24

Sound-Generation by means of

With the advent of modern automatic, high speed digital computers, it has become possible to generate a very large class of musical sounds and other acoustic signals with a degree of precision that is difficult to achieve with ordinary electronic (analog) devices. The computer program described in this article was originally conceived and executed by M. V. Mathews, of the Bell Telephone Laboratories, in January, 1958.*1 Since that time, the program has been considerably revised and expanded, so that it now constitutes an extremely flexible musical medium, capable of producing virtually any sound or sequence of sounds that can be described or specified numerically. Note that this program is designed specifically for the utilization of the computer as a sound producing medium, not as a means of composing or analyzing music.

a Digital Computer

JAMES C. TENNEY

The advantages of such a medium should be obvious to any composer, but the above statement of its capabilities defines not only the potentialities of the computer as a musical instrument, but a limitation as well, since it is often very difficult to give an accurate objective description of an imagined or intended sound, even when that sound is quite familiar to the ear. This limitation will become less severe, however, as time gives us more experience with the medium, and with the new modes of specification required.

Another obvious advantage in using the computer to generate musical sounds is that the process is automatic. That is, an entire composition can be recorded on tape with no splicing, editing, timing, mixing, etc. In fact, no manual operation of

any kind is required of the composer. His "score" (in the form of a deck of punched cards) must go through several stages of transformation, using certain special devices in addition to the computer, but the composer is not directly involved in these intermediate operations. The final result is an ordinary magnetic tape recording of the sounds and sound sequences described in the score.

Here again, however, one of the advantages of the technique implies a corresponding disadvantage. Automation brings with it a certain indirection – a time delay (of from a few hours to a few days) between the composer's decisions or experimental actions and any auditable results. The standard tape and electronic music techniques thus have a definite advantage over the computer medium, with respect to their immediacy, and the greater ease with which unfamiliar sounds may be tried and tested before actually being used in a composition. Still, the time-delay with the computer is far shorter than that usually involved in the realization of music for conventional instrumental groups, so that this need not be considered a very serious limitation. There are other problems that arise in using the computer as a sound source, but these are of a more technical nature, and will be more easily defined later on, when the details of the process have been explained.

The "stages of transformation" of the composer's score, mentioned above, are of three kinds. First, the numbers on the cards (which specify the various parameters of each sound – i.e., its duration, amplitude, frequency, waveform, etc.) must be transformed into a new set of numbers, which define the successive instantaneous amplitudes of the sound waves themselves. Second, this new set of numbers must be transformed into an electrical signal, and recorded on standard magnetic tape. Third, this tape recording is converted to sound by means of ordinary playback equipment. How these transformations are achieved will be the subject of the first section of this article, though the descriptions will necessarily be brief.

In addition to these, however, there is a fourth stage involved, which in practice comes before the others. This is the transformation from musical conception to numerical specification, that must be carried out by the composer before the other stages can even begin. And this is not simply a matter of notation, though that is also involved. The composer must make decisions that are not required when using conventional musical instruments, and many that are not required in any other

tape or electronic music technique. It will be seen, for example, that he must design his own (symbolic) "instruments" before writing the score, and, although this does not involve any very specialized knowledge of acoustics or electronics (or even of computer programming), it does involve a number of compositional decisions that are not necessary in any other medium. These problems will be dealt with in the second section, and in much greater detail than the more mechanical aspects of the process, since these are the problems that will most directly concern the composer.

The Three Stages of Transformation from Numbers to Sound

The digital computer is essentially a device which performs the ordinary arithmetic operations of addition, subtraction, multiplication and division, but at a very high speed, and automatically — which is to say that it can perform a whole sequence of these operations without any human intermediation. Both the input and the output of the digital computer consist exclusively of numbers in some form, but at the input these numbers may represent not only magnitudes of some quantity involved in a mathematical computation; they may also represent — by means of an appropriate code — instructions to the machine.

The numbers that are fed into the computer thus represent two kinds of information — the numbers that are to be used in the computations, called the data, and a set of coded instructions, called the program, which defines the arithmetic operations to be carried out on the data. In the case of the music-generating process, the data or score includes a deck of punched cards (called "note-cards"), which give the values of the various parameters of each sound to be generated. Figure 1 shows one of these note-cards, with numbers specifying the duration, intensity and frequency of the tone to be generated.

The program contains a set of instructions which cause the computer to transform the numbers in the score into a new sequence of numbers, which represent the instantaneous amplitudes of the corresponding acoustic signals at successive, equally spaced intervals in time. These instantaneous amplitude-values are called samples, and this way of representing a sound wave (or any function of time) in terms of successive amplitude-values is called sampling. This process is illustrated in Figure 2.

Now the variations in air pressure, particle velocity or particle displacement that constitute a sound wave are continuous

28

figure

1 & 2

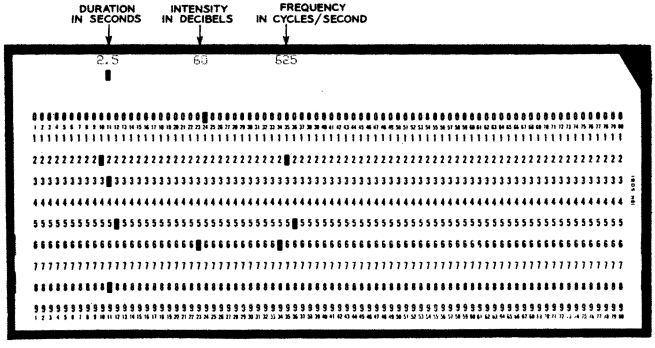
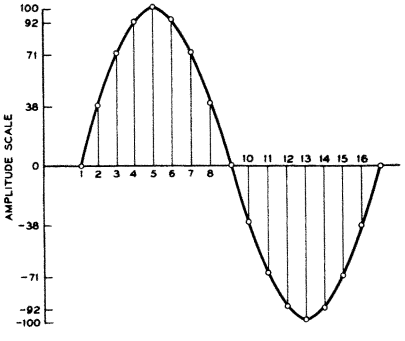


FIGURE 1
A TYPICAL NOTE-CARD



SAMPLE NUMBER	AMPLITUDE
1	0
2	38
3	71
4	92
5	100
6	92
7	71
8	38
9	0
10	-38
11	-71
12	-92
13	-100
14	-92
15	-71
16	-38
ETC.	

FIGURE 2
SAMPLING A SINE-WAVE

functions of time, whereas the digital computer can only deal with discrete (i.e., noncontinuous) numbers. It has been found, however, that a continuous function of time such as an acoustic signal can be specified completely (i.e., without "distortion") by just such a sequence of discrete amplitude samples – provided only that the time interval between successive samples is short enough – i.e., that the sampling rate is fast enough – and the sampling rate that is necessary depends simply upon the bandwidth of the signal. More specifically, to adequately specify a sound wave containing frequency components no higher than some upper limit, F. cycles per second, a sampling rate of at least 2F samples per second is required. Thus, sounds with frequency components below 5000 cycles per second would have to be sampled at the rate of 10,000 samples per second, while a signal bandwidth from zero to 10,000 cycles per second would require a sampling rate of 20,000 samples per second, in order for the signal to be specified completely. This relation between sampling rate and bandwidth (or the highest frequency component) to be produced is evident intuitively, if we observe with reference to Figure 2, that the accuracy with which a signal will be represented by sampling increases as the rate of sampling is increased. That is, any variations in the instantaneous amplitude of the signal that occur within the period between successive samples will necessarily be lost in the process – and it is these smaller variations that correspond to the higher frequency components in the signal.

To say that a sound wave of bandwidth F requires 2F samples per second for its "specification" means that it can (theoretically) be recorded or stored in symbolic form using 2F numbers for each second of sound, and further, that the signal could be transmitted, in digital form, and reconstructed by a receiving device that can process these numbers at the same rate of 2F per second. The sample values that are calculated by the computer are recorded at the output on a special-purpose digital magnetic tape, in the form of on-off patterns of magnetic pulses – and thus in binary (rather than in decimal) form – as illustrated in Figure 3.

Many aspects of this first stage of transformation involve details of computer operation and programming which are beyond the scope of this article, and would be of little interest to a musician anyway. It may be said, however, that this transformation is achieved by way of a sequence of arithmetic operations that could, in principle, be carried out by a human being, though this would scarcely be practical. The sample

values are calculated one at a time, and the sequence of arithmetic operations defined by the program must be performed at least once for each sample on the output tape. At 10,000 samples per second, this involves several million operations for a few minutes of sound. Obviously, it is only because of the enormous speed at which a modern digital computer can perform these calculations that the sampling process becomes a relatively practical way of generating sound.

At this stage in the process, no sound has yet been produced, but simply a numerical description of a sound wave. The information on the output tape must next be converted from digital to "analog" form. That is, the numbers on the digital tape must be made to determine a continuous (nonimpulsive) pattern of magnetization on ordinary recording tape. This is accomplished by a special digital-to-analog conversion device called the data-translator*2. By means of the data-translator, each number on the digital tape is transformed into an electrical pulse, whose amplitude (voltage) is proportional to the magnitude of that number. The sequence of pulses is then put through a lowpass filter (with its upper cut-off frequency set at half the sampling rate), which removes the discontinuities in the signal. The smoothed signal is then recorded on standard magnetic ("analog") tape, and this, finally, is converted to sound by means of ordinary playback equipment and a loudspeaker. This conversion from digital-to-analog form is illustrated in Figure 4. It should be noted that no sound is actually generated by the computer itself. It is the data-translator (in conjunction with ordinary tape playback equipment) that generates the sound, so that when I refer here to "sound-generation by means of the computer," it is really the whole process that is involved, including the digital-to-analog conversion equipment. Figure 5 shows in schematic form the three stages of transformation described above.

Designing the Instruments

The music program consists of several smaller sets of instructions (sub-programs or subroutines) which perform various specific functions in the overall process, such as input and output control, storage allocation, certain kinds of numerical conversions, etc. The set of instructions that is specifically responsible for the transformation from parametric values to sample amplitudes has come to be called the orchestra—by analogy with conventional musical terminology. The orchestra is made up of one or more instruments (again, by analogy only), and these, in turn, consist of several smaller

figure

3 & 4

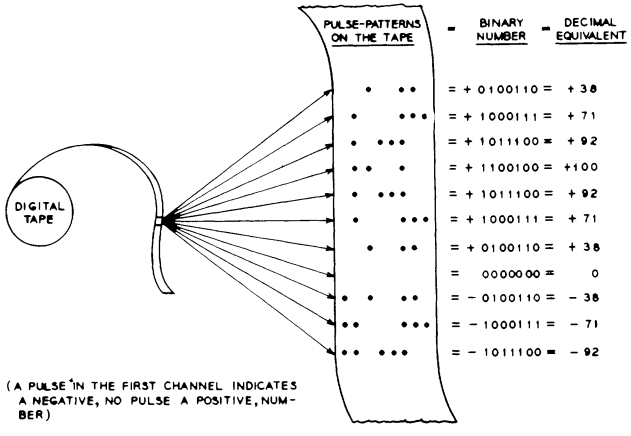


FIGURE 3
REPRESENTATION OF SAMPLE NUMBERS ON DIGITAL TAPE

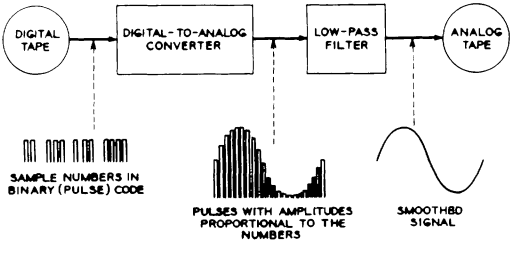


FIGURE 4
CONVERSION FROM DIGITAL TO ANALOG FORM

figure

5

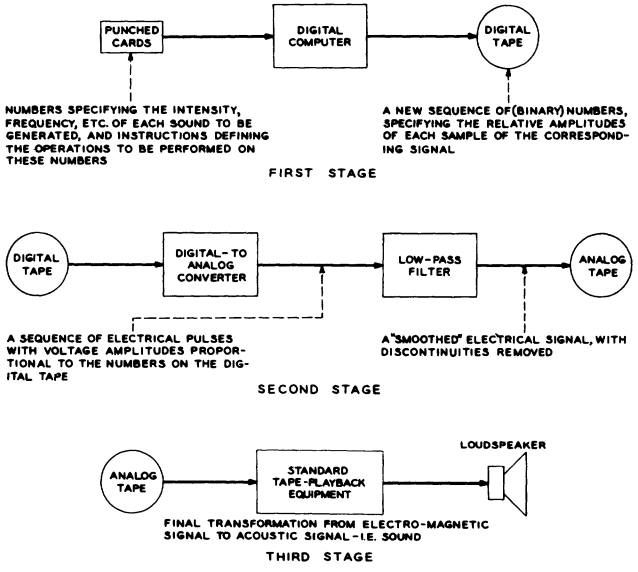


FIGURE 5
THE THREE STAGES OF TRANSFORMATION

elements called unit generators. Each of the above designations (i.e., orchestra, instrument, unit generator) thus represents a component sub-set of instructions, which may be combined in various ways with other sub-sets to make up the next larger group of instructions. This means that the composer may design his own instruments – and assemble his own orchestra – to suit the particular requirements of a composition. How the instruments are to be designed is thus one of the first questions the composer must answer when he begins to program a composition using the computer, and an understanding of this aspect of the process is necessary for an effective utilization of the medium.

Computer programming is generally done with the aid of block-diagrams or "flow-charts," which show in graphic form the flow of control in the machine from one operation to the next, determined by the sequence of instructions in the program. In designing an instrument, symbols are used which characterize each of the several different types of unit generators that may be combined to form an instrument. In Figure 6 the simplest type of instrument is shown in block-diagram form, along with the letters and numbers that are used to designate these particular unit generators. The symbols 1u1 and 1u2 designate the instrument number (#1) and the number of the unit generator (i.e., its order of sequence in that instrument), while G2 and G1 specify the particular type of generator employed. The symbol G2 designates a quasi-periodic function generator or oscillator. The generator marked G1 is an output unit, a type which must be included in every instrument, and must always be the last in the sequence of unit generators constituting an instrument. In terms of the actual operations in the computer, the G1 unit represents an instruction which says, essentially, add ϕ (the output sample) to the final acoustic output (which may already contain samples generated by other instruments, being "played" simultaneously).

The arrows in the diagram indicate input and output connections. In this instrument, the two inputs to the G2 unit are fixed for each note, and will be specified by the appropriate parameters on the note-cards. The output of 1u1 forms the input to 1u2; whenever two unit generators are connected in this way, it means that the results of the operations defined by one group of instructions (i.e., by one unit generator) will be used in the computations determined by the next group.

The M and I inputs to the oscillator determine, respectively, the amplitude and the frequency (or the duration of each period)

34

figure

6 & 7

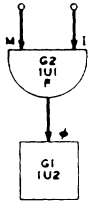
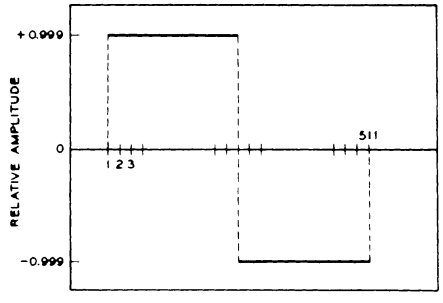


FIGURE 6
A SIMPLE INSTRUMENT



POSITIONS IN COMPUTER MEMORY	STORED VALUES OF THE FUNCTION
1	+ 0.999
2	+ 0.999
3	+ 0.999
⋮	⋮
255	+ 0.999
256	- 0.999
257	- 0.999
⋮	⋮
510	- 0.999
511	- 0.999

FIGURE 7
REPRESENTATION OF A SQUARE-WAVE FUNCTION
IN THE COMPUTER MEMORY

of the signal to be generated. In addition to these two inputs, the G2 unit has a function number, F associated with it, which determines the waveform of the output signal. This function may either be fixed — as designated in the original description of the instrument in the program — or it may be varied from note to note, in which case it is given as one of the parameters on the note-cards. The way the G2 oscillator generates output samples on the basis of these three numbers (the amplitude input, M; the frequency input, I; and the function number, F) may be described in a general way as follows.

By means of one of a set of function-generating subroutines, which will be described later, a series of numbers is stored in the computer memory (this information is put into the computer just ahead of the note-cards) which represent successive sample values of a function (this may be a waveform, or, as will be seen later, an amplitude- or frequency-envelope). Up to twenty different functions can be stored by the computer in any one run, and each of these twenty functions may take on as many as 512 independent amplitude-values, ranging from minus .999... to plus .999... As a simple example, Figure 7 shows a square-wave function, both graphically and as it would be represented in the computer.

Once these numbers have been stored in the computer, the sequence of operations runs something like this: (1) The value of the function that is located in position 1 of the memory is selected, and this number is multiplied by the number given by the M input (M for "multiplier"); it is this operation which determines the amplitude of the output signal. (2) The product thus obtained goes to the output unit (G1), and from there, eventually, to the output tape. (3) The next value of the function to be selected is determined by the number given by the I input; if I = 1, the number will be taken from position 2; if I = 10, or 100, it will be taken from position 11 or 101, etc. The I input (I for "increment") thus determines how many places in the computer memory will be skipped — in selecting the successive samples of the function — from one cycle to the next in the generating process, and thus determines how many values of the function will be sampled for each period of the output waveform. Since the samples of the output tape will always be "read" (by the digital-to-analog converter) at a constant rate — (usually 10,000 samples per second), the number of sample values that are used out of the 512 values that are actually available will determine the duration of each period, and thus the frequency of the output signal.

For example, if I = 1, each of the 512 stored values of the function being taken once, every period of the output signal will contain 512 samples, and will last 512/10,000 = .0512 seconds. The frequency of the signal will be the reciprocal of this, which is 10,000/512, or about 20 cycles per second. If, on the other hand, I = 16, (so that the generating routine selects every 16th value of the 512 that are stored for the function), each period of the output signal would contain 512/16 = 32 samples, and would last 32/10,000 = .0032 seconds, which, repeated periodically, results in a frequency of 10,000/32, or about 300 cps. When the value of I is less than 1 (i.e., for a frequency of less than 20 cps, and thus for all durations greater than 1/20th of a second), the generating routine will select the same sample value several times in succession, until the sum of all previous increments reaches the next larger integer, at which point the sample value in the next position is selected. This summing of increments is treated "modulo 512;" that is, when the sum reaches the value 512, it is reset to zero, and the sampling begins again at the beginning of the sequence of stored values of the function.

The above description of the G2 oscillator can be summarized by saying that it produces, at its output, a sequence of sample numbers, ϕ_i, according to the following relations.

$$\phi_i = M_i . F_j([S_i] \bmod 512), \tag{1}$$

And, for the summing of increments described above,

$$S_{i+1} = S_i + I_i \tag{2}$$

where i is an index specifying sample sequence, starting at zero at the beginning of each note, and stopping at the value determined by the total duration of the note (in terms of number of samples). The frequency, f, of the periodic signal at the output will be

$$f = I/.0512 \text{ cps} \tag{3}$$

and its amplitude will be proportional to M.

This number 512 ($=2^9$) arises from the fact that there are just this many positions ("words") in each storage location that can be used to store the sample values of any function. It is thus one of the fixed limits of the machine, but it does not result in any very serious limitations to the composer in practice. Another such numerical constant associated with the machine is the limit on the magnitude of the numbers that can be stored in

each position, (or recorded on the digital tape). The sample values at the output must lie within the range from -2047 to +2047, ($\cong -2^{11}$ to $+2^{12}$) which means that the number of distinct amplitude levels used in the sampling process is 4094. This results in a signal to quantizing-noise ratio of approximately 60 db.

In the simple instrument shown in the figure above, both the M and I inputs to the oscillator are constant for each note, specified by the appropriate parameters on the note-cards. But such inputs to a unit generator may be made to vary continuously with time, by letting them be determined by the (variable) outputs of other unit generators, as in Figure 8, where a second oscillator has been added, to modulate the envelope of the signal. This instrument would thus use two stored functions, one determining the envelope of the sound, the other its waveform, as in the first instrument. When an oscillator is used in this way to control the envelope of a signal, the I input is usually set equal to the reciprocal of the note duration, so that just one period of the function is generated per note.

A third type of unit generator that is used very often in even the simpler instruments is the mixer or adder, whose output is the sum of its inputs. At present, there are adders with two, three and four inputs, called G3, G4 and G5, respectively. A typical use of a two-input adder, G3, is illustrated in Figure 9, where one input (A1) is added to another (A2), which is taken from the output of an oscillator (3u2). With appropriate settings of the M and I inputs to 3u2, the result will be a periodic modulation of the I input (and thus of the frequency) of the signal generated by 3u4, i.e., a vibrato around a center frequency given by A1.

It was noted above that each input to a unit generator may either be specified by one of the parameters on the note-cards, or be taken from the output of another generator. There is a third possibility, and this is that an input may be defined as a constant for the instrument. This is done when a parameter does not need to be varied from note to note, as might sometimes be the case for such parameters as vibrato rate or range, for example. By this means, the number of parameters that have to be specified on the note-cards is reduced.

Other Unit Generators

Several other kinds of unit generators are also available for building instruments. These include a special oscillator (G2F),

38

figure

8 & 9

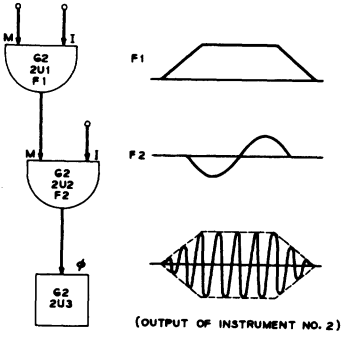


FIGURE 8
AN INSTRUMENT WITH ENVELOPE CONTROL

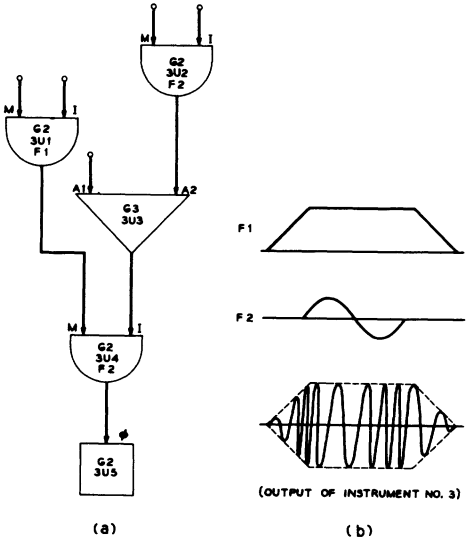
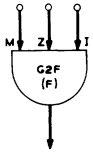


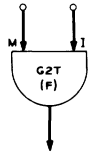
FIGURE 9
AN INSTRUMENT WITH ENVELOPE CONTROL AND VIBRATO

figure

10 & 11

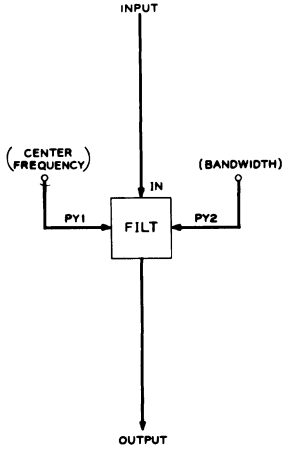


(a) THE G2F UNIT GENERATOR

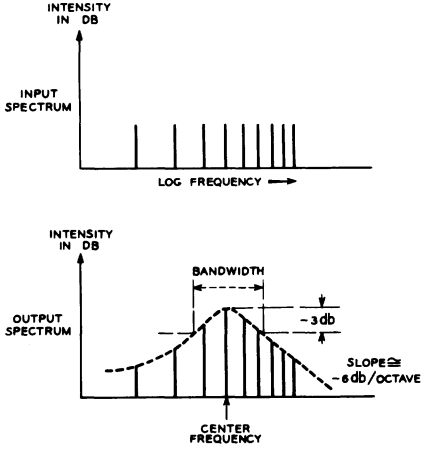


(b) THE G2T UNIT GENERATOR

FIGURE 10



(a) THE BANDPASS OR FORMANT FILTER



(b) TYPICAL INPUT AND OUTPUT SPECTRA FOR THE FILTER

FIGURE 11

which is like the ordinary G2 oscillator, except that it has a third input (Z), by means of which the function number may be varied within a single note, and a nonperiodic function generator (G2T), which may be used to modify the output of a previous generator according to some conversion function (to change, for example, a sequence of numbers from a linear to a logarithmic scale, etc.). The G2F and G2T unit generators are shown in Figures 10a and 10b. In addition to these, there is a bandpass or formant filter (FILT), which can be used to shape the spectrum of a compound tone, and two different kinds of random number or noise generators (called RAND and RANDX), which will be described in greater detail below.

The formant filter is shown in Figure 11a with a plot of typical input and output spectra in Figure 11b. The signal to be filtered is taken from the output of an oscillator, and enters the filter at the input marked IN. PY1 and PY2 control, respectively, the center frequency and the bandwidth of the passband or formant. As shown in Figure 11b, the bandwidth of formant is defined as the distance, on the frequency axis, between the two points in the output spectral envelope that are 3 db below the peak, on either side of the center frequency.

If two or more formant peaks are required, a corresponding number of these filters may be connected in series, as shown in Figure 12. As with all the other unit generators, the control inputs to the filter (i.e., PY1 and PY2) may either be (1) specified as constants for the instrument, (2) made to vary from note to note (in which case the input settings would be specified on the note cards), or (3) be taken from the output of some other unit generator, and thus be continuously variable within a single note.

The Random Number Generators

The two types of random number or noise generators listed above are alike to the extent that they both put out a sequence of independent random numbers at a rate determined by an I input (just as with the oscillators), and within a range limited by the value of an M input (also corresponding to the oscillators). But the final forms of their output signals differ. One, RAND produces an essentially linear interpolation between successive independent numbers whereas RANDX repeats each of the random numbers until a new one is computed. Thus, given the same sequence of random numbers, and the same settings for their M and I inputs, the output signals from each of these two generators would look quite different. This can

figure

12

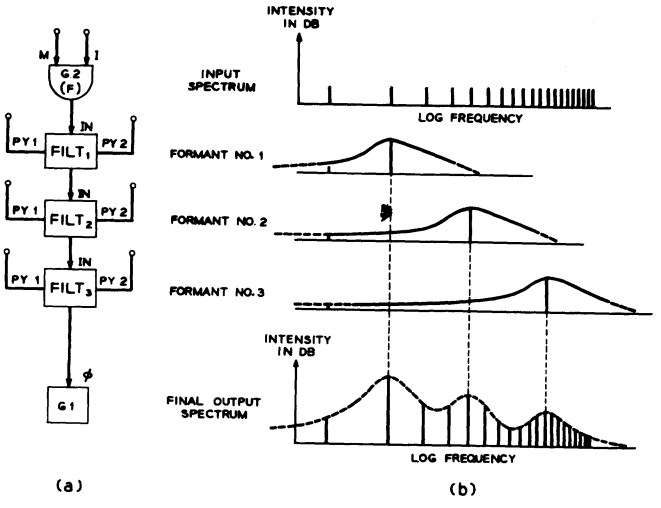


FIGURE 12
AN INSTRUMENT WITH THREE FORMANT FILTERS

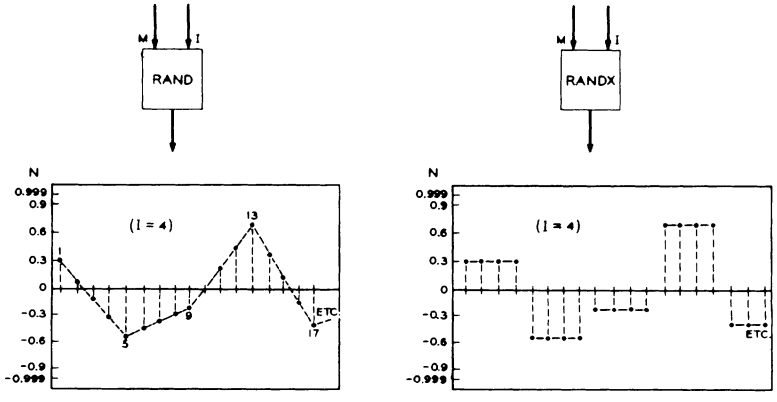
be seen in Figure 13, where each generator is shown in a block-diagram, along with a plot of a portion of a typical sequence of output samples from each. The operational meaning of the I inputs to these random generators is somewhat different from the meaning it had with respect to the oscillators. In the latter, values of a stored function are selected successively in a way determined by the I input, but while these values may have been nonadjacent samples in the original function (separated, that is, by the I - 1 intervening values of the stored function that were skipped), they are always adjacent at the output of the oscillator.

In practice, the RAND generator is most often used to modulate one of the inputs of an oscillator, to produce a band of noise. The waveform of the carrier signal generated by the oscillator in each instrument (designated by the (F) in 4u3 and 5u3) is usually sinusoidal, although other waveforms might be used as well. It has been found, however, that the effect of applying such random modulation to a complex wave such as a saw-tooth, whereby a large number of harmonic partials are modulated together – i.e., synchronously and in the same way – is very different from that of a complex signal in which all the harmonics have been modulated independently. The former sounds rather like radio "static," while the latter would sound like a white noise with more or less "coloration" provided by some kind of filter circuit. When random amplitude modulation is employed, the output of a RAND being connected to the M input of a G2, the bandwidth of the noise is determined by the value of the I input to the random generator, while its M input determines the amplitude of the noise. When the output of a RAND generator is connected to the I input of an oscillator, on the other hand, the resultant noise band is produced by random frequency modulation, and the bandwidth of the noise is determined by the M input of the random generator (the amplitude of the noise will be controlled in tnis case simply by the M input to the oscillator itself).

The reasons behind these relations between the noise bandwidth and the particular inputs of the random generator when the two kinds of modulation are employed are derived from modulation theory, and they are much too complicated to be considered here. But the basic relations that are of practical interest to the composer in using these generators may perhaps be clarified by some examples. Figures 14a and 14b show two simple instruments, each using a random generator to modulate an oscillator.

figure

13



(a) THE RAND GENERATOR, AND A TYPICAL OUTPUT SEQUENCE

(b) THE RANDX GENERATOR, AND A TYPICAL OUTPUT SEQUENCE

FIGURE 13

figure

14 & 15

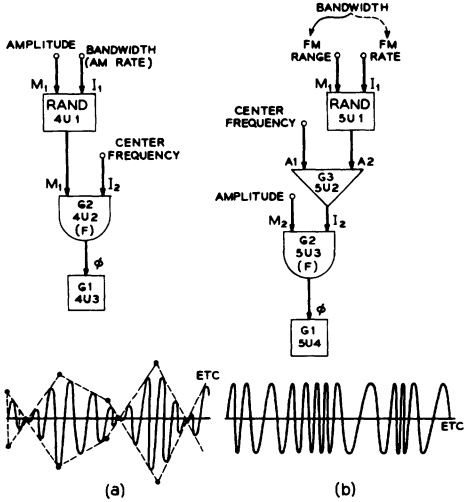


FIGURE 14

AN INSTRUMENT WITH A RANDOM AMPLITUDE-MODULATED OUTPUT SIGNAL

AN INSTRUMENT WITH A RANDOM FREQUENCY MODULATED OUTPUT SIGNAL

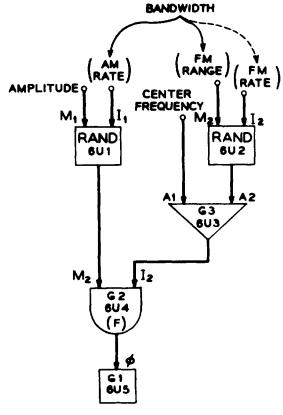


FIGURE 15

AN INSTRUMENT WITH A RANDOM AMPLITUDE AND FREQUENCY MODULATION

The mean amplitude and center frequency of the noiseband produced by instrument #4 (through random amplitude modulation) are controlled by the M_1 and I_2 inputs, respectively. The bandwidth of the signal is determined by the I_1 input (which specifies the rate of AM) and is approximately equal to I_1 times the sampling rate over 512 (i.e., $BW = \frac{10,000}{512} \times I_1$). In instrument #5, on the other hand (where the noise is produced by random frequency modulation), the amplitude is determined by the M_2 input, whereas the bandwidth is a function of both M_1 and I_1 (i.e., both the range and the rate of FM), and is approximately equal to their sum $(M_1 + I_1)$, multiplied by the sampling rate over 512, (i.e., $BW_{FM} = \frac{10,000}{512} \times (M_1 + I_1)$). (In addition, it has been found, experimentally, that the best results are obtained when $I_1 \cong 4M_1$.) Note that, in the FM case (instrument #5), the output of the random generator is put through an adder before going to the I input of the oscillator. The center frequency of the noiseband is given by the A1 input to the adder (if there were no such adder in the instrument, the noiseband would extend from zero frequency up to a frequency equal to one-half the specified bandwidth). The noisebands that are produced by an electronic noise-generator — and most noises produced by natural or mechanical sources — consist of random modulation of both the amplitude and frequency of a carrier signal, whereas with the instruments illustrated above only one of these parameters is modulated in a given signal. Such instruments will produce a noise band that is quite satisfactory in most cases, but if both kinds of modulation are required, an instrument like that in Figure 15 would be constructed, employing two separate RAND unit generators.

Instrument #6 essentially combines all the functions of instruments 4 and 5 — (rather like having two separate electronic circuits together on one chassis, with a single power supply common to both) — since the inputs to either one of the RAND generators could be set to zero, if a signal was wanted with only amplitude modulation or only frequency modulation. It is important to note that the two random number generators in instrument #6 (and, in fact, all those that might be used in the several instruments of any one orchestra are independent. The sequence of numbers generated by one RAND will never be the same as that generated by another in the same orchestra and in the same run on the computers, whereas two runs of the same score, using the same instruments and the same settings of all the inputs will always be identical. The reason for this is that all of the random generators derive their sequence of numbers from a common source, which distributes the numbers alternately among the several generators.

The practical usefulness of this method of distribution will be clarified in the following paragraphs, where some of the other uses of the RAND and RANDX unit generators are described.

The last three examples dealt with the random number generator, RAND, as a source of bands of noise. When intended for this purpose, relatively high values of its M and I inputs will be involved. Thus, in order to generate a signal with a mean amplitude of 1000 amplitude units (corresponding to 60 db), a center frequency of 1000 cps, and a bandwidth of 500 cps (i.e., a band of frequencies from 1000 + 250 to 1000 - 250 cps) using instrument 6, the settings of the various inputs would be equivalent to the following.

M_1	(amplitude),	1000 a.u.
I_1	(AM rate ~ bandwidth),	500/second
AI_2	(center frequency),	1000 cps
M_2	(FM range ~ bandwidth)	100 cps
I_2	(FM rate ~ bandwidth)	400/second

If we use much lower values for the I inputs to the random generators, however, the output signal of this instrument will no longer sound like a noise at all, but rather like a simple tone with (more or less) gradual though irregular variations in amplitude and/or frequency. The nature of the sounds produced in this way depends, of course, upon the particular ranges and rates of modulation that are used, but the ability to apply this kind of modulation to a carrier signal has proved to be of use in several ways. For example, we have found that such random fluctuations in the parameters of a tone — when used within certain rather narrow limits — impart a "naturalness" to the quality of the tone that is usually associated with the sounds of conventional musical instruments. Experiments are now being conducted to establish these limits more precisely, and to determine the effects of other settings — and for such experiments, the computer is an invaluable instrument.

It is also possible to use the random number generators in such a way that they control whole sequences of sounds, generating the successive elements of these sequences at a rate determined by the I inputs, and within a range given by the M inputs. For example, the I input to a RAND which modulates the amplitude (M) input of an oscillator could be set so that new numbers are generated at a rate of one per second, within a range from, say, 30 to 60 db, and with the duration of this "note" set at 20 seconds. The result would be a tone, lasting

20 seconds, with a gradual increase or decrease in amplitude every second. The same thing could be done at the frequency (I) input to the oscillator, with the range of the RAND generator set at, say, 4000 cps, (the size of the frequency band now used most often), in which case the tone would be heard as a series of glissandi from one (random) frequency to another.

If a sequence of tones is wanted in which the pitch of each is constant, but in which the changes from one pitch to another are still random, the other type of random number generator mentioned earlier – the noninterpolating generator called RANDX – would be used instead of RAND. With these two kinds of random generator – in conjunction with the other unit generators, of course – complex instruments can be designed which virtually "compose" long quasi-random sequences within the limits of parametric range that are specified by the composer. The importance of such instruments in the exploration of various kinds of musical structures should be evident.

Conversion Function Subroutines

The inputs to each unit generator represent numbers that are to be used in the arithmetic operations defined by the corresponding group of instructions in the program. But since the range of numbers used by the computer does not generally correspond to the range of numbers ordinarily used to specify the parameters of a sound, it is desirable to include a group of subroutines in the program, which will define the necessary transformations from the one set of numbers to the other. The use of these conversion function subroutines thus enables the composer to use numbers that are directly meaningful to him (e.g., frequency in cycles per second, intensity in decibels, duration in seconds or fractions of a second, etc.), the necessary transformations being carried out automatically by the computer. The composer must decide, therefore, in what form he wishes to be able to specify the various parametric values of the sounds to be generated, and either write the necessary conversion functions himself, or select these from the library of subroutines that have already been compiled.

The formulation of these conversion functions is determined by (1) the sampling rate (10,000 samples/second), (2) the range of amplitude values used (-2047 to +2047), and (3) the specific mechanics of sample-generation in each unit generator (e.g., the relation between frequency, in the G2 oscillator, and the number 512, as described earlier). In addition, each conversion function must specify where, on the note-card, the num-

ber that is to be converted will be found. The note-cards are divided into 12 numeric fields, designated P1 through P12, each of which may be used to specify one parameter of the sound to be generated. Each conversion function is given a name, consisting of the letters CVT followed by a two-digit decimal number (e.g., CVT03, CVT17, etc.). Just as with the waveform functions mentioned earlier, these conversion functions are stored in the computer memory at the beginning of the computation, and there is room in storage for up to 20 of them in any one run. The general form of these conversion functions is then

$$CVTm = f(Pn)$$

where m is a 2-digit number from 01 to 20, n is a number from 2 to 12 (P1 is always used for the designation of the instrument number, and does not involve a conversion function), and f(Pn) simply means "some function of the number in the nth field on the note-card." Examples of the more frequently used conversion functions will be given later, in the section on program-language description of the instrument. In these descriptions, CVT is abbreviated to C, and the numerical suffix may be a single digit (e.g., C6 for CVT06, etc.).

Function-Generating Subroutines

The instruments shown in block-diagrams above all involve a number of stored functions (F1, F2, etc.). These must be defined by a set of function-generating subroutines, fed into the computer ahead of the note-cards and stored in the memory so that they are available for the computations. Each of the subroutines is able to generate a particular type of function, on the basis of a sequence of numbers punched into cards. Several of these are available, in the form of separate decks of cards that are used with the main program whenever the corresponding types of functions are involved in the composition. One such generating subroutine, called GEN07, enables the composer to define functions made up of straight-line segments, and is often used for relatively simple envelopes and waveshapes. Another, GEN09, makes it possible to define a function in terms of a distribution of harmonic partials, with provisions for the specification of both the relative amplitude and phase of each partial. A third, GEN05, is used to generate exponential functions, and so on.

Writing up the function definitions is relatively simple for most of these subroutines. In the case of the GEN07, for example, cards are punched with a set of numbers which give the values of each successive juncture of the line segments in the vertical

dimension, and the horizontal distances between them, as shown in Figures 16a and b. The other generating subroutines are slightly more complicated to use than the GEN07, but they are easily learned. The information that has to be specified for the GEN09 subroutine (sum of sinusoids) includes the number, amplitude and phase of each harmonic. With the GEN05 (sum of exponentials), the time-constant and coefficient of each term have to be determined and specified on the cards. Other function-generating subroutines are available, and still others can be written when they are needed, so that there is virtually no limit to the kinds of timbre waveforms, amplitude and frequency envelopes, etc., that can be used by the composer.

Program-Language Description of the Instrument

With this much introduction to the logic behind the designing of instruments and the formulation of conversion function subroutines, it should not be difficult for the composer to become accustomed to the instrument descriptions themselves, when they are translated from block-diagram form into actual programming language. In Figure 17b such a description is given for the instrument shown previously (instrument #3, Figure 9). A slightly modified version of the block-diagram of this instrument is also shown (Figure 17a) for easier reference. (The modifications consist simply of more specific designation of the function numbers in the oscillators than were used previously.) The essential information that must be contained in the description includes the type of each unit generator (G2, G3, etc.), the instrument-number and the sequential order of each generator (3u1, 3u5, etc.), the number of the function to be assigned to each of the oscillators, designations of the inputs to each generator and of the location of its output, etc.

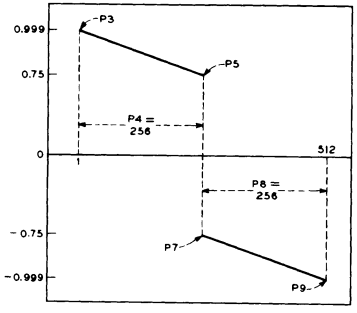
The "MAC" (for macro) in columns 8-10 of the punched card designates the type of instruction which follows (beginning in column 16). A macro instruction is one that involves a group of instructions somewhere else in the program. (These are "built-into" the music compiler, however, so the composer does not have to be concerned with these instruction groups.) Thus, MAC G2 actually calls up one group of instructions labelled G2; MAC G3 another group (named G3); MAC S1 still another, and so on.

When one of the inputs to a unit generator is to be a constant for that instrument (as with the I input to the vibrato generator, 1u2, determining the frequency or rate of vibrato) this constant is specified in the MAC instruction for that generator, as

50

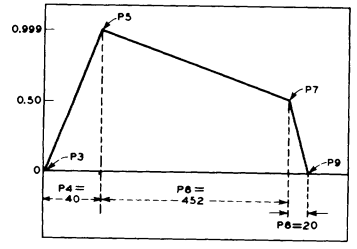
figure

16



VALUE OF EACH JUNCTURE IN VERTICAL DIMENSION	HORIZONTAL DISTANCE BETWEEN JUNCTURES
P3 = 0.999 P5 = 0.75 P7 = -0.75 P9 = -0.999	P4 = 256 P6 = 0 P8 = 256

(a) GEN 07 REPRESENTATION OF A MODIFIED SQUARE-WAVE

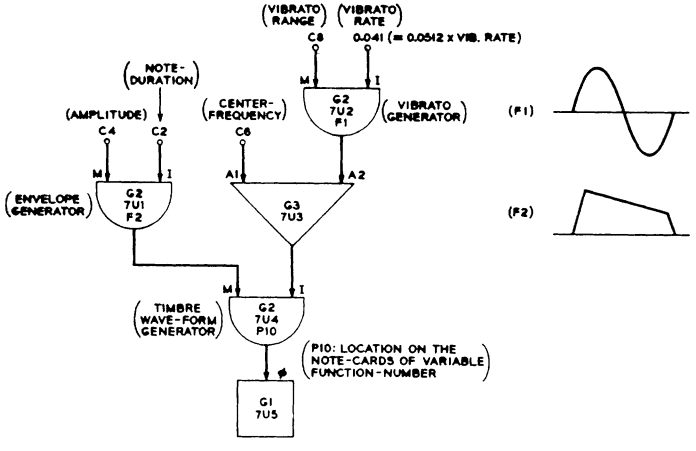


VERTICAL	HORIZONTAL
P3 = 0 P5 = 0.999 P7 = 0.75 P9 = 0	P4 = 40 P6 = 452 P8 = 20

(b) GEN 07 REPRESENTATION OF AN ENVELOPE FUNCTION

FIGURE 16

figure 17



(a) INSTRUMENT NUMBER 7 (17)

```
      8    16
1  | MAC | G2, 3U1, F2, (3U4, M), X, X
2  | MAC | G2, 3U2, F1, (3U3, A2), X, 0.041 B17 *
3  | MAC | G3, 3U3, (3U4, I), X, X
4  | MAC | G2, 3U4, X, (3U5, 0), X, X
5  | MAC | G1, 3U5
6  | MAC | S1, I7, ((3U1, M, C4)(3U1, I, C2)(3U2, M, C8) &
7  | ETC | (3U3, A1, C6)(3U4, F, P10))
```

(b)

* B17 IS SIMPLY A PROGRAMMING CONVENTION WHICH SPECIFIES THE LOCATION OF THE DECIMAL POINT OF THIS NUMBER IN STORAGE, AND NEED NOT CONCERN US HERE

FIGURE 17

shown in line 2. Otherwise, the spaces in the instruction that are available for such input constants are filled by an X. Thus, an X in one of these instructions indicates that the value of that input is either a parameter to be specified on the note-cards, or is given by the output of some previous unit generator. Whenever an instruction is too long to fit into one card, the symbols $ (at the end of one line) and ETC (in place of MAC, at the beginning of the next line) are used to indicate the continuation of a single instruction from one card to the next. Thus, in the example above, the MAC S1 instruction includes both lines 6 and 7.

The allocation of conversion functions to each of the inputs that requires one is determined by the setting routine (MAC S1, lines 6 and 7 in the example) for this instrument (I3). Also given by MAC S1 are any designations of the note-card field (Pn) in which a variable function number is to be found. In this instrument, for example, while 3u1 and 3u2 have function numbers that are constant, and thus cannot be varied from note to note, the function to be generated by 3u4 is variable, and the last expression in the MAC S1 routine (line 7) says that the function number for each note will be given in field ten (P10) of the note-cards.

In the above instrument description, four conversion functions are listed, viz. C2, C4, C6 and C8, involving note-duration, amplitude (or intensity), center frequency and vibrato range, respectively. If a linear amplitude scale is to be used, C4 would be set equal to the number in P4, i.e.,

$$\text{CVT04} = \text{P4}$$

and the amplitudes would be specified on the note-cards by numbers from 1 to 2047. If a logarithmic or decibel scale is to be used, the appropriate conversion is

$$\text{CVT04} = 10^{\text{P4}/20}$$

(Thus, 60 db will become $10^{60/20} = 10^3$ or 1000 amplitude units; 40 db in P4 will mean $10^{40/20} = 10^2$ or 100 amplitude units, etc.)

For center frequency conversion, there are, again, several alternatives. For frequency in cps, the conversion function would be written as follows.

$$\text{CVT06} = \frac{512}{10,000} \times \text{P6} = .0512 \times \text{P6}$$

But suppose the composer would like to specify this parameter on a logarithmic (rather than linear) frequency scale, by putting on each note-card a number which represents the number of pitch-steps (assumed in this case to be equal) the given tone is above some lowest base frequency. A conversion function can be written which will do this. For example, for a scale like that of the piano, with 12 equal steps to the octave, and a base frequency of 27.5 cps, the conversion function would be

$$CVT06 = .0512 \times 27.5 \times 2^{P6/12}$$

More generally, a conversion function, C6, for any tempered scale with a base frequency F cycles per second, and N equal steps to the octave would have the form:

$$CVT06 = .0512 \times F \times 2^{P6/N}$$

For the vibrato-range conversion function, C8, a form that has been found convenient is the following (when P6 is used for specifying center frequency in cps):

$$CVT08 = .0512 \times P8 \times P6 \times .01$$

Using this conversion function, the range of the vibrato (in cycles per second above and below the center frequency) can be expressed on the note-cards as the percentage of deviation from the center frequency.

The C2 conversion function requires a rather more elaborate explanation than do the others. In the first instrument shown in block-diagram form above (Figure 6) there were only two inputs: the M and I inputs to the G2 oscillator. These were used to control the amplitude and frequency of the signal to be generated by the instrument, and the parametric values for each note were to be found in appropriate fields on the note-cards. The duration of each note must also be specified on the note-cards, and the second numeric field has been reserved for this purpose. That is, P2 is always used for the specification of note-duration. In order for the computer to make use of this number in its computations, it has to represent the duration in terms of the number of samples included by the note. But with a sampling rate of 10,000 per second, it would obviously be impractical to write in one field the actual number of samples, so a multiplying factor is used, given by a special TME (for time) card. In the third field of this card (P3) a number is punched which is multiplied (in the computer) by the numbers in P2 on subsequent note-cards. The product

of these two numbers then represents the number of samples in the note, and therefore its actual duration. The number on the TME card thus determines the size of the temporal unit that will be assumed for the P2 specification. For example, if the TME card has the number 10,000 in P3, then the number in P2 on subsequent note-cards will represent duration in seconds (assuming a sampling rate of 10,000 per second); if the time-factor is 1,000, P2 will express the duration in tenths of a second; if the TME card says 10, P2 will represent milliseconds, etc.

The fact that these TME cards may be inserted at any point within the sequence of note-cards constituting the score makes it possible to change the time-scale or tempo of the sequence of sounds that are generated, in a way that is precisely analogous to the way time-values are varied in conventional musical notation. There, the actual durations represented by a given note-value (half-note, quarter-note, etc.) can be varied by simply changing the tempo indications. Similarly, with the music program, the numbers in P2 can be kept constant, while the actual durations are made to vary (gradually, suddenly) by means of a change in the scale-factor given by a TME card.

Now the C2 conversion function that is needed for the I input to the "envelope generator" of our instrument must cause this oscillator to generate just one period of the stored function (F2) per note. The conversion function that does this is the following.

CVT02 = 512/P2

where it is assumed that the value referred to by "P2" has already been multiplied by the scale factor on the TME card, and thus represents the number of samples in the note.

The Computer Score

When the descriptions of each instrument to be used have been written up in this form, and the information has been punched into cards, (one card for each line of instruction), these are put into the computer with the music compiler (which is in the form of another deck of punched cards). The computer punches out a new deck of cards, which actually consists of a recoded version of the music compiler itself, with the set of instructions describing the instruments inserted at appropriate points. This new deck is what is called the "orchestra," and it comprises the basic set of instructions that are necessary to effect

the transformation from the parametric values in the score to the amplitude samples on the output tape.

Two musical examples will now be used to illustrate the writing of the computer score. In Figure 18 a short sequence of tones is shown in conventional musical notation. Where this notation alone is not sufficient to describe the sounds – i.e., with respect to their envelopes, timbre waveforms and vibrato ranges – a verbal description is used. The instrument described in the last section (Instrument #7, Figure 17 and reproduced in Figure 19a) may be used for the tones on the upper staff in Figure 18, but a second instrument (#8, Figure 19b) will be required for the others, because of the difference in the shape of the envelopes indicated for the two parts – the one having a decreasing, the other an increasing amplitude in the main body of each tone. This difference is represented in Figure 16 by the function numbers assigned to the first unit generator in each instrument (F2 in 7u1, F3 in 8u1). Otherwise the two instruments are identical. The set of conversion functions selected for this example is also given in Figure 19.

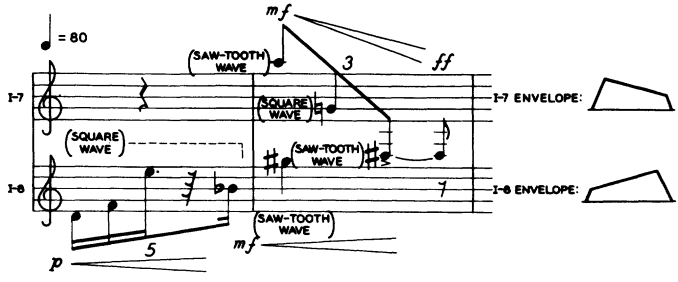
In addition to the two envelope functions (F2 and F3) and the vibrato function (F1) which are permanently assigned to these instruments, the score itself calls for two timbre-functions; a square-wave and a sawtooth wave, which will be designated F4 and F5, respectively. These variable functions will be specified in field P10 on the note-cards for both instruments. All five functions are plotted in Figure 20 with the numbers that will be used in writing up the function-definitions in the score. Finally, Figure 21 shows the computer score itself, each line representing a single punched card.

The several different kinds of card which are included in the computer score are distinguished by the letters (or blanks) in columns 1 to 3 (called the OP or operation field). Blanks in this field indicate a regular note-card, and are used to conserve effort in the writing of the score, since note-cards are used more frequently than any of the others. Rests in an instrument are indicated by a rest-card (denoted by RST, as in line 10 of the score), with the duration of the rest specified in P2 (just as with the note-cards).

The GEN cards in lines 1 through 5 define the functions to be used in the instruments. Note that four of the five functions are composed of straight line segments, and thus refer to the subroutine called GEN07, while F1 (the sinewave function for the vibrato generators) uses the GEN09 subroutine. The TME

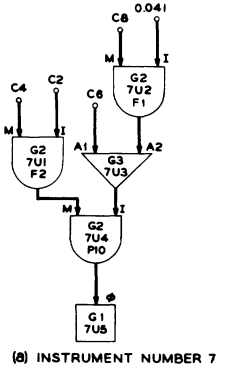
figure

18 & 19

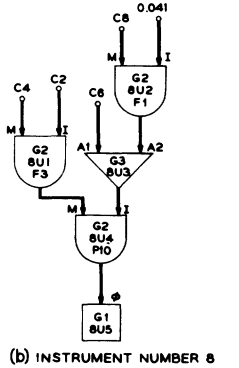


I-7 VIBRATO-RANGE: ±1.0% CENTER FREQUENCY
I-8 VIBRATO-RANGE: ±1.5% CENTER FREQUENCY
THRU B♭; ±0.5% C.F. ON G#

FIGURE 18
SEQUENCE IN MUSICAL NOTATION



(a) INSTRUMENT NUMBER 7



(b) INSTRUMENT NUMBER 8

FIGURE 19

CONVERSION FUNCTIONS TO BE USED

C2 = 512/P2
C4 = $10^{P4/20}$
C6 = 0.0512 x P6
C8 = 0.0512 x P6 x P6 x 0.01

figure

20

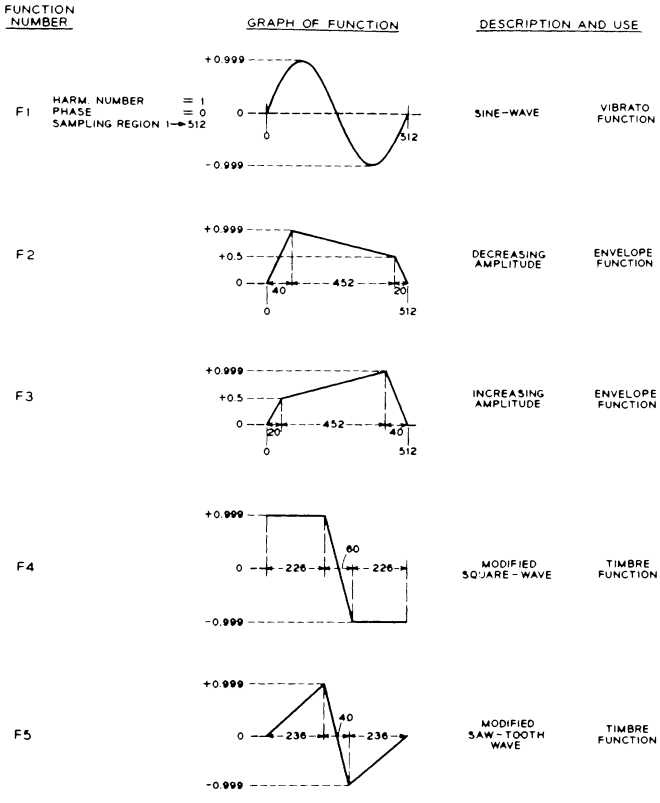


FIGURE 20
WAVE-FORM FUNCTIONS FOR EXAMPLE 1

figure

21

	OP 1-3	P1 ←6	P2 ←12	P3 ←18	P4 ←24	P5 ←30	P6 ←36	P7 ←42	P8 ←48	P9 ←54	P10 ←60	P11 ←66	P12 ←72	73→
	FUNCTION-GENERATING SUBROUTINE		FUNCTION NUMBER	NUMERICAL DESCRIPTIONS →										
1	GEN	09	1	0.999	1	0	1	512	0					
2	GEN	07	2	0	40	0.999	452	0.5	20	0				
3	GEN	07	3	0	20	0.5	452	0.999	40	0				
4	GEN	07	4	0.999	226	0.999	60	-0.999	226	-0.999				
5	GEN	07	5	0	236	0.999	40	0.999	236	0				
6	TME			1000										
	OPERATION CODE	INSTRUMENT NUMBER	DURATION IN TENTHS OF A SECOND		INTENSITY IN DECIBELS		FREQUENCY IN CPS		VIBRATO-RANGE (± % OF CENTER-FREQUENCY)		TIMBRE-FUNCTION NUMBER			
7		8	1.5		50		293.7		1.5		4			
8					52		349.2							
9			2.25		54		659.3							
10	RST		0.75											
11			1.5		56		466.2							
12	MES													
13		8	7.5		60		830.6		0.5		5			
14		7	2.5		60		880.0		1.0		5			
15					62		392.0				4			
16			3.75		64		185.0				5			
17	MES													
18	TER													

FIGURE 21
COMPUTER SCORE FOR EXAMRLE 1

card (line 6) sets the time scale of the sequence, as described earlier. In this case, the time-factor is 1000, so the numbers in P2 on the following cards will represent the duration of the note (or rest) in tenths of a second (a value of 1.0 in P2 = 1 × 1000 samples = 1000/10,000 or 1/10 second).

In every computer score there will be one or more measure cards (MES), which serve to subdivide the sequences into smaller segments, and are similar, in certain respects, to the barlines in musical notation, though their function is actually quite different. Finally, there must always be a termination card (TER), to signal the end of the composition (or rather, the end of the computation).

The function of the measure card may be clarified by the following. The music generating process in the computer is carried out measure by measure, and is divided into three main phases: (1) a card-reading phase, (2) a sorting phase, and (3) the sample-generating phase. The computer reads in the data-cards until a MES card is reached, at which point the program causes a shift into the sorting phase — the arranging of the samples generated by different instruments into their proper temporal sequence. This is necessary whenever two or more instruments are to be "played" simultaneously in a given measure, and since the computer does this sorting automatically, the several instrumental parts in one measure do not have to be ordered in their proper temporal sequence in the score itself. The sequence of notes in each instrumental part must be so ordered, however.

If certain instruments of the orchestra are not used at all in a given measure, it is not necessary to list any rests for these instruments, and the computer will automatically insert rests between the end of the last note in a given instrument and the end of the measure (determined by the duration of the longest instrumental part), so that such final rests need not be specified in the score. Thus, in Figure 21, no rest is specified for instrument #7 in the first measure, and the final eighth-note rest in instrument #8 has not been given, because the computer will supply this rest automatically. If an instrument does not begin at the beginning of the measure, the initial rest does have to be specified, of course.

A single measure may not exceed 100 seconds in duration, and may not contain more than 100 "notes" (including both note- and rest-cards). Usually, much shorter measures are used, this having the advantage that any errors in the specification

of duration (for notes or rests) will have only a limited effect. That is, a single error near the beginning of a measure will produce a temporal displacement throughout the entire measure, and it is well to guard against this by inserting frequent MES cards in the score.

Within one measure, parametric values need to be specified only when they change. Thus, if some parameter remains constant over several notes, these fields on the cards following the first can be left blank; the computer will carry the previous value over into the next note automatically. After a measure card, however, all parameters must be specified anew, whether they are different from those in the previous note (i.e., the last note of the previous measure) or not. This, again, serves to facilitate the writing of the score. In the present example, the vibrato-range and timbre-function are constant throughout the first measure, and are thus specified only for the first note in the measure. Similarly, the durations of the first two notes in the first measure, and the intensities of the first two notes in the second measure are identical, and have been specified only once in each case.

The first example was intended merely to illustrate the basic procedures involved in writing up a computer score. For this reason, the musical sequence used was quite simple – and, in fact, could almost have been played by conventional musical instruments. In the second example, more sophisticated instruments will be employed to generate a more complex musical sequence, one that could not be realized by conventional instruments, in order to demonstrate more explicitly some of the unique potentialities of the computer medium. At the same time, this last example will serve as a general review and summary of some of the material covered in earlier sections of this article. The sequence to be generated consists of three independent "voices," shown [graphically] in Figures 22a through 22c. Clearly, conventional musical notation is no longer adequate to describe such a sequence, so that a graphic and/or verbal description must be used instead*3. Three instruments will be needed, and the structure of each instrument will be determined by the kinds of sounds and sound-sequences it will be used to generate.

The first voice begins as a wideband noise, then the bandwidth narrows to zero, to produce a simple tone. Thereafter, the signal varies in bandwidth from zero to 200 cps, the latter corresponding to an interval of approximately a minor third (more or less, depending on the center frequency). The band-

figure

22

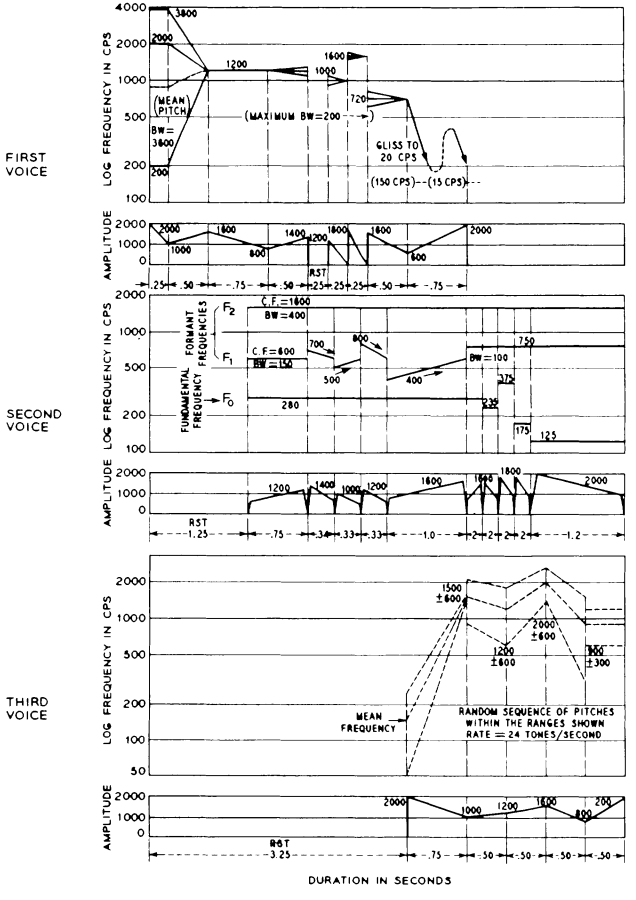


FIGURE 22

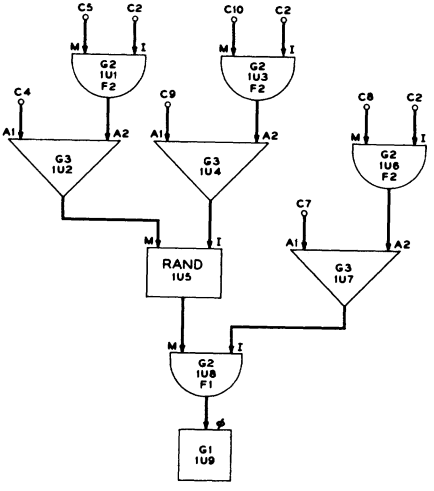
width of the noise must thus be continuously variable within each note. In addition, there are two places at which the center frequency of the signal glides continuously from one point to another, and the amplitude variations are also continuous in this way. An instrument which can effect these continuous changes in the amplitude, bandwidth and center frequency of a noiseband is diagrammed in Figure 23a, and its program-language description is given in 23b. This instrument incorporates three pairs of unit generators whose sole purpose is to produce linear interpolations between an initial and a final value in each of the three parameters of the output signal that are to be varied in this way. Each pair or interpolation couple consists of an oscillator and an adder. The initial value of the parameter forms the A1 input to the adder, while the A2 input, taken from the output of the oscillator, must represent the difference between the initial and final values of the parameter. This is achieved by (1) assigning to the oscillator an interpolation function (F2) which goes from zero to +.999 in one period (This function, with the others to be used for this sequence, is shown graphically in Figure 26), (2) using the C2 conversion function described earlier (CVT02 = 512/P2), so that the oscillator generates just one period per note, and (3) either specifying directly the difference between the initial and final parametric values, or writing a conversion function which will compute this difference, and specifying (at G2, M) simply the final value itself. In this example, the latter alternative will be used.

The graph of the second voice (Figure 22b) shows the variations in frequency of the fundamental (F0) and the first and second formants (F_1 and F_2) of a complex tone with many harmonic partials. The center frequency of the first formant is continuously variable (like the three parameters of voice 1, described above), while that of the second formant is constant throughout the sequence, and so will be defined as a constant for this instrument. The bandwidth of the first formant is constant within each note, but takes two different values in the course of the sequence (150 cps at the beginning, 100 cps at the end) and so must be variable from note to note, and therefore specified as one of the parameters on the note-cards. The bandwidth of the second formant, like its center frequency, will be a constant for the instrument.

In the amplitude plot for the second voice, a different means of control or specification is implied than for the first voice. Here, instead of the continuous linear change from one amplitude to the next, two fixed envelope forms will be used, one

figure

23



(a) INSTRUMENT NUMBER 1

```
8→ 16→
MAC  G2, 1U1, F2, (1U2, A2), X, X
MAC  G3, 1U2, (1U5, M), X, X
MAC  G2, 1U3, F2, (1U4, A2), X, X
MAC  G3, 1U4, (1U5, I), X, X
MAC  RAND, 1U5, (1U8, M), X, X
MAC  G2, 1U6, F2, (1U7, A2), X, X
MAC  G3, 1U7, (1U8, I), X, X
MAC  G2, 1U8, F1, (1U9, φ), X, X
MAC  G1, 1U9

MAC  S1, I1, ((1U1, M, C5)(1U1, I, C2)(1U2, A1, C4)(1U3, M, C10) &
ETC  (1U3, I, C2)(1U4, A1, C9)(1U6, M, C8)(1U6, I, C2) &
ETC  (1U7, A1, C7))
```

CONVERSION FUNCTIONS USED

C2 = 512/P2
C4 = P4
C5 = P5 − P4
C7 = 0.0512 x P7
C8 = 0.0512 x (P8 − P7)
C9 = 0.0512 x P9
C10 = 0.0512 x (P10 − P9)

(b)

FIGURE 23

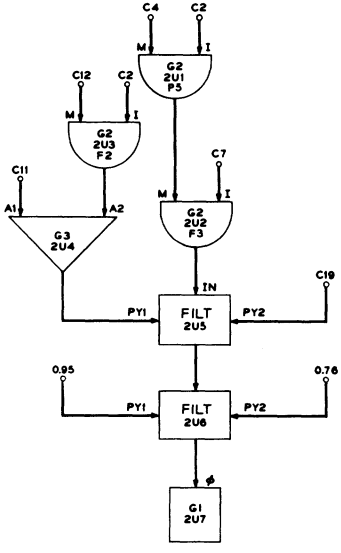
increasing, the other decreasing in amplitude from the beginning to the end of the central portion of the envelope. This change in amplitude will always be by a factor of 2, which corresponds to an intensity change of 6 db. In addition, the envelopes will have a fast but not instantaneous attack and decay, so that each note will be clearly articulated from the previous note. (In the first voice, this will only be so when the interpolation function begins or ends at zero amplitude. Otherwise, the sounds will pass gradually from one to the next.)

Instrument #2, Figure 24, is designed to generate the sounds of this second voice. The PY1 input to the first filter (2u5) is the same kind of interpolation couple used in instrument #1, making it possible to vary the center frequency of the first formant in the way shown in the graph. The bandwidth of this filter will be specified in field P9 on the note-cards. Both the center-frequency and the bandwidth inputs to the second filter, however, are specified as constants for the instrument (see line 6 of the instrument description, Figure 24b). The envelope of each note is variable, and will be specified in P5 (see the MAC S1 setting routine, line 8), while the timbre waveform assigned to the second oscillator (2u2) is fixed, and a sawtooth wave-form will be used (F3, in Figure 26).

Instrument 3 (Figure 25), uses a RANDX (noninterpolating) random number generator and an interpolation couple, modulating the I (frequency) input of the main oscillator, to generate a fast sequence of tones whose pitches are essentially random within the limited ranges plotted in Figure 22c, the graph of the third voice. The mean frequency of the distribution of pitches is thus continuously variable, while the range of deviations from this mean (given at the M input to 3u4) is constant for each "note" (this word "note" takes on a more generalized meaning here, since each of the note-cards for this instrument will actually cause many tones to be generated). For the control of amplitude, linear interpolation will again be used, just as in instrument #1. In this case, however, the period of interpolation will include many distinct tones, so that the output of the adder must provide the M input to another oscillator (3u3), which will be used to generate the envelope of each of the single tones within the longer "note." The value at this input will determine the maximum amplitude of each envelope, while the changes of amplitude within each tone will primarily be determined by the shape of this envelope function itself (F6, Figure 26). The rate at which these tones are to be articulated within the longer note is 24 per second. This will be specified in field P6 on the note-cards, and it should be noted that the

figure

24



(a) INSTRUMENT NUMBER 2

```
6→ 16→
|MAC | G2, 2U1, X, (2U2, M), X, X
|MAC | G2, 2U2, F3, (2U5, IN), X, X
|MAC | G2, 2U3, F2, (2U4, A2), X, X
|MAC | G3, 2U4, (2U5, PY1), X, X
|MAC | FILT, 2U5, (2U6, IN), X, X
|MAC | FILT, 2U6, (2U7, φ), 0.95B17, 0.76B17
|MAC | G1, 2U7
|
|MAC | S1, I2, ((2U1, M, C4)(2U1, I, C2)(2U1, F, P5)(2U2, I, C7) &
|ETC | (2U3, M, C12)(2U3, I, C2)(2U4, A1, C11)(2U5, PY2, C19))
```

CONVERSION FUNCTIONS USED

C2 = 512/P2

C4 = P4

C7 = 0.0512 x P7

C11 = A FUNCTION OF P11 (INITIAL CENTER-FREQUENCY)

C12 = A FUNCTION OF P12-P11 (FINAL CENTER-FREQUENCY)

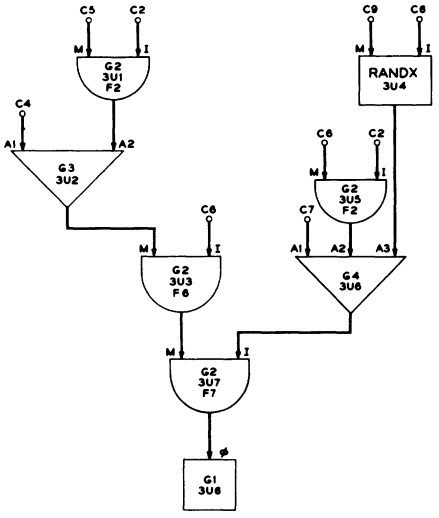
C19 = A FUNCTION OF P9 (BANDWIDTH)

(b)

FIGURE 24

figure

25



(a) INSTRUMENT NUMBER 3

6→	16→	CONVERSION FUNCTIONS USED
MAC	G2, 3U1, F2, (3U2, A2), X, X	C2 = 512/P2
MAC	G3, 3U2, (3U3, M), X, X	C4 = P4
MAC	G2, 3U3, F6, (3U7, M), X, X	C5 = P5-P4
MAC	RANDX, 3U4, (3U6, A3), X, X	C6 = 0.0512 x P6
MAC	G2, 3U5, F2, (3U6, A2), X, X	C7 = 0.0512 x P7
MAC	G4, 3U6, (3U7, I,), X, X, X,	C8 = 0.0512 x (P8-P7)
MAC	G2, 3U7, F1, (3U8, ø), X, X	C9 = 0.0512 x P9
MAC	G1, 3U8	
MAC	S1, I3, ((3U1, M, C5)(3U1, I, C2)(3U2, A1, C4)(3U3, I, C6) &	
ETC	(3U4, M, C9)(3U4, I, C8)(3U5, M, C8)(3U5, I, C2)(3U6, A1, C7))	

(b)

FIGURE 25

figure

26

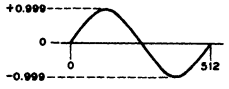
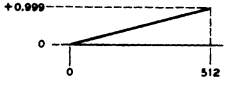
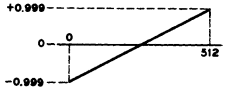
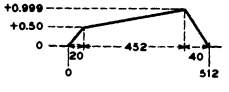
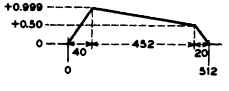
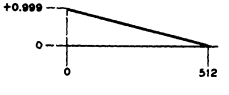
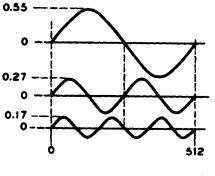
FUNCTION NUMBER	GRAPH OF FUNCTION	DESCRIPTION AND USE	
F1		SINE-WAVE	TIMBRE FUNCTION
F2		LINEAR	INTERPOLATION FUNCTION
F3		SAW-TOOTH WAVE	TIMBRE FUNCTION
F4		INCREASING AMPLITUDE	ENVELOPE FUNCTION (FOR I-2)
F5		DECREASING AMPLITUDE	ENVELOPE FUNCTION (FOR I-2)
F6		DECREASING AMPLITUDE	ENVELOPE FUNCTION (FOR I-3)
F7		SUM OF FIRST THREE HARMONIC PARTIALS	TIMBRE FUNCTION (FOR I-3)

FIGURE 26

WAVE-FORM FUNCTIONS FOR EXAMPLE 2

same number will be used to control the I inputs to two different generators in this instrument (i.e., both 3u3 and 3u4), just as P2 – by way of the C2 conversion function – is often common to several unit generators in the same instrument.

The timbre waveform (F7) assigned to 3u7 is shown in Figure 26, and consists of the first three harmonic partials, with relative amplitudes of approximately 1 to 1/2 to 1/3. This function has been used here to illustrate how the GEN09 function definitions are written when more than one sinusoidal component is involved. The definition of F7 is shown in lines 7 and 8 of the computer score (Figure 27a).

The conversion functions that will be used for each instrument are listed along with the block-diagrams and descriptions in Figures 23 through 25. C5, C8 and C10, for instrument #1 are examples of the conversion functions mentioned above which are used for the M input to the oscillator in each interpolation couple, allowing for the specification, on the note-cards, of the initial and final values of the parameter. In some cases, it has been possible to use the same conversion functions for all three instruments (viz., C2, C4 and C7), while others are common only to two of them. With instrument #2, however, the group of functions to be used for specifying the center-frequencies and bandwidths of the filters are unique, involving some rather elaborate equations. These would be of little interest here, so they are not given, but they make it possible to specify both center-frequency and bandwidth in cycles per second. In addition, one of them (C12) provides the conversion from final to final-minus-initial center-frequency that is needed for the interpolation couple in this instrument.

In the first instrument, the conversion function for specifying the bandwidth of the noise to be generated could have included the factor of 512/5,000 that would be necessary to allow the specification of bandwidth directly as indicated, but this has not been done here, in order to make this one conversion function available also for the M input of the RANDX generator in instrument #3, specifying the range (i.e., "bandwidth") of variation in pitch around the mean of the random sequence of tones in the third voice.

As for the computer score itself, shown in Figure 27, there is not much left to be said, except perhaps to note the following: (1) In writing a GEN 09 function definition involving more than two harmonics, the number in P12 on the first card must be preceded by a minus sign as in line 7. This is merely a code,

figure

27

	OP 1-3	P1 ←6	P2 ←12	P3 ←18	P4 ←24	P5 ←30	P6 ←36	P7 ←42	P8 ←48	P9 ←54	P10 ←60	P11 ←66	P12 ←72	73→
1	GEN	09	1	0999	1	0	1	512	0					
2	GEN	07	2	0	512	0999								
3	GEN	07	3	-0999	512	0999								
4	GEN	07	4	0	20	0.50	452	0.999	40	0				
5	GEN	07	5	0	40	0.999	452	0.50	20	0				
6	GEN	07	6	0999	512	0								
7	GEN	09	7	0.55	1	0	1	512	0.27	2	0	1	-512	
8				0.17	3	0	1	512	0					
9	TME			100										
10		1	25		2000	1000		2000	2000	360	360			
11			50		1000	1600			1200		0			
12			75		1600	800		1200		0				
13			50		800	1400					40			
14	RST		25											
15					1200	0		1000	1000	40	0			
16					1800	0		1600	1600					
17			50		1600	600		720	720					
18			75		600	2000		720	20	0				
19	RST	2	125											
20			75		1200	4		280		150		600	600	
21			34		1400	5						700		
22			33		1000							500		
23					1200							800		
24			100		1600	4						400		
25			20							100		750	750	
26						5		235						
27					1800			375						
28								175						
29			1.20		2000			125						
30	RST	3	325											
31			75		2000	1000	24	150	1500	100				
32			50		1000	1200		1500	1200	600				
33					1200	1800		1200	2000					
34					1800	800		2000	900					
35					800	2000		900		300				
36	MES													
37	TER													

FIGURE 27
COMPUTER SCORE FOR EXAMPLE 2

which causes the computer to read the numbers on the following card or cards, interpreting them as a continuation of the same function definition. (2) The time-factor of 100, set by the TME card in line 9, means that the numbers in P2 on the following cards will specify duration in hundredths of a second (25 in P2 = .25 seconds, 50 = 1/2 second, etc.). (3) As noted earlier, repetitions of a parametric value in a given field do not have to be written out; the computer will carry the numbers over from one card to the next automatically.

It is hoped that this last example will give some indication of the potentialities of the digital computer as a musical medium, with respect to both the enormous variety of sounds that can be produced, and the high degree of precision that is possible in the control of all the parameters of these sounds. These attributes alone should make it an attractive medium to the composer searching for new materials and new ways of organizing these materials. The extent to which computers will be more widely used for musical purposes depends, of course, on their becoming more generally available to composers than they are now. It is to be expected, however, that as faster and more efficient computers are built, and computer time becomes less expensive this problem will become less critical than it is at present.

REFERENCES

1 Mathews and Guttman, Generation of Music by a Digital Computer, Proceedings, 3rd International Congress on Acoustics, Stuttgart, 1959 (Elsever Publishing Co., Amsterdam). See also: M. V. Mathews, An Acoustic Compiler for Music and Psychological Stimuli, Bell System Technical Journal, May 1961, and Mathews, Pierce and Guttman, Musical Sounds from Digital Computers, Gravesaner Blätter, April 1962.

2 David, Mathews and McDonald, A High Speed Data Translator for Computer Simulation of Speech and Television Devices, Bell System Technical Monograph No. 3405.

3 In these graphs a logarithmic scale is used for frequency, a linear scale for amplitude. The frequency plots are not strictly accurate, however, because the interpolating function that will be used in the instruments is based on a linear scale. This would mean that all the diagonals in the log frequency plots should really be curved (concave downward) instead of straight.