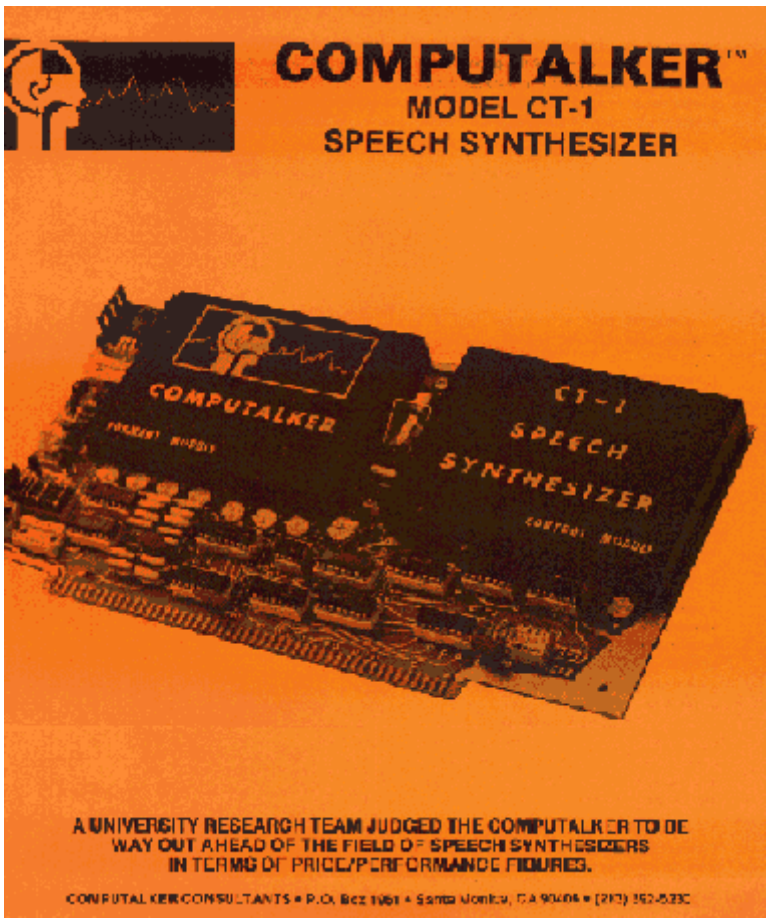
ph Computalker

COMPUTALKER CT-1 speech syntheser

dropbox
Computalker Sounds



The COMPUTALKER Model CT-1 Speech Synthesiser is a high quality voice generator unit designed for the standard S-100 I/O bus configuration. The synthesiser is controlled by acoustic-phonetic parameters transmitted on the microcomputer data bus. These parameters control the perceptually and physiologically fundamental aspects of speech as determined by contemporary phonetic research.
With the COMPUTALKER Model CT-1, sound are defined in real time under software control. Parameters which represent the phonetic structure of human speech are transmitted to the CT-1 at a rate of 500 to 900 bytes per second, depending on the data compression techniques used. This allows the production of highly intelligible and quite natural sounding speech output. Speaker characteristics and language or dialect variations are retained in the output.

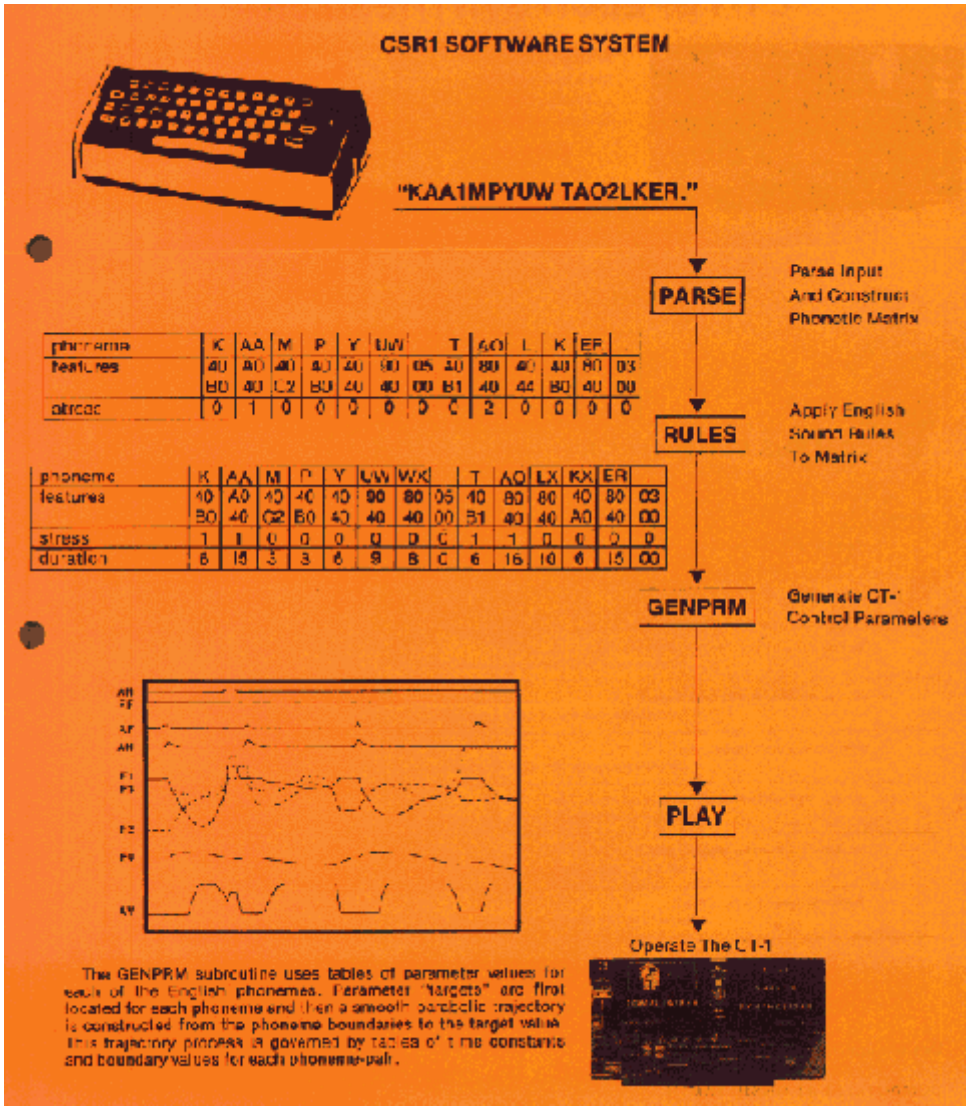
COMPUTALKER CT-1 Speech Synthesiser Hardware Specifications

Standard S-100 compatible board: 10x51/4 PC board with 100 pin (dual 50 .125 CTRS) edge connector pattern Depth, approx. 11/16 overal (occupies one slot on I/O board)
Components on board include: CT-1 Synthesiser module set (2 calibrated modules, ea. 3x4x51/8) 14 digital and analog IC's, Power regulators, Address selector switch 2 extra sockets for Expansion and External Parameter Control Bus interface: Uses 10 output addresses, one byte (8 bits) each Block of 10 addresses is relocatable to any hex boundary via on-board switch Remaining 6 ports in block of 16 are reserved for future use A parameter data frame consists of a sequence of 9 output instructions which update each of the 9 parameter values. After addressing any of the 9 ports, a minimum of 20 microseconds must be allowed before adressing another port.

Control Parameter	Mnemonic	Out addr
Voicing Amplitude	AV	xØ*
Voising Frequency	FØ	x1
Formant 1 Frequency	F1	x2
Formant 2 Frequency	F2	x3
Formant 3 Frequency	F3	x4
Aspiration Amplitude	AH	x5
Frication Amplitude	AF	x6
Frication Frequency	FF	x7
Nasal Amplitude	AN	x8
Audio On-Off Switch	SW	xF

*Ø ⩽ x ⩽ F is determined by a 4-bit selector switch on the CT-1 board.

Good quality speech requires a frame rate of approx. 100 frames per second. Updating at this frame rate, the 8080 CPU is occupied approx. 2 to 3 percent of the time. Connections on the PC board are provided for controlling the speech fundamental frequency (F0) from an external square wave source (such as an electronic music synthesiser) rather than the software controlled F0 parameter. This allows real-time control of the Compu-singer.
Audio output: RCA type phono jack mounted on PC board 1 V peak-to-peak output into 10K ohm load resistance
Power requirements: + 8 V 170 mA typ., 250 mA max. (on-board regulation to ca 5 V) ca 16 V at 85 mA (on-board regulation to ca 12 V)
The COMPUTALKER Model CT-1 can also be operated in a low data rate mode using phoneme definitions contained in the CSR1 Synthesis-by-Rule software package.



The COMPUTALKER speech synthesis system, used in this way, has the advantage that the software driver can easily be modified to keep the naturalness and intelligibility of the speech output up to date with the constantly evolving state of the art of rule governed speech.

Synthesizing speech by rule with the COMPUTALKER MODEL CT-1

Synthesis-by-Rule is a method of producing synthetic speech which is considerably easier than computer/hand analysis of recorded human speech. The word or phrase to be synthesized is entered in the form of a phonetic code to a software system which generates the control parameters for the CT-1 Synthesizer board. The result is speech which is understandable to most people in all but the most difficult perceptual situations with high noise levels or speech material having completely unexpected content.

The demonstration cassette contains a portion of the Gettysburg Address synthesized using a system of software rules. Such a set of software acousticphonetic rules is available from Computalker Consultants coded for the 8080 CPU. This software system accepts a string of ASCII coded phonetic symbols with stresses marked, and produces a set of control parameters for the Model CT-1 Synthesizer. The example on the cassette was generated using a previous version of this software system coded in FORTRAN, and running on a DEC PDP-12. As the parameter data was generated, it was punched on paper tape in the data format as described in the CT-1 Hardware User's Manual, and then read into the IMSAI 8080 for playback. That program, as run on the larger machine, was originally written for a different speech synthesizer and some parameters required special treatment for conversion to the CT-1 parameter format. In some cases, this conversion was not accurately fine-tuned for the CT-1, and the direct output of the 8080 version of the program is somewhat clearer in some of the fine details.

The CSR1 Synthesis-by-Rule software system is organized around the philosophy of attempting to produce natural sounding, human quality speech, rather than trying to produce a stereotypical robot-like sound. Because the true structure of real human speech is not yet correctly represented in the software rules, the resulting speech sometimes has an eerie quality that makes the listener try to assign human-like traits and qualities to the 'speaker" behind the voice. This psychological reaction to the voice does not occur when it is synthesized in a "robot" stereotype having little or no pitch variation and aupt, blocky formant frequency transitions. The pitch control parameter (F0) can easily be held to a constant value if the speech output sounds better to you that way. The CSR1 software system is structured around phonological, phonetic and acoustic principles in such a way that it can be modified to keep pace with the state of the art of synthesis of natural speech. The Model CT-1 has been designed a general acoustic synthesizer so that the hardware will not pose limitations to further improvements in the obtainable speech output quality.

The CSR1 software system is set up as a general callable suoutine which accepts a string argument containing the phonetic text, and on completion, plays the speech data in the buffer directly to the CT-1. With this structure, CSR1 may be called either from a keyboard input loop (supplied with the code) giving an on-line phonetic synthesizer, or from another system such as BASIC or an operating system, which passes a stored or computed string argument containing the material to be synthesized. On return, the buffer contains the actual CT-1 data as synthesized, which may be written out to cassette or paper tape for editing with the CTMON Monitor/Editor program. The 8080 assembly code version of CSR1 fits in less than 6K bytes of memory, including all phoneme feature and target tables. This code may be located in ROM or RAM. Additional RAM will be required for parameter data storage during the actual synthesis. The buffer space required is 300 bytes per second of speech. By comparison, the introductory phrase, "Hello, I'm Computalker, A speech synthesizer designed to plug into the standard bus on your 8080 microcomputer" is less than 7K bytes long. CSR1 version 1.0 completes the coniputation of parameter data before beginning playback. An interrupt driven

version is currently under development, which will begin playback as soon as sufficient data has been computed and stored in the buffer.

How to get natural sounding speech output from the COMPUTALKER MODEL CT-1

The demonstration cassette, "Sounds of Couputalker",illustrates several methods of obtaining the control parameters to operate the Computalker Model CT-1 Speech Synthesizer. High quality speech output, as exemplified by the introductory phrases, "Hello, I'm Computalker. A speech synthesizer ... ", involves computer processing of recorded human speech followed by a fair amount of hand work. The recordings were initially digitized at 10K samples/second and then analysed using a linear prediction algorithm to extract the formant frequencies, and a cepstrum algorithm to measure the fundamental frequency. These techniques are described in several texts on speech analysis (Flanagan,J.L., Speech Analysis, Synthesis, and Perception, 2nd Ed., Springer Verlag 1972; Markel,J.D. and Gray,A.H.,Jr., Linear Prediction of Speech, Springer '1erlag 1976). In addition to these analyses, the amplitude was measured by RMS averaging a smooth window each 10 msec. to obtain the AV parameter. Some editing of the formant frequency data was done by hand to eliminate falsely detected peaks and fill in occasional gaps in the true formant data before converting the frequency data to the Computalker parameters Fl, F2, and F3. Since the CT-1 control parameters consist of numerical values within the range of 0-255, all frequency and amplitude data is converted so that it stays within this range. All the above steps required approximately 6 hours of time on a DEC PDP-12 set up for speech analysis processing to produce the original data for the introductory phrases on the cassette. At this stage, this data was punched on paper tape and then read into the CT-l Control Monitor program running on my IMSAI 8080. From that point, I spent several more evenings entering the datd for parameters AH, AF, FF, and AN, and a bit more touching up of the other parameters.

Given the frequency vs. time information obtained from the initial computer analysis, the remaining aspiration and frication data can be inserted by fairly straight-forward procedures. These procedures will be described in the completed CT-1 Hardware User's Manual. The Manual will also discuss the approximate formant frequency patterns needed to construct the sounds of the various phonemes of English. It would be feasible (although tedious work) to construct intelligible sounds by hand editing based on this data. However, it is still quite difficult to form these patterns to make natural sounding speech without access to a spectrum analysis process of some kind. Such an analysis gives you the frequency structure as a function of time, i.e. retaining the natural timing structure.

It is my plan to publish more extensive descriptions of the above mentioned speech analysis techniques, to make them accessible to a wider audience than they now have. The recent developments in floating point hardware with multiplication In the 50-100 microsec. range make it reasonable to do this sort of analysis on a microcomputer. The setup would require a filter and A/D converter capable of sampling the speech at at least 10K samples/sec. The low-pass speech filter ahead of the A/D converter should be reasonably flat to at least 1/3 of the sampling rate, and then down by at least 30-40 db at 1/2 the sampling rate. 32K of RAM memory would allow sampling up to 3 seconds continuously which is a workable sized chunk. Without floating point hardware the analysis would proceed quite slowly but in many cases that is not a drawback on a micro system.

Alternatively, for a modest consulting fee, Computalker Consultants could supply the basic, rough formant frequency, Fill and AV data from your tape recording, leaving out the aspiration, frication and nasal values, which must be added by hand. As a preliminary estimate, I believe this work could be done for approx. $25 per second of speech material to be analyzed. Working from this basic data the desired speech could be produced following the tables and information given in the CT-1 Hardware User's Manual, using the CTMON Monitor/Editor to synthesize speech from the data as the work progresses.

Friends, Humans and Countryrobots: Lend me your Ears
by D Lloyd Rice published Byte in aug 76

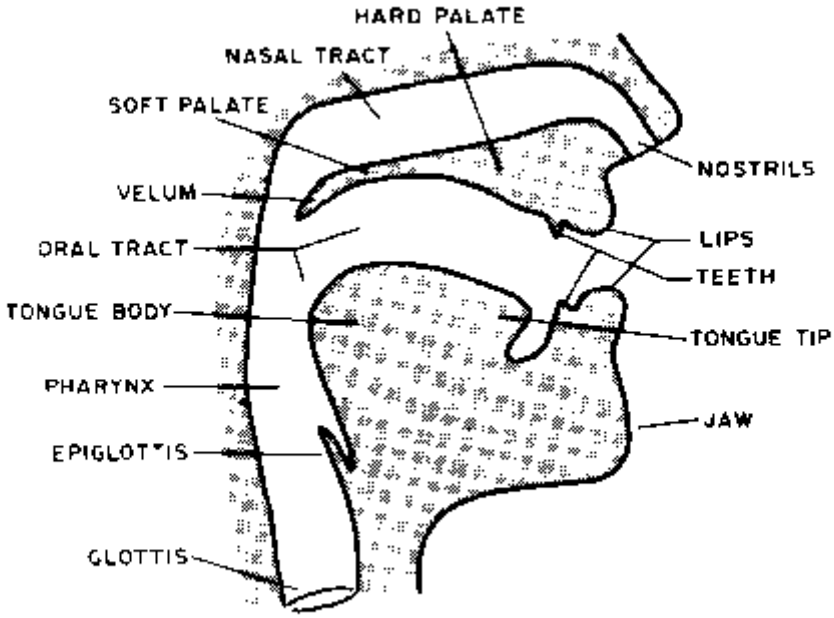


Figure 1: The Human Vocal Tract. The human vocal tract is roughly described as a tube approximately 17.4 cm long with varying resonance characteristics as muscles control the shape. The tract splits into two parts, nasal and oral, at the top, with a valve called the velum providing flexible control of the nasal resonances in given utterance. An electronic model of this natural organ roughly parallels the function of the tract.

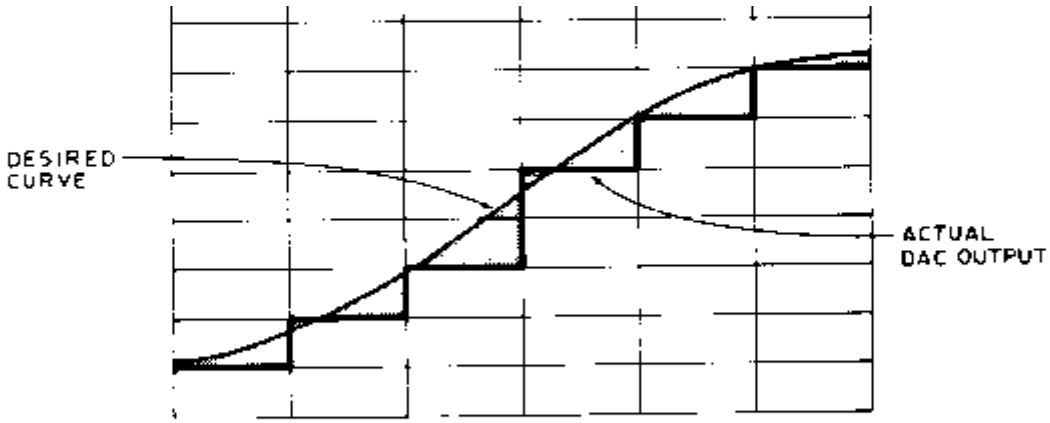


Figure 2: DAC Quantization Errors. The actual output of a computer to the analog world is a step function (in the absence of any filtering). This leads to the problem of quantization errors, depicted conceptually here by the shaded areas in between the smooth analog function and its closest step function approximation. Low precision digital to analog conversions accentuate this problem.

You've got your microcomputer running and you invite your friends in to show off the new toy. You ask Charlie to sit down and type in his name. When he does, a loudspeaker on the shelf booms out a hearty Hello, Charlie! Charlie then starts a game of Star Trek and as he warps around thru the galaxy searching for invaders, each alarming new development is announced by the ship's computer in a warning voice, Shield power low!, Torpedo damage on lower decks! The device that makes this possible is a peripheral with truly unlimited applications, the speech synthesizer. This article describes what a speech synthesizer is like, how it works and a general outline of how to control it with a microcomputer. We will look at the structure of human speech and see how that structure can be generated by a computer controlled device. How can you generate speech sounds artificially, under computer control? Let's look at some of the alternatives. Simplest of all, with a fast enough digilal to analog converter (DAC) you can generate any sound you like. A 7 or 8 bit DAC can produce good quality sound, while some-where around 4 or 5 bits the quantization noise starts to be bothersome. This noise is produced because with a 5 bit data value it is possible to represent only 32 discrete steps or voltage levels at the converted analog output. Instead of a smoothly rising voltage slope, you would get a series of steps as in figure 2.

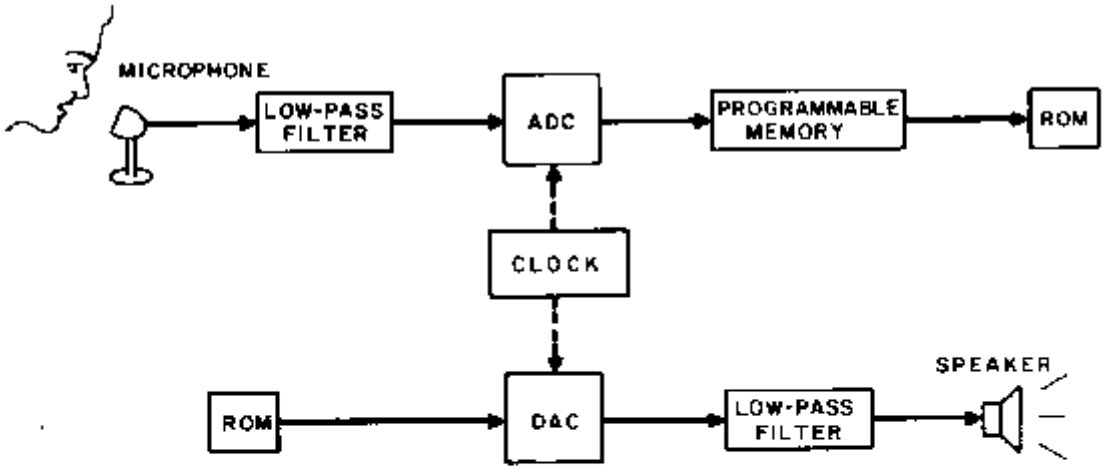


Figure 3: Waveform Playback from ROM Storage. One way to achieve a digitally controlled vocal output is to first digitize a passage of human speech, then store the digital pattern in memory. For a commercial product, such as a talking calculator, the limited vocabulary required makes this a feasible avenue of design, especially when a single mass produced ROM can be used in the final product. In an experimenter's system, the ROM is not needed, and programmable memory can be substituted during experiments. This is probably the least expensive way to augment an existing computer's capability with vocal output, but the memory requirements limit its use to small vocabularies. The quality of the result varies with the ADC (and DAC) sampling rate and precision.

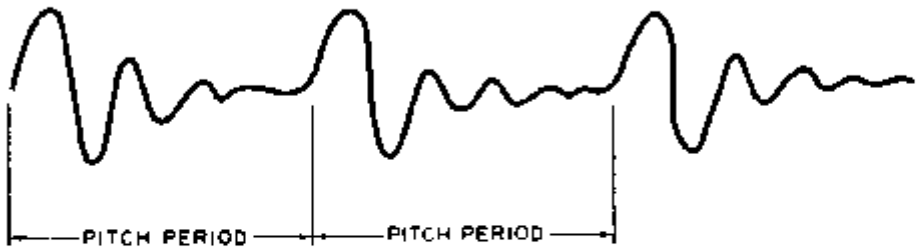


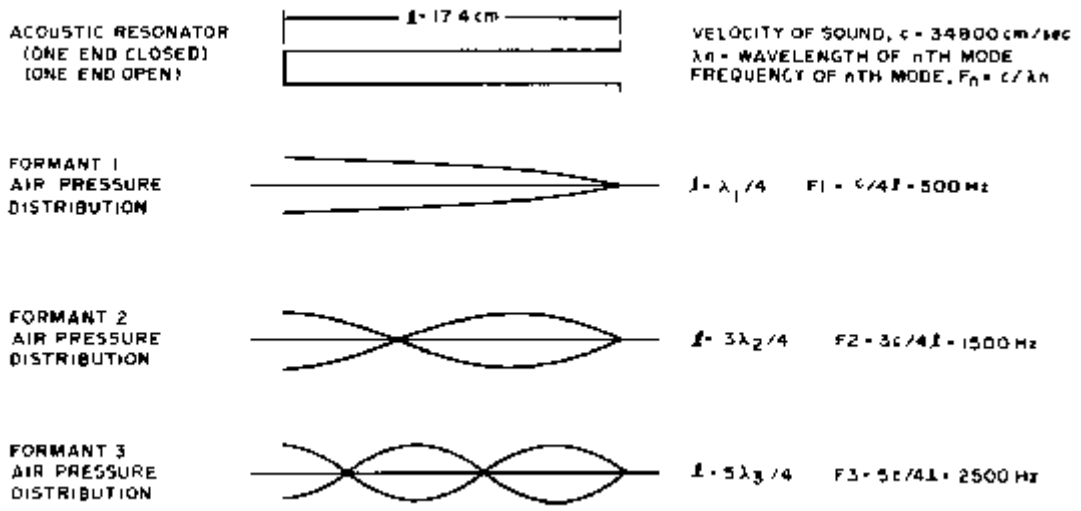
Figure 4: Typical Vowel Waveform. In principle, a vowel is a fairly long sustained passage of sound with repetitive characteristics. The vowel sounds are produced physiologically by the resonances of the vocal tract, and are controlled electronically by the formant filters which produce the equivalent of vocal tract resonances.

As for the speed of the DAC, a conversion rate of 8,000 to 10,000 conversions per second [The sample rate in conversions per second or samples per second is often quoted in units of Hertz. We will use that terminology here, although conversions per second is a generalization of the concept of cycles per secondi is sufficient for

fairly good quality speech. With sample rates below about 6 kHz the speech quality begins to deteriorate badly because of inadequate frequency response. Almost any microprocessor can easily handle the data rates described above to keep the DAC going. The next question is, where do the samples come from? One way td get them would be by sampling a real speech signal with a matching analog to digital converter (ADC) running at the same sample rate. You then have a complicated and expensive, but very flexible, recording system. Each second of speech requires 8 K to 10 K bytes of storage. If you want only a few words or short phrases, you could store the samples on a ROM or two and dump then sequentially to the DAC. Such a system appears in figure 3. If you want more than a second or two of speech output, however, the amount of ROM storage required quickly becomes impractical. What can be done to minimize storage? Many words appear to have parts that could be recombined in different ways to make other words. Could a lot of memory be saved this way? A given vowel sound normally consists of several repetitions of nearly identical waveform segments with the period of repetition corresponding to the speech fundamental frequency or pitch. Figure 4 shows such a waveform.
Within limits, an acceptable sound is produced if we store only one such cycle and construct the vowel sound by repeating this waveform cycle for the duration of the desired vowel. Of course, the pitch will be precisely constant over that entire interval. This will sound rather unnatural, especially for longer vowel durations, because the period of repetition in a naturally spoken vowel is never precisely constant, but fluctuates slightly. In natural speech the pitch is nearly always changing, whether drifting slowly or sweeping rapidly to a new level. It is of interest that this jitter and movement of the pitch rate has a direct effect on the perception of speech because of the harmonic structure of the speech signal. In fact, accurate and realistic modelling of the natural pitch structure is probably the one most important ingredient of good quality synthetic speech. In order to have smooth pitch changes across whole sentences, the number of separate stored waveform cycles still gets unreasonable very quickly. From these observations of the cyclic nature of vowels, let us move in for a closer look at the structure of the speech signal and explore more sophisticated possibilities for generating synthetic speech.

How Do We Talk?

The human vocal tract consists of an air filled tube about 16 to 18cm long, together with several connected structures which make the air in the tube respond in different ways (see figure 1). The tube begins at the vocal cords or glottis, where the flow of air up from the lungs is broken up into a series of sharp pulses of air by the vibration of the vocal cords. Each time the glottis snaps shut, ending the driving pulse with a rapidly falling edge, the air in the tube above vibrates or rings for a few thousandths of a second. The glottis then opens and the airflow starts again, setting up conditions for the next cycle. The length of this vibrating air column is the distance from the closed glottis up along the length of the tongue and ending at the lips, where the air vibrations are coupled to the surrounding air. lf we now consider the frequency response of such a column of air, we see that it vibrates in several modes or resonant frequencies corresponding to different multiples of the acoustic quarter wavelength. There is a strong resonance or energy peak at a frequency such that the length of the tube is one quarter wavelength, another energy peak where the tube is three quarter wavelengths, and so on at every odd multiple of the quarter wavelength. If a tube 17.4 cm long had a constant diameter from bottom to top, these resonant energy peaks would have frequencies of 500 Hz, 1500 Hz, 2500 Hz and so on. These resonant energy peaks are known as the formant frequencies. Figure 5 illustrates the simple acoustic resonator and related physical equations.



The vocal tract tube, however, does not have a constant diameter from one end to the other. Since the tube does not have constant shape, the resonances are not fixed at 1 000 Hz intervals as described above, but can be swept higher or lower according to the shape. When you move your tongue down to say ah, as in figure 6, the back part is pushed back toward the walls of the throat

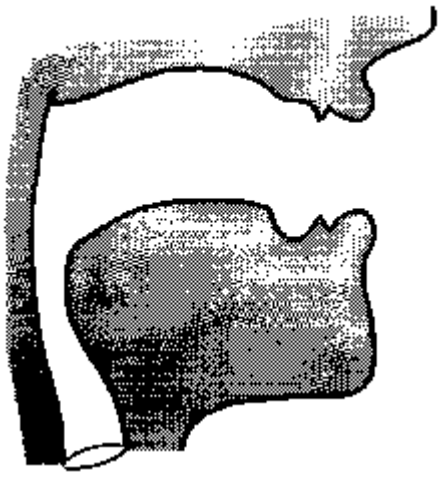


Figure 6: "ah" as in "father." In figure 1, the vocal tract was shown in schematic form. Here is a similar figure showing how the tract has been modified to produce the vowel sound "ah." The human typically closes off the nasal cavity and widens out the oral cavity by opening the mouth during this sound.

and in the front part of the mout the size of the opening is increased. Th effect of changing the shape of the tube this way is to raise the frequency of the fir' resonance or formant 1 (Fl) by sever hundred Hz, while the frequency of formant 2 (F2) is lowered slightly. On the other hand, if you move your tongue forward ar upward to say ee, as in figure 7,

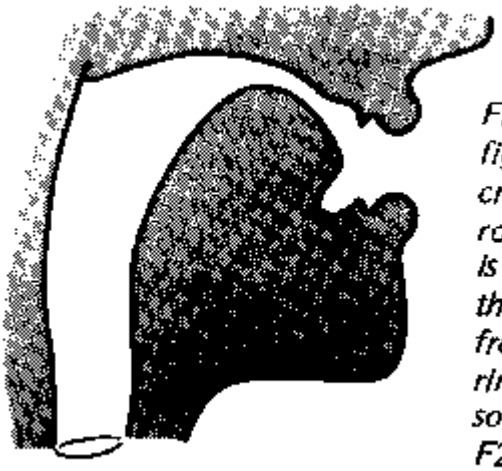


Figure 7: "ee" as in "heed." In contrast to figure 6, when the "ee" vowel sound is created, the mouth opening tends to be narrowed; and the upper end of the vocal tract is restricted. This lowers the frequency of the first resonant mode and raises the frequencies of the second and third. Referring to table 1, the "ee" vowel sound has some of the highest resonances for formants F2 and F3 and the lowest for F1.

Figure 8: Voiced Sounds from the Glottis. Sounds which have definite pitch are called voiced sounds. In the natural larynx, these sounds are generated by the vocal chords and drive the vocal tract at the glottis. In an electronic analog, the voiced sounds can be generated by a programmable counter (to set the frequency) which in turn creates a sine wave of the same frequency. A rectified sine wave is a good source for the glottal pulses used in the electronic model of a larynx used in the author's approach to speech generation.

the size of the tube at the front, just behind the teeth, is much smaller, while at the back the tongue has been pulled away from the walls of the throat, leaving a large resonant cavity in that region. This results in a sharp drop Fl down to as low as 200 or 250 Hz, with F2 being increased to as much as 2200 or 2300 Hz. We now have enough information to put together the circuit for the oral tract branch of a basic formant frequency synthesizer. After discussing that circuit, we will continue on in this way, describing additional properties of the speech mechanism building up the remaining branches of synthesizer circuit.

A Speech Synthesizer Circuit

To start with, we must have a train of driving pulses, known as the voicing source, which represents the pulses of air flowing up thru the vibrating glottis. This could be simply a rectified sine wave as in figure 8. To get different voice qualities, the circuit may be modified to generate different waveform shapes. This glottal pulse is then fed to a sequence of resonators which represent the formant frequency resonances of the vocal tract. These could be simple operational amplifier bandpass filters which are tunable over the range of each respective formant. Figure 9 shows the concept of a typical resonator circuit which meets our requirements. 1C1, 1C2 and 1C4 form the actual bandpass filter, while 1C3 acts as a digitally controlled resistance element serving to vary the resonant frequency of the filter.

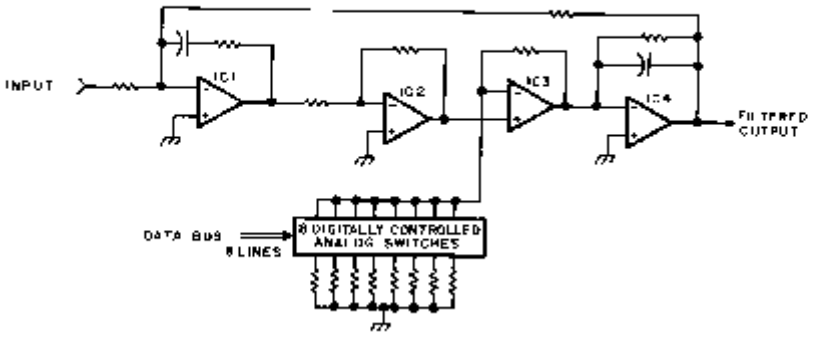


Figure 9: Typical Formant Resonator Circuit. A digitally controlled band pass filter can be built from four operational amplifiers and 8 digitally controlled analog switches. The filter characteristics are set by the choice of the resistance and capacitance elements as well as the digital control word. The operational amplifier IC3 serves as a gain controlled amplifier in the feedback loop, which alters the filter resonance.

Several such resonator circuits are then combined as in figure 10 to form the vocal tract simulator. The voicing amplitude control, AV, is another digitally controlled resistance similar to 1C3 of figure 9. This gain controlled amplifier configuration is the means by which the digital computer achieves its control of speech signal elements. The data of one byte drives the switches to set the gain level of the amplifier in question. In figures 10, 13 and 1 5 of this article, this same variable resistance under digital control is shown symbolically as a resistor with a parameter name, rather than as an operational ampliefier with analog switches.

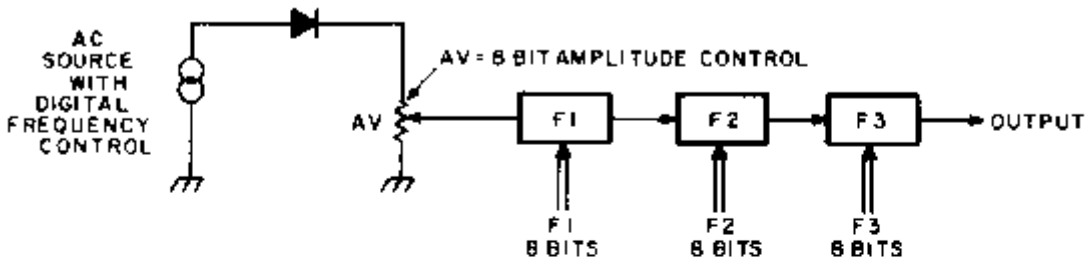


Figure 10: A first approximation of the voice synthesizer can be constructed by using three formant filters in series with differing resonance settings all controlled by 8 bit digital words. The resistance indicated as AV is an operational amplifier circuit (see IC3 of figure 9) with a digital gain control input. It is thus a programmable element of gain less than unity, in other words the electronically controlled equivalent of a variable resistance. This notation of a controlled resistance is used in figures 13 and 15 as well.

Generating Vowel Sounds

The vocal tract circuit as shown thus far is sufficient to generate any vowel sound in any human language (no porpoise talk, yet). Most of the vowels of American English can be produced by fixed, steady state formant frequencies as given in table 1.

	F1	F2	F3
heed	250	2300	3000
hid	375	2150	2800
head	550	1950	2600
had	700	1800	2550
hod	775	1100	2500
paw	575	900	2450
hood	425	1000	2400
who	275	850	2400

Table 1: Steady State English Vowels. The vowel sounds are made by adjusting the formant resonances of the human vocal tract to the frequencies listed in this table. These figures are approximate, and actual formant resonances vary from individual to individual. In a speech synthesizer based upon an electronic model of the vocal tract, the formant frequencies are set digitally using operational amplifier filters with adjustable resonant peaks.

A common word is given to clearly identify each vowel. The formant frequency values shown here may occasionally be modified by adjacent consonants. An alternative way to describe the formant relationships among the vowels is by plotting formant frequencies Fl vs F2 as in figure 11. F3 is not shown here because it varies only slightly for all vowels (except those with very high F2, where it is some what higher).

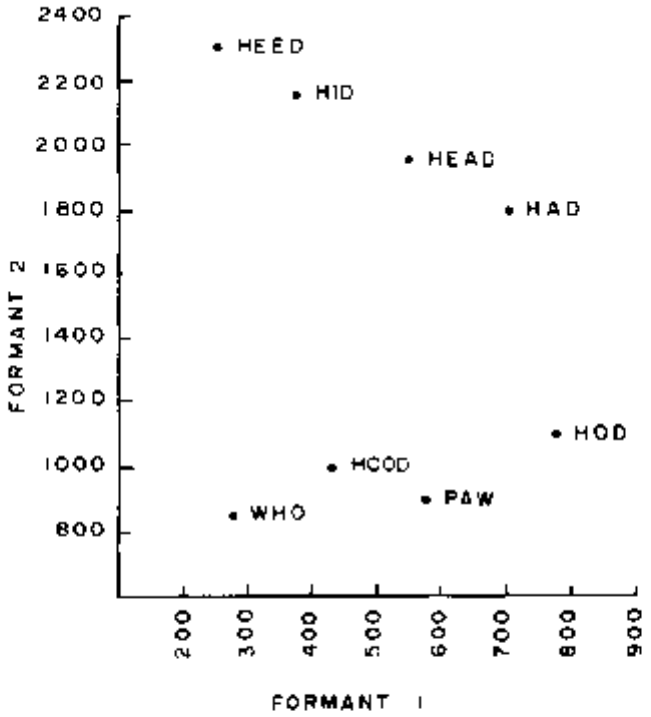


Figure 11: The Steady State English Vowels. The distinctions between various vowel sounds can be illustrated by plotting them on a two dimensional graph. The horizontal axis is the formant 1 frequency, the vertical axis is the formant 2 frequency. A location for each vowel utterance can be determined experimentally by locating the resonance peaks with an audio spectrum analyzer.

The F1-F2 plot provides a convenient space in which to study the effects of different dialects and different languages. For example, in some sections of the United States, the vowels in hod and paw are pronounced the same, just above and to the right of paw on the graph. Also, many people from the western states pronounce the sounds in head and hid alike, about halfway between the two points plotted for these vowels on the graph. A few English vowels are characterized by rapid sweeps across the formant frequency space rather than the relatively stable positions of those given in table 1. These sweeps are produced by moving the tongue rapidly from one position to another during the production of that vowel sound. Approximate traces of the frequency sweeps of formants F1 and F2 are shown in figure 12 for the vowels in bay, boy, buy, hoe and how. These sweeps occur in 150 to 250 ms roughly depending on the speaking rate.

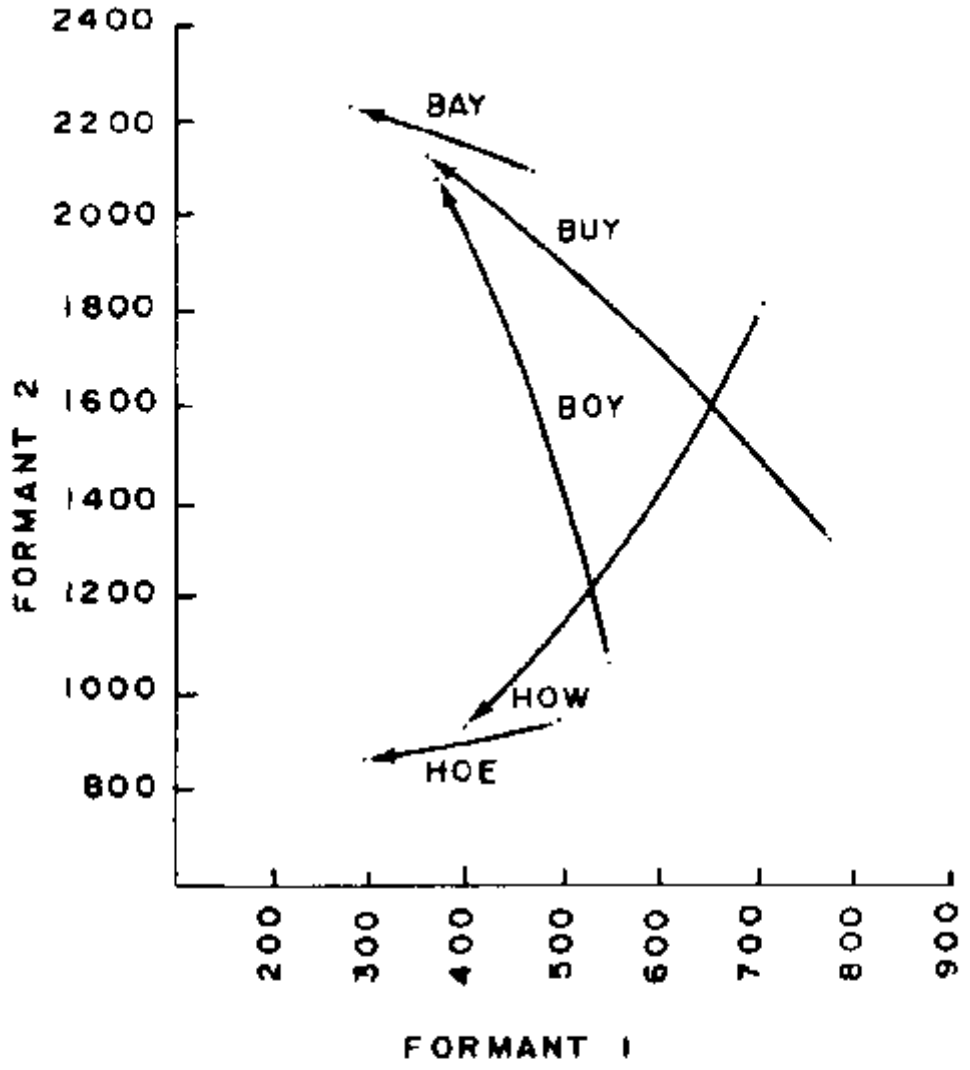


Figure 12: English Diphthongs. A diphthong is a sound which represents a smooth transition from one vowel sound to another during an utterance. The time duration of the swap from one point to another in formant space is typically 150 to 250 ms. This graph shows typical starting and ending points for several common diphthong sounds.

Consonant Sounds

Consonant sounds consist mostly of various pops, hisses and interruptions imposed on the vibrating column of air by the actions of several components of the vocal tract shown in figure 1. We will divide them into four classes: 1) stops, 2) liquids, 3) nasals, and 4) fricatives and affricates. Considering first the basic 'stop consonants, p, t, k, b, d and g, the air stream is closed off, or stopped, momentarily at some

point along its length, either at the lips, by the tongue tip just behind the teeth or by the tongue body touching the soft palate near the velum. Stopping the air flow briefly has the effect of producing a short period of silence or near silence, followed by a pulse of noise as the burst of air rushes out of the narrow opening. The shape of the vocal tract with the narrow opening at different points determines the spectral shape of the noise pulse as well as the formant locations when voicing is started. Both the noise burst spectrum and the rapid sweeps of formant frequency as the F1-F2 point moves into position for the following vowel are perceived as characteristic cues to the location of the tongue as the stop closure is released. We need only add a digitally controlled noise generator to the vocal tract circuit of figure 10 to simulate the noise of the burst of air at the closure release and we can then generate all the stop consonants as well as the vowels.

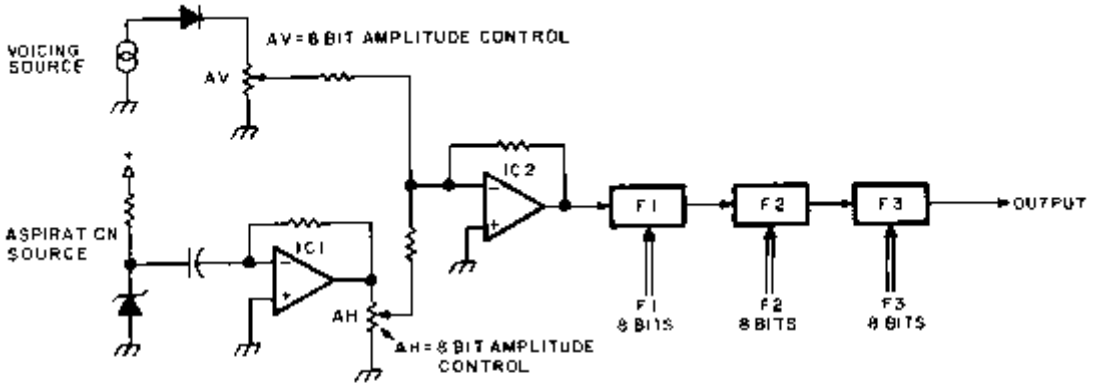


Figure 13: Synthesizer with Aspiration Noise Generator. Not all utterances are vowels. By adding a digitally controlled noise generator to the circuit of figure 10, it is possible to synthesize the consonant sounds known as "stops." In this circuit, the amplitude versus time characteristics of the noise pulse are determined by an 8 bit programmable gain control AH (shown symbolically as a resistor). The output of the noise source is mixed with the voicing source with the analog sum being routed to the formant filters. The noise generator is a zener diode.

Figure 13 shows the speech synthesizer with such a noise generator added. The breakdown noise of a zener diode is amplified by lC1 and amplitude is set by the digitally controlled resistor AH. 1C2 is a mixer amplifier which combines the glottal source and aspiration noise at the input to the formant resonators. It is important to notice at this point the range of different sounds that can be generated by small changes in the relative timing of the control parameters. The most useful of these timing details is the relationship between the pulse of aspiration noise and a sharp increase in the amplitude of voicing (see figure 14).

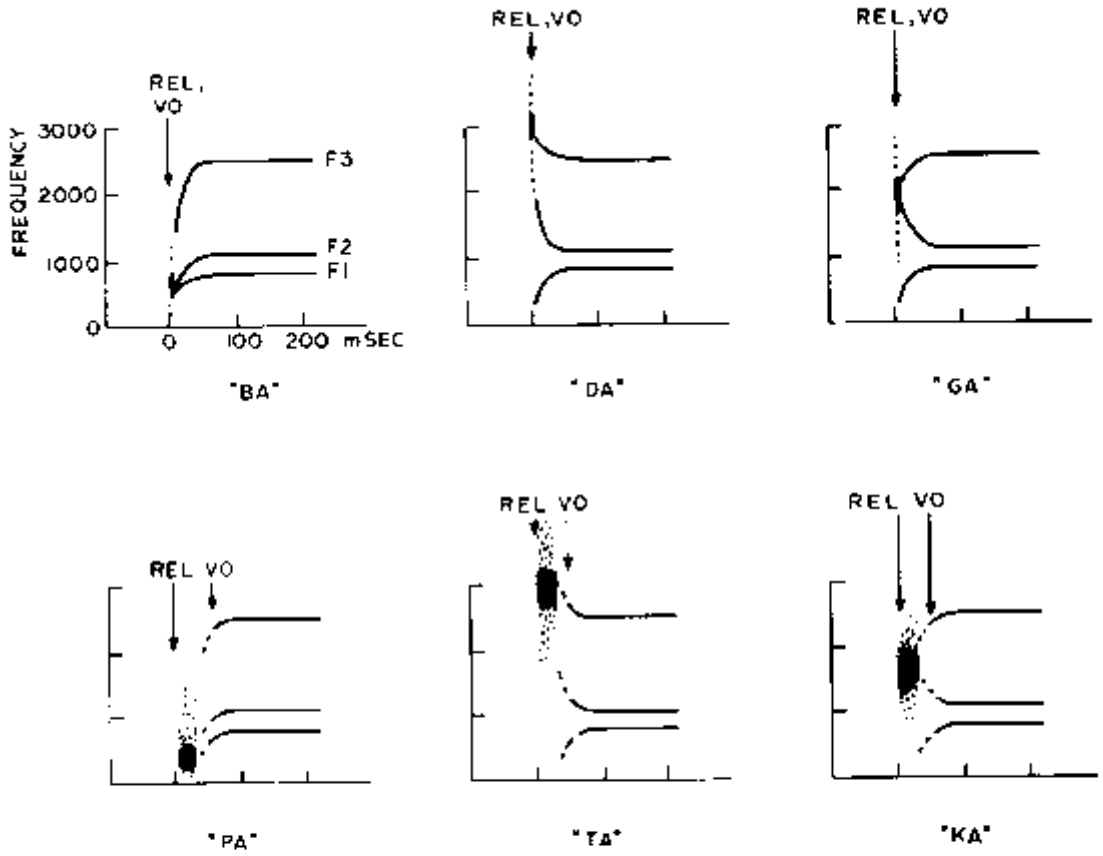


Figure 14: Stop Consonant Patterns. This figure illustrates 6 different stop consonant patterns. The release of the stop closure (start of noise pulse) is at the point marked by "Rel" and the beginning of the voicing sounds is marked by "VO". Note the typical transition of the vowel formants as the steady state is reached.

For example, if we set the noise generator to come on for a noise pulse about 40 ms long and immediately after this pulse, Fl sweeps rapidly from 300 up to 775 Hz and F2 moves from 2000 down to 1100 Hz, the sound generated will correspond to moving the tip of the tongue down rapidly from the roof of the mouth. Observe, however, that the formant output is silent after the noise pulse until the voicing amplitude is turned up. If voicing is turned on before or during a short noise burst, the circuit generates the sound da, whereas if the voicing comes on later, after a longer burst and during the formant frequency sweeps, the output sounds like ta. This same timing distinction characterizes the sounds ba vs pa and ga vs ka, as well as several other pairs which we will explore later. Figure 14 gives the formant frequency patterns needed to produce all the stop consonants when followed by the vowel ah.When the consonant is followed by a different vowel, the formants must move to different positions corresponding to that vowel. The important thing to note about a stop transition is that the starting points of the frequency sweeps correspond to the point of closure in the vocal tract, even though these sweeps may be partially silent for the unvoiced stops p, t and k, where the voicing amplitude comes on after the sweep has begun. The second consonant group comprises the liquids, w, y, and I. These sounds are actually more like vowels than any of the other consonants except that the timing of formant movements is crucial to the liquid quality. W and y can be associated with the vowels oo and ee, respectively. The difference is one of timing. If the vowel oo is immediately followed by the vowel ah, and then the rate of Fl and F2 transitions is increased, the result will sound like wa.

A comparison of the resulting traces of Fl and F2 vs time in wa with the transition pattern for ba in figure 14 points out a further similarity. The direction of movement is basically the same, only the rate of transition of ba is still faster than for wa. Thus we see the parallelism in the acoustic signal due to the common factor of lip closeness in the three sounds ua, wa and ba. Y can be compared with the vowel ee in the same way, so the difference between ia and ya is only a matter of transition rates. Generally, I is marked by a brief increase of F3, while r is indicated by a sharp drop in F3, in many cases, almost to the level of F2.
The third group of consonants consists of the nasals, m, n and ng. These are very similar to the related voiced stops b, d and g, respectively, except for the addition of a fixed nasal formant. This extra formant is most easily generated by an additional resonator tuned to approximately 1400 Hz and having a fairly wide bandwidth. It is only necessary to control the amplitude of this extra resonator during the closure period to achieve the nasal quality in the synthesizer output. The fourth series of consonants to be described are the fricatives, s, sh, zh, z, f, v and th and the related affricates ch and j. The affricates ch and j consist of the patterns for t and d followed immediately by the fricative sh or zh, respectively, that is, ch = t+sh and j = d+zh. The sound zh is otherwise rare in English. An example occurs in the word azure. With the letters th, two different sounds are represented, as contained in the words then and thin. All the fricatives are characterized by a pulse of high frequency noise lasting from 50 to 150 msec. The first subdassification of fricatives is according to voicing amplitude during the noise pulse, just as previously described for the stop consonants. Thus, s, sh, f, ch and th as in thin' have no voicing during the noise pulse, while z, zh, v, j and th as in then have high voice amplitude. When a voiceless fricative is followed by a vowel, the voicing comes on during the formant sweeps to the vowel position, just as in the case of the voiceless stops. The different fricatives within each voice group are distinguished by the spectral characteristics of the fricative noise pulse. This noise signal differs from that previously described for the stop bursts in that it does not go thru the formant resonators, but is mixed directly into the output after spectral shaping by a single. pole filter.

	Resonator Frequency (FF)	Fricative Amplitude (AF)
sh, zh	2500	.9
s, z	5000	.7
f, v	6500	.4
th	8000	.2

Table 2: Fricative Spectra. A fricative sound typically consists of a pulse of high frequency noise. The various types of fricatives are classified according to the spectral profile of the pulse. For the electronic model described here, the fricative amplitude and resonator frequency for several sounds are listed in this table.

Table 2 gives the fricative resonator settings needed to produce the various fricative and aifricate consonants. Fricative noise amplitude settings are shown on a scale of 0 to 1.

The system level diagram of a complete synthesizer for voice outputs is summarized in figure 15.

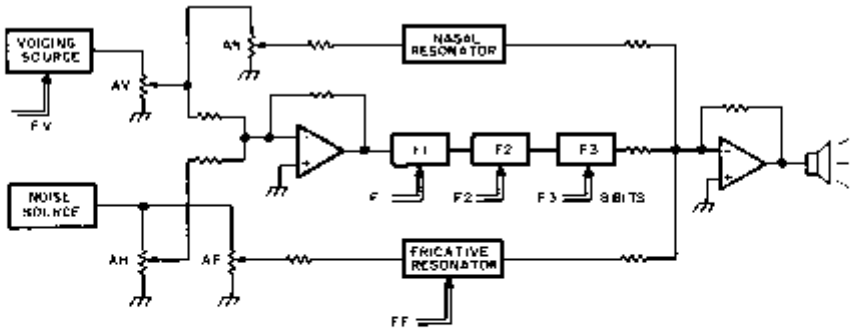


Figure 15: The Complete Synthesizer. This diagram shows the organization of a complete synthesizer which includes a wide variety of parameters. The voicing frequency and amplitude are set by parameters FV and AV. The noise pulses of stop consonants are generated with the programmable gain element AH. The fricative resonator with amplitude AF and frequency resonance FF are used to generate fricatives like "s" and "sh." The normal vowel sounds are generated by control of the formant frequencies F1, F2 and F3, and a nasal resonator with amplitude AN and fixed frequency characteristics is used to add varying amounts of nasal sounds. The result of signals processed through the nasal, formant and fricative paths is summed by a final operational amplifier and used to drive the output speaker.

The information contained in this article should be sufficiently complete for individual readers to begin experimenting with the circuitry needed to produce speech outputs. In constructing a synthesizer on this model, the result will be a device which is controlled in real time by the following parameters: AV amplitude of the voicing source, 8 bits FV frequency of the voicing source, 8 bits AH amplitude of the aspiration noise component, 8 bits AN amplitude of the nasal resonator component, 8 bits AF amplitude of the fricative noise component, 8 bits Fl frequency of the formant 1 filter, 8 bit setting. F2 frequency of the formant 2 filter, 8 bit setting. F3 frequency of the formant 3 filter, 8 bit setting. FF frequency of fricative resonator filter, 8 bit setting. This is the basic hardware of a system to synthesize sound; in order to complete the system, a set of detailed time series for settings for these parameters must be determined (by a combination of the theory in this article and references, plus experiment with the hardware). Then, software must be written for your own computer to present the right time series of settings for each sound you want to produce. Commercial synthesizers often come with a predefined set of phonemes which are accessed by an appropriate binary code. The problem of creating and documenting such a set of phonemes is beyond the scope of this introductory article, but is well within the dollar and time budgets of an experimenter.

Product Information

At the time this article goes to press, a synthesizer module incorporating several detail refinements and improvements over the circuits of this article is being developed by the author and associates. A detailed user's guide will be supplied with the Computalker module which illustrates the timing relationships needed to produce all the consonant-vowel and vowel-consonant combinations which occur in natural speech. This can serve as a reference guide for creating your speech output software which generates the proper control patterns from text inputs. Write to Computalker, 821 Pacific St No. 4, Santa Monica CA 90405 for the latest information on this module.

BIBLIOGRAPHY

1. Erman, Lee, ed, IEEE Symposium on Speech Recognition, April, 1974, Contributed Papers, IEEE Catalog No. 74CH0878-9 AE.

2. Flanagan, J L, and Rabiner, L R, eds, *Speech Synthesis*, Benchmark Papers in Acoustics, Dowden, Hutchinson & Ross, Inc, 1973.

3. Lehiste, Ilse, ed, *Readings in Acoustic Phonetics*, MIT Press, 1967.

4. Moschytz, George S, *Linear Integrated Networks Design*, Van Nostrand, New York, 1975.