 Hz	3 %
F1 bandwidth increment	B1	9	1 0 0 1	X X X X	0 - 188 Hz	12 Hz
F2 bandwidth increment	B2	10	1 0 1 0	X X X X	0 - 470 Hz	31 Hz
F3 bandwidth increment	B3	11	1 0 1 1	X X	0 - 600 Hz	200 Hz
Fricative excitation	AC	12	1 1 0 0	X X X X	-∞, 2 - 30 dB	2 dB
Fricative formant 1	K1	13	1 1 0 1	X X X X X X	800 - 4935 Hz	3 %
Fricative formant 2	K2	14	1 1 1 0	X X X X X X	1600 - 9870 Hz	3 %
Fric. pole/zero ratio	AK	15	1 1 1 1	X X X X X X	0 - 31.5 dB	0.5 dB

Figure 3. Specification of speech synthesizer control parameters.

F3. The intensity levels are adjusted automatically as a function of the formant frequencies. Some control of the formant intensity levels is possible through specification of the bandwidth controls B1, B2, and B3. Narrowing the bandwidth increases the peak intensity of the appropriate formant.

The fricative sounds make use of the fricative branch of the synthesizer. The source CO, a pseudorandom noise generator, is modified by two cascaded formant resonators, K1 and K2. The frequencies of K1 and K2 may be independently controlled. Control AK allows the introduction of a variable antiformant into the fricative branch. The intensity level of the resulting signal is controlled by the fricative excitation AC.

The stop consonants make use of both the vowel and fricative branches. A stop consonant-vowel syllable can be partitioned into burst, transition, and steady state vowel segments. The burst created when the stop consonant is released is synthesized along the vowel and/or fricative branches. The burst is followed by a transition to the levels of the following steady state vowel. Voiceless aspirated sounds are synthesized by introducing the noise source CO into the vowel branch through the level control amplifier AH. The presence of voicing without aspiration vs. the presence of aspiration without voicing during the transition period are major cues to distinguishing /b, d, g/ from /p, t, k/.

The nasal sounds /m, n, ŋ / are similar to the voiced stop consonants except that the additional nasal formant N1 is used. The nasal formant is excited by the voice source F0, and the intensity level is controlled by AN. The glides or semivowels are synthesized through the vowel branch.

To communicate with the synthesizer it was necessary to construct an appropriate interface. On the PDP-8/L most I/O is accomplished through the accumulator (AC). From the AC, a 12-bit buffered AC (BAC) bus is distributed to all peripheral devices. There is also a set of six lines (BMB) for device selection and several buffered I/O pulse (BIOP) lines. Both the PDP-8/L and the input stage of the OVE-IIId use TTL logic, so no level conversion was necessary. A schematic of the interface is shown in Figure 4. When a I/O instruction specifying device 47 is executed, the 7430 gate is set to false and, negated by the 7402, enables the BIOP gates. If a BIOP-2 pulse or an initializing pulse is issued by the computer, a master clear pulse is sent to the OVE-IIId for about 600 nsec. This clears all registers in the OVE-IIId. If a BIOP-4 pulse is issued, the 74121 issues a 3-μsec pulse which gates the BAC lines to the OVE-IIId by enabling the 7408s and sends a set request pulse to the synthesizer.

STIMULUS SPECIFICATION

In order to control the synthesizer, detailed information about the speech sound to be produced must be specified. This information must then be coded and typed into a file on the computer disk. A suitable computer program (the PALD assembler with a supplemented symbol table) can then translate the code into a form acceptable to the synthesizer control subroutine.

The first step in stimulus description is to divide the speech sound into timed segments. For example, in the coding of a simplified syllable /ba/, one would have two segments, first a transition period and then a steady state period. This is illustrated in the schematic spectrogram in Figure 5. After dividing the sample into segments, one must specify the desired values of the control parameters at the segment boundaries. These values may be obtained from a frequency table. For example, if F0 is desired to be 126 Hz, the proper value is 40_8. Only those parameters that are to be changed from one segment to the next need be specified. In our example, at time a one would specify values for AV, F0, F1, and F2. At time b one would only specify values for F1 and F2. The programmer indicates whether the parameters that differ between adjacent boundaries should be interpolated or whether they should maintain their present values until the next boundary (steady state). Currently, all interpolation is carried out in a linear fashion, but we are developing an exponential interpolation for more realistic synthesis.

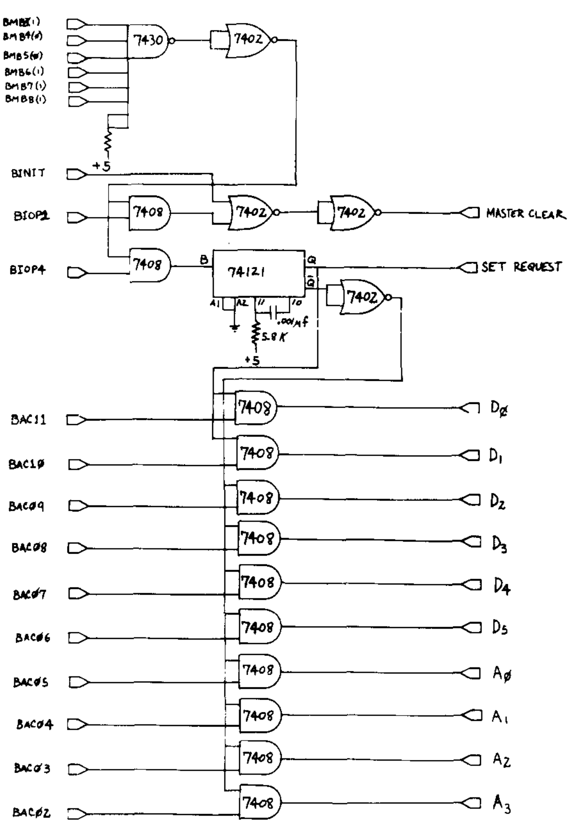

Figure 4. Schematic diagram of computer-synthesizer interface.

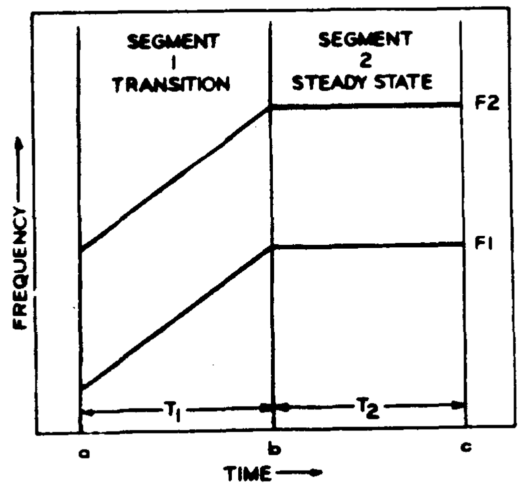

Figure 5. Schematic diagram of simplified /ba/.

The data structure of the sample specifications in the computer memory consists of two types of elements: control blocks (CB) and parameter lists (PL). Each of these is composed of a group of control fields in adjacent 12-bit memory locations. Figure 6 shows the structure of the elements. The CB always consists of three computer words. The first field in the CB is the time field (T), which indicates how much time there is to the temporally adjacent boundary (in 5-msec units). The next field is the plot (P) field. This is used by a display program that will be described later. The control field (C) indicates whether the segment is steady state (SS: C = 1), interpolating forward (IF: C = 3), or interpolating backward (IB: C = 2). If C is SS or IF, T specifies the time taken to get from the currently specified parameter values to those next specified. If C is IB, T specifies the time taken to get from the parameter values at the preceding segment boundary to those addressed by the current CB. The reason for these two interpolation directions will be explained shortly. The PPL field contains the memory address of the PL associated with the CB. If PPL = 0, then no PL is associated with this CB. The PCB field contains the memory address of the next CB. If PCB = 0, then the present CB is the last. The PL is composed of a variable number of computer words, and each word is divided into three fields. The PN field contains the 4-bit address of the specified parameter in the synthesizer (see Figure 3). The PV field contains the 6-bit value of that parameter. The E field indicates whether or not the PL word is the last in the list. If it is the last, E is set to 1, otherwise it is 0. Figure 7 shows how those data structures are connected in memory. An

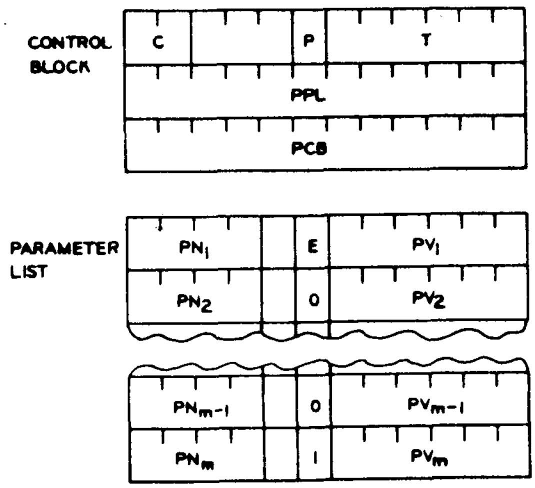

Figure 6. Structure of data specification elements.

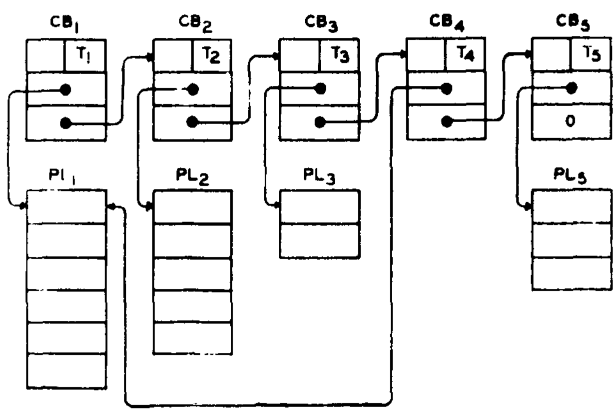

Figure 7. Connections of control blocks (CB) and parameter lists (PL) in memory. A field with a dot holds the address indicated by the pointed arrow.

arrow in the illustration indicates that a given field holds the address of what is pointed to. Each CB need not reference a unique PL; rather, a PL can be referenced by many CBs. In Figure 7, for example, PL_1 is referenced by both CB_1 and CB_4.

Let us return for a bit to the syllable /ba/. The data structure for our example is illustrated in Figure 8. AV* refers to the 4-bit number which corresponds to the AV register address within the synthesizer. Note that, although /ba/ has three segment boundaries, only two CBs are necessary to describe it. In general, a sound divided into m segments requires m CBs. Figure 9 shows a schematized diagram of the syllable /bag/ cut into three segments. The data in Figure 8 specify the sound up to the c boundary. From c to d it is necessary to interpolate the values of F1 and F2. Rather than specifying F1 and F2 in a CB representing c and constructing another CB to represent the values at d, we can make use of the IB feature. It is possible to construct a CB for the d boundary in the IB mode which specifies the time from d back to c. In this case, there is no CB representing the segment boundary c.

This scheme of data specification precludes the direct specification of transitions across two or more segments. Consider the case in which it is desirable to have F0 fall linearly from time b to time d in /bag/ (cf. Figure 9). It would not be sufficient to specify the F0 values at times b and d, respectively. The programmer must calculate the appropriate F0 value at time c and include this value in the PL of the CB representing c. In general, to have a parameter interpolate across a segment boundary, one must calculate the value at the intermediate boundary and include it in the PL of the intermediate CB.

The syntax of the language used to describe the speech sample is as follows:

```
CONTROL BLOCK FORMAT:
CBNAME,        CC TT P
               PNAME
               NCNAME
```

where CBNAME, PNAME, NCNAME are symbolic names up to six alphanumeric symbols starting with a letter. They name the CV, the PL, and the next CB. CC = "SS," "IB," or "IF" for steady state, interpolate backward or interpolate forward, respectively. TT = number of 5-msec time units in octal $0 \leq TT \leq 77_8$. P = optional flag for display routine.

```
PARAMETER LIST FORMAT:
PNAME,         PN1 PV1
                .
                .
                .
               PNm PVm E
```

where PN = parameter address name, e.g., "AV," "F0," "K1," PV = parameter data value in octal $0 \leq PV \leq 77_8$, and E = end flag, include only for last parameter in the current parameter list. According to this syntax, we would code the sound display represented in Figure 8 as follows:

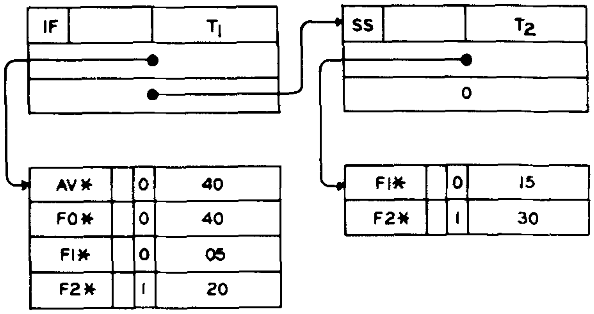

Figure 8. Data structure for /ba/.

```
           *12φφφ
CONTR1,    IF 1φ    }
           LIST1    }  CB1
           CONTR2   }
LIST1,     AV 4φ    }
           Fφ 4φ    }
           F1 φ5    }  PL1
           F2 2φ E  }
CONTR2,    SS 5φ    }
           LIST2    }  CB2
LIST2,     F1 15    }
           F2 3φ E  }  PL2
           PAUSE
```

Given that the data structure is held together with address pointers, this particular ordering of the CBs and PLs is not mandatory.

Two additional codes are necessary. Before the first

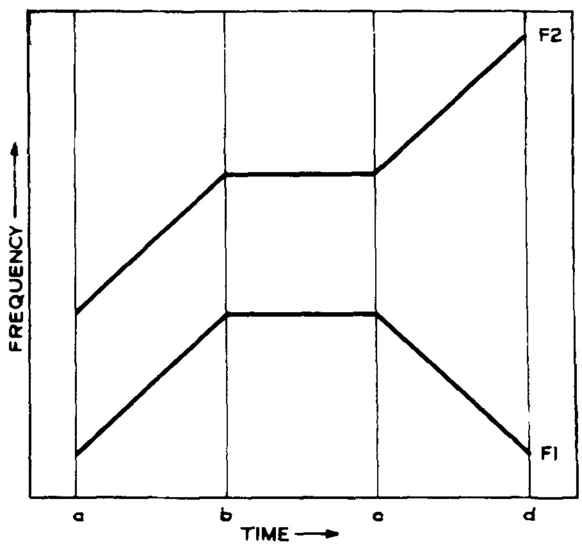

Figure 9. Schematic diagram of simplified /bag/.

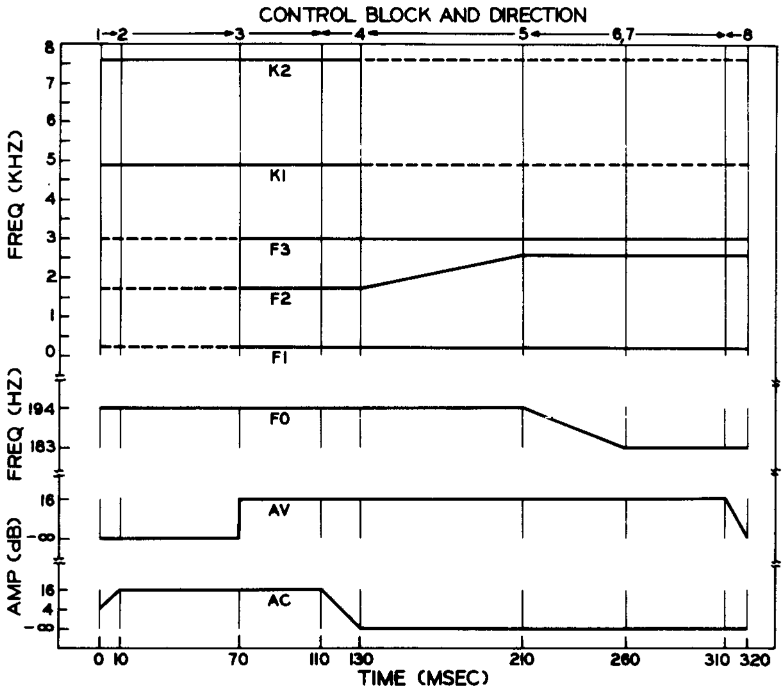

Figure 10. Schematic diagram of the syllable /zi/.

CB or PL, one must include a memory origin statement of the form: *1XXXX, where $0 \leqslant XXXX \leqslant 7377_8$.

This will cause the first CV or PL to be assembled at location 1XXXX. After the last CB or PL, one must include an end flag, simply: PAUSE. Comments may be added in the coding by preceding them with a /.

SYNTHESIZER PERFORMANCE

In this section, some comments on the performance of the synthesizer will be presented along with a more complex example of stimulus coding and synthesis. For an example, we will consider the synthesis of the syllable /zi/. A schematic diagram of the syllable is shown in Figure 10; the coding is shown in Figure 11, and a spectrogram of the resulting sound is shown in Figure 12.

In coding speech sounds, we have found that the most natural sounding results are obtained by gradually bringing up the amplitude at the beginning and gradually bringing it down at the end. In our example, the first 10 msec, controlled by CON1, and the last 10 msec, controlled by CON8, accomplish this purpose. Note that the frequencies for the vowel in /zi/ are specified in the first parameter list, PAR1, although they are not actually used until, at 70 msec, PAR3 sets AV to 16 dB. This is done because setting AV to a high value directly from $-\infty$ at the same time as setting the formant frequencies sometimes results in a distorted signal. In general, a certain amount of care must be taken whenever specifying rapid parameter transitions. Especially susceptible are the bandwidth controls. Bandwidth transitions which change several steps at a time will usually cause sharp transients (i.e., clicks) to occur in the output.

Another problem that we have had with the synthesizer is that of repeatability. Each time we synthesize a sound, we may not get exactly the same sound. This occurs because both the noise and voicing sources are free-running; if we start our speech sample at a different

```
.PAL1
*OUT-S:PAP
*
*IN-S:POUL,S:SSYM
*
*
*OPT-T

               / SAMPLE DATA FOR SPEECH SYNTHESIZER - /ZI/

                        *10200
0200  6102  CON1,  IF F 2   /INTERPOLATE FWD, SET PLOT FLAG, 10 MSEC
0201  0203         PAR1     /NAME OF PARAMETER LIST
0202  0213         CON2     /NAME OF NEXT CONTROL BLOCK
0203  2057  PAR1,  F0 57    /194 HZ
0204  6477         K1 77    /4935 HZ
0205  7066         K2 66    /7611 HZ
0206  3455         F3 55    /2934 HZ
0207  3055         F2 55    /1849 HZ
0210  2410         F1 10    /252 HZ
0211  6010         AC 10    /4 DB
0212  7510         AK 10 E  /4 DB, END OF LIST

0213  2014  CON2,  SS 14    /STEADY STATE, 60 MSEC
0214  0216         PAR2
0215  0217         CON3
0216  6140  PAR2,  AC 40 E  /16 DB...TRANSITION FROM 4DB,END OF LIST

0217  2010  CON3,  SS 10    /STEADY STATE, 40 MSEC
0220  0222         PAR3
0221  0223         CON4
0222  0540  PAR3,  AV 40 E  /16 DB... VOWEL COMES ON

0223  4004  CON4,  IB 4     /INTERPOLATE BACKWARDS, 20 MSEC
0224  0226         PAR4
0225  0227         CON5
0226  6100  PAR4,  AC 00 E  /TURN OFF FRICITIVE

0227  4020  CON5,  IB 20    /INTERPOLATE BACKWARDS, 80 MSEC
0230  0232         PAR5
0231  0233         CON6
0232  3166  PAR5,  F2 66 E  /2397 HZ..F2 TRANSITION FROM 1849 HZ

0233  4012  CON6,  IB 12    /INTERPOLATE BACKWARDS, 50 MSEC
0234  0236         PAR6
0235  0237         CON7
0236  2155  PAR6,  F0 55 E  /183 HZ... F0 TRANSITION FROM 194 HZ

0237  2012  CON7,  SS 12    /STEADY STATE, 50 MSEC
0240  0000         0        /NO PARAMETER LIST !!!
0241  0242         CON8

0242  4002  CON8,  IB 2     /INTERPOLATE BACKWARDS, 10 MESC
0243  0245         PAR8
0244  0000         0        /END OF DATA
0245  0500  PAR8,  AV 00 E  /TURN OFF VOWEL GRADUALLY

                   PAUSE
```

Figure 11. Data coding for the syllable /zi/.

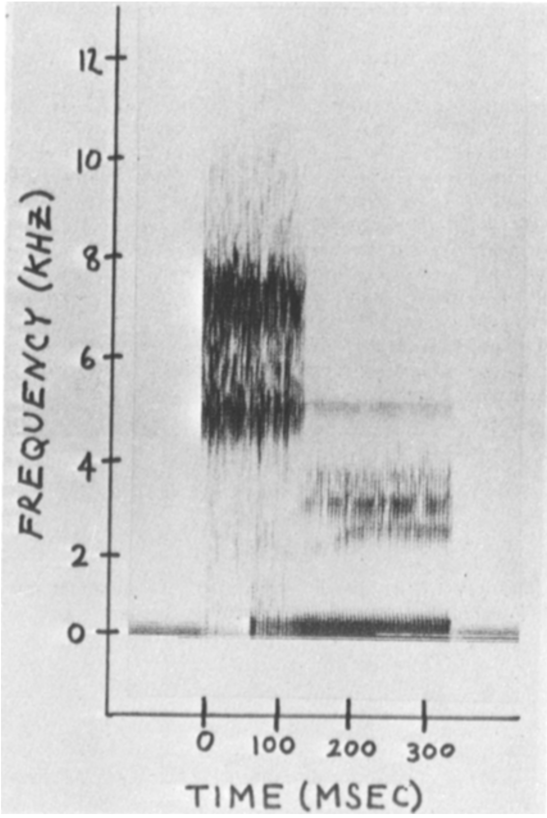

Figure 12. Sound spectrogram of /zi/.

time, the source may be intercepted at a different point. With the noise source, this is really not a problem. With the voice source, however, the difference is detectable. As a solution to this problem, we have installed circuitry to allow the computer to monitor the state of the F0 pulse. In the circuit, the F0 pulse is fed through a voltage follower to a Schmitt trigger which sets a flip-flop when the F0 pulse reaches a certain voltage in a positive direction. When a skip-on-F0 instruction is executed by the computer, the output of this flip-flop is gated to the PDP-8L skip bus. If the F0 flip-flop has been set, the skip will occur. If the skip fails, the program jumps back to test again, until it succeeds. By delaying initiation of synthesis until the rising edge of the F0 pulse is encountered, repeatability of the stimuli is insured.

USING THE PROGRAM

In order to use the synthesizer control subroutine from a main program, one uses the following code:

```
JMS I SPEAK
ARG1
ARG2
```

ARG1 may include one or both of the two commands, plot enable (PE) and clear (CL). When PE is specified, the spectrogram will be plotted. If CL is specified, certain tables within the program are cleared. If the tables are not cleared, one can use a SS or IB CB to continue or interpolate from the values set at the end of the last call to the subroutine. ARG2 is the memory address of the first CB of the sound to be synthesized.

The display routine plots axes and the center frequency of each formant on a Tektronix RM503 oscilloscope. Solid lines are plotted when a formant has an amplitude other than $-\infty$, unless it has a bandwidth larger than the minimum, in which case a dashed line is plotted. The horizontal axis of the display represents time in 100-msec increments, covering from 0 to 1,000 msec. The vertical axis represents frequency from 0 to 10 KHz. Setting the P flag in a CB will reset the display

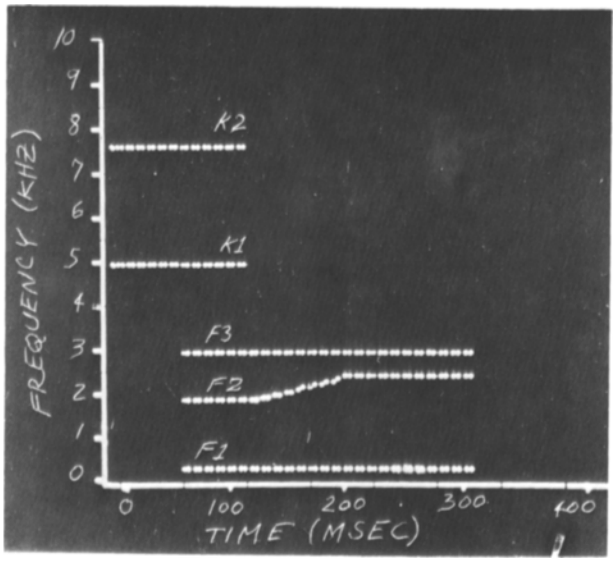

Figure 13. Computer-generated display of /zi/.

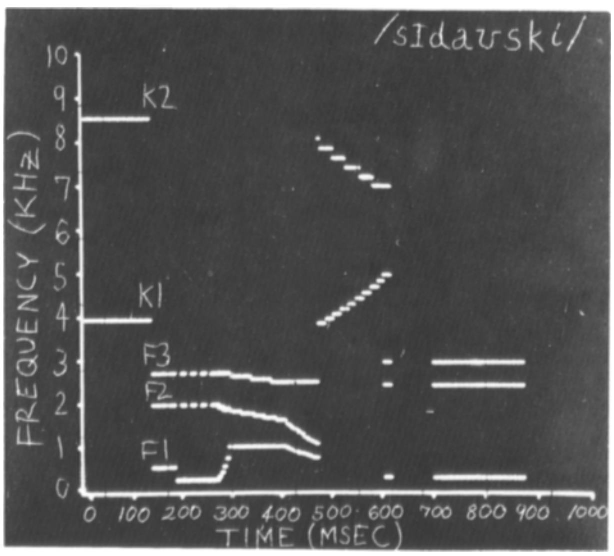

Figure 14. Computer-generated display of /sIdaʊski/.

to the 0 point of the time scale. The display routine will display the stimuli while the synthesizer produces it. This feature is quite useful for debugging stimuli during preparation. Figure 13 shows the scope display of the syllable /zi/ as programmed in Figures 10 and 11. Figure 14 shows the display of a slightly more complex example, /sIdaʊski/.

REFERENCE NOTE

1. Coker, C. H., Denes, P. B., & Pinson, E. N. *Speech synthesis: An experiment in electronic speech production.* Bell Telephone Laboratories, 1963.

REFERENCES

DUDLEY, H., & TARNOCZY, T. H. The speaking machine of Wolfgang von Kempelen. *Journal of the Acoustical Society of America*, 1950, **22**, 151-166.

FANT, G., & MARTONY, J. Instrumentation for parametric synthesis (OVE-II). *Quarterly progress report.* Speech Transmission Laboratory, Stockholm, July 1968. Pp. 18-24.

FLANAGAN, J. L. Note on the design of "terminal analog" speech synthesizers. *Journal of the Acoustical Society of America*, 1957, **29**, 306-310.

FLANAGAN, J. L. The synthesis of speech. *Scientific American*, 1972, **226**, 48-58.

HOLMES, J. N. *Speech synthesis.* London: Mills & Boon, 1972.

HOLMES, J. N., MATTINGLY, I. G., & SHEARME, J. N. Speech synthesis by rule. *Language and Speech*, 1964, **7**, 127-143.

LILJENCRANTS, J. C. W. A. The OVE-III speech synthesizer. *IEEE Transactions on Audio Electroacoustics*, 1968, **AU-16**, 137-140.

MATTINGLY, I. G. Synthesis by rule of general American English. *Status reports on speech research* (Suppl.). New York: Haskins Laboratories, 1968.

RAHIMI, M. A., & EULENBERG, J. B. A computer terminal with synthetic speech output. *Behavior Research Methods & Instrumentation*, 1974, **6**, 255-258.

TOMLINSON, I. G. SPASS—An improved terminal analog speech synthesizer. *Quarterly progress report.* MIT Research Lab of Electronics, Cambridge, Mass., Vol. 80, 1966.

NOTE

1. The OVE-IIId speech synthesizer is manufactured by A. B. Fonema, Box 1010, S-640 25, Julita, Sweden.