The 'SCORE' Program for Musical Input to Computers

Leland Smith
Stanford University

Abstract:

The score program has been in use at Stanford for about ten years now. What began as a system for generating parameters for a particular piece of music has evolved into a general purpose musical input language that has been used at several computer music centers. Since the program is in FORTRAN it is not machine dependent. SCORE can be adapted to create note lists for a variety of computer sound generating programs.

With SCORE it is possible to use much of the standard terminology of music when dealing with pitch and rhythm. However SCORE by no means limits one just to the tempered scale and conventional rhythms. This discussion will give an overview of the principal features of SCORE with emphasis on the program's newer features. These include an added capability to aid in the simulation of musical phrasing and increased flexibility in regard to the use of several different procedures in a single parameter.

The Stanford sound generating system in use from the late '60s until 1979, evolved from a form of Max Mathews' MUSC4 into what came to be known as MUSIO. In the late 1970s Stanford put into operation Peter Samson's real-time digital synthesizer, the 'Samson Box.' These two systems, along with the well-known MUSICS and several other computer music systems, use for input some kind of list of "instrument" definitions followed by note lists which include a name or number which refers to a particular instrument followed by a list of parameters which convey detailed information about the nature of each sound to be produced. In a very simple case there might be as few as four parameters: begin time, note duration, pitch, amplitude. However we often work with instruments which have as many as 99 parameters. These added parameters will refer to things such as quadrifonic position, envelopes, vibra, reverberation, glissando limits, various ways of creating different timbres, etc., etc.

A rather short piece of music can include more than 1000 notes. Clearly then we are faced with a massive typing job unless we can get the computer to help us to deal with the many redundancies which are inherent in music. SCORE has been created for the purpose of facilitating input to the computer of data for any musical style.

For example, the begin time (parameter 1) of each successive note, or sound event, for a given instrument is usually dependent on each preceding note duration (parameter 2). Thus, after stating an instrument's initial begin time, PI is always generated automatically by SCORE. Any parameter that is to remain unchanged need only

220
be stated once. If all the notes in a single voice were to be .1" long, the statement
P2.1s

would generate the durations for thousands of notes.

However we usually want a little more variety than that, at least in the parameters of rhythm and pitch. SCORE allows you to specify strings of input for any parameter. The required syntax for this is a parameter number be given, followed by one of the special code words and then the desired sequence of data. There are several code words that can be used in this regard. I will give brief examples of the three most often utilized, NJM, RHY, NOTES

A string of time values, in seconds, might be entered as follows:

P2 NUM/.5/.2/.1/.3/.6/.2/.1///4

The slashes are delimiters, and when no new number appears between slashes, the last one seen is repeated. The semicolon is always the terminator. There are several features available to indicate various kinds of repetition.

P2 NUM/.35/.2/.08/.1/.9

In this case the duration .12" will repeat eight times.

Usually when we use the code word NUM we are dealing with real time values which will not be transformed in any way. However if we prefer to enter time data in terms of musical rhythmic notation we will be dealing with relative time values which become real time values only after being multiplied by a tempo factor. The default tempo factor in SCORE is 1, or, in musical terms, MA=60. i.e., there will be 60 quarter notes played per minute - one per second. This mode of input is invoked with the code word RHY. Now the numbers entered will be the denominators of musical fractions; /4/ is a quarter note, /8/ an eighth, /1/ a whole note, etc. A dot can be placed after any number to increase its time value by 50%. (Several dots can be added if desired) However dots followed by numbers are taken as decimal points. /5/ represents a double whole note. /5/ represents a dotted double whole note.

Any complex rhythmic relationship can be expressed. A half note tied to a sixteenth would be /2.16/. An ordinary triplet is a /12/ (there are twelve of them in a whole note) a quadruplet is a /25/. Each note in a group of seven in the time of a dotted quarter would be a /18.6667/. (This latter is arrived at by means of a simple formula:

R=94/T

n is the number of equal units to be heard in the time span of T quarters. n the example above we have n=7 and T=1.5; 18.6667=7*4/1.5.)

Of course all these numbers are relative time values. SCORE produces the real time values by dividing the number into 4 and then dividing the result by the tempo factor. Thus the real time value of an eighth note at a tempo of MA=60 is 1/8 second.
At a tempo of MA=120 it would be: (4/8/2x=25 second). SCORE provides the capability of frequent tempo changes, ritardando and accelerando at any rate. There is even the possibility of having different streams of tempo changes in each voice. The tempo feature can be used both with the RHY and NUM code words, but of course the data given after NUM will also be treated as relative values which will be transformed by the tempo factor.

At Stanford, P3 is usually reserved for pitch input. SCORE has three different modes for pitch input. The NUM code word can be used if frequency numbers are to be used. This mode would have to be used for micro-tonal scales of an irregular (non-tempered) nature. If the conventional tempered scale is used the ordinary letter names of the notes are used. A suffix of '1' is used for sharps and 'F' for flats. The octaves are numbered by a system which sets the lowest C on the piano keyboard as C1. This makes middle C into C4 and the highest C into C6. It is easy to keep track of these by remembering that the low C of the cello is C2, the low C of the viola is C3, the high C of the bassoon is C5, the high C of the trumpet is C6, and the high C of the flute is C7. Once an octave number has been stated it need not be typed again unless the notes move into a new octave range. The code for a mid-range D scale would appear as follows:

P3 NOTES/ EF4/ F/ G/ A/) BE/ C5/ D/ EF

A third mode of pitch input allows for the establishment of microtonal scales made up of any number of equal tempered steps to an octave. This kind of input requires three parameters. The fundamental pitch is given in P3. This can be stated as either a frequency number or the letter name of any note of the ordinary twelve note scale. Then in two higher parameters we must give the number of equal divisions of the octave and a list of the particular scale steps desired. In the following example we will give the code for a seventeen tone scale starting from a basic pitch of 440 hertz.

P3 440;
P21 MICRO 17;
P20 NUM/0/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17;

Both the basic pitch in P3 and the scale division number in P17 can be changed as often as desired. In P20, minus numbers can be used for notes which descend below the basic pitch.

Time does not permit a full description of all the features of the SCORE program. (Much of this is covered in my article "SCORE - A Musician's Approach to Computer Music" which first appeared in the Journal of the Audio Engineering Society, January 1972, and later in NAMUS West, 1972) however I would like to give examples of a couple of the newer features of SCORE.

One of the most difficult problems in computer music is to achieve good, musical articulation and phrasing. SCORE helps to solve this problem by giving complete control over the "duty factor," that is, the actual duration of each note irrespective of the perceived rhythm as defined by the time interval between attacks. The time values given in P2 are the intervals between the attacks. The actual note durations are established by use of the code DF used in conjunction with other SCORE code words. Take for example the conventional musical situation where we find a string of quarter notes with the
indication, "poco a poco piu alta!" Here the attack interval remains constant but the sound duration of each note becomes less and less. This could be notated in SCORE:

\[
P_2 \text{RHY} / 4 \times 12 ; \\
P_{12} \text{DF NUM} / 1 / 53/ 86/ 79/ 72/ 66/ 58/ 51/ \\
64/ 67/ 37 / 37.
\]

In this case the duty factor list will serve as a simple multiplier applied the real time values produced by \( P_2 \). For the third note the output of \( \text{SCORE} \) will give \( P_1 \) as time 2 even though the previous note began at time 1 and the value 20 was found in \( P_2 \). In other words, a rest of .07 seconds has been inserted at the end of the second note.

It is also possible to create a legato effect by increasing the duty factor to a number greater than 1. Since the actual duration of the note is greater than the time between attacks, this causes any given note to overlap with the following note. In the next example the MOVE feature causes DF to gradually decrease from 4 to .08. This will cause the passage to begin with extreme legato and progress to the point where the notes are so short that they are heard almost as clicks.

\[
\text{TODT} 0 20 ; P_2 .12 ; \\
P_9 \text{NOTES} / E F F _1 \\
A_4 6; P_6 F_4; P_8 F_1; \\
P_9 .05; \\
P_{10} \text{DF MOV} / 16 4 .08; \text{MOVE exponentially from 4 to .08} \\
<.08 during 16 seconds.
\]

In the cases described above the duty factor was a simple multiplier. There are several other ways of using the duty factor. A fixed amount of time can be added or subtracted from the notes in a string of varied \( P_2 \) values. Or the sound duration of the notes can be set to a constant without regard to the time between attacks. This last can be especially useful when dealing with undamped bell-like sounds. The following example is made up of glockenspiel sound. The wave form used is a simple sine wave and the envelope is an exponential curve. The rhythm of the attacks includes fixed values, 20th and 12th notes, etc., as well as some random elements. (This will be explained below.) The pitches and amplitudes are chosen at random from fixed ranges. The duty factor is established in a user-added subroutine which causes the sounding time of each note to be dependent on the randomly chosen pitch. The higher the pitch, the shorter the note.

\[
\text{TODT} 0 24 ; P_2 \text{RHY} / 20 X 5 / P_{12} / 12 // P_{12} / 36 / P_{12} / \\
P_9 1 / 04 / C7; P_4 1 / 05, 6; \\
P_8 F_6; P_8 F_1; P_9 .05; \\
R D .1; \\
P_{11} \text{DF SUBR} / \\
P_{12} / 1, 4; \\
<\text{Random deviation of attack} = \pm \text{or} = \pm .1 \text{.} \\
<\text{Call subroutine to set the duty factor.} \\
<\text{Choose random numbers between 1 and 4.} \\
\text{END};
\]

\[
<\text{Crucial lines of subroutine.}> \\
X=P(20-69) * (X<55 (C5-41) \\
* X<10X<0 \\
* \text{USES RANGE CCSS TO DS7 (BB)}
\]

229
IF X > 20 X = 20  ; SETS NOTE DURATION
PUPAR1 = 207 - 5 X / 20 ; FROM 2" TO 7"

In the example you will notice that P12 appears at three places in the rhythm list. When SCORE sees a secondary parameter designation within one of its lists, it then places whatever the current value of that parameter may be in the list. In this case the effect was that a rhythm of longer, varied duration was inserted at several points in P2's string of values. This is a bit like using the fermata sign in conventional musical notation.

In the next example secondary parameter calls are made in the pitch list, P3. First of all a string of 10 tempered scale notes is established as motive "Z". Following this, P12 is called upon five times in succession. Looking down P12, we find that this will produce randomly chosen notes between F# and D#. Then the statement /eqz-1/ brings back motive "Z", but 1/2 step (-1) lower. After this P13 will be called upon for the next 7 notes, random frequencies between 1200 and 1400 hertz. And so forth.

<Tape example 3>
TOOT 0 177.1, P2 -1 . ; Will play 177 notes.
P4 5 P6 8 P8 Fi 5 P9 JS P10 OF HMAC 1/10 /5/0 2/10 /1.5/1.3/1.1/9.9.7/ 5/-2/-0.7/ P12 1 F6 D7 P13 1 1200 1400.
END.
TEMPO/ 70. ; Converts attack durations to .085"

The last tape example incorporates all the features I have discussed. The timbre input for all the notes in this as well as all my previous examples, is a simple sine wave.
I hope this will serve to demonstrate that a relatively wide variety of sounds can be attained from very limited means. In all, three different envelopes are used, a sine wave, and the features of the SCORE program.

<Tape example 4>