From "Music V" to "Creative Gestures in computer music"

Daniel Arfib & Loïc Kessous
CNRS-LMA
31 Chemin Joseph Aiguier
13402 Marseille Cedex 09
France

email : arfib@lma.cnrs-mrs.fr

abstract : Music V is one of the first programs dedicated to sound creation. During the seventies, computers were rather slow, and the possibility of performing real time sound synthesis on the computers available at that time was only a dream. However, many concepts were already being developed based on concepts such as curves, musical intention, mapping, and data reduction, which the latest computers are now able to cope with.

Thanks to the latest developments in computer music software and hardware, it is now possible to recreate these sounds in real time and to interpret them. The idea of interpreting computer music (including sounds stored in archives, where appropriate) via gestural control devices gives rise in this article to some reflections on the meaning of "interpretation".

Introduction :

Music V is a sound synthesis program which was originally described by Mathews & al in their book on "The Technology of Computer Music". Music V is in fact a computer program that uses the data provided by the user to produce a digital sound file. It provided the pioneers of computer music with a useful tool, because it did away with the need to write the entire code to calculate a sound by introducing what were called "unit generators".

The first versions were written in Fortran, but some of the more recent versions (which were developed about 10 or 20 years ago) were written in Pascal and C for home computers (DOS systems and Macintosh). Some sound archives make use of this syntax alone, and this can be said to be an advantage because it makes for stability with time (scores dating back 30 years can still be processed making only a few minor modifications). All that will be said below about music V can of course also be applied to the Csound style of writing, but we carried out this study only on Music V scores.

It is tempting to look at previous musical compositions to see whether it might possible to interpret the sounds in a different way. This approach might also contribute to research on gesture, because it involves looking for the link between the musical goal in mind and the appropriate gesture, unlike the approach usually adopted in developing gestural control interfaces, which involves starting with the mechanical device and working towards the sound production.

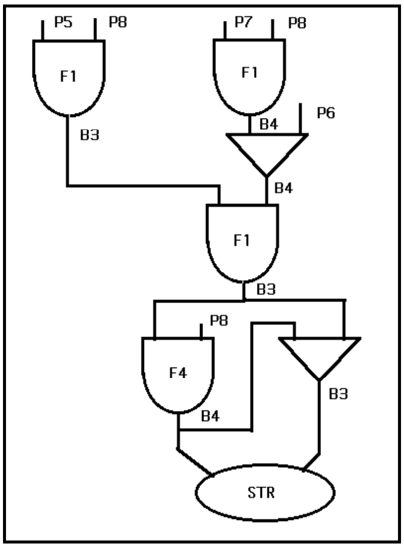
1 Music V, an overview.

The basic Music V syntax is subdivided into INS , GEN and NOT instructions. PLF functions provide additional help when composing with Music V

1.1 INS instructions

INS instructions define a model of a sound but do not play it, just as an instrument maker is not the same person as the player. You devise a program by linking together various modules using a specific syntax. Max-MSP (working in real time) programming is very reminiscent of this kind of programming, except that it uses a graphical interface instead of a numerical one. A Music V Instrument can be defined by its input values (the P values), which are fixed. In most implementations, less than 30 parameters need to be used.

```
COM-----------------------------------;
INS 0 1;
P9_HZ(W9) P6_HZ(W6) P7_HZ(W7) P8_HZ(W8)
P5_W5/4;
IOS P5 P8 B3 F1 P30;
IOS P7 P8 B4 F1 P29;
AD2 B4 P6 B4;
IOS B3 B4 B3 F1 P28;
IOS B3 P8 B4 F4 P26;COM DTE GCHE;
SB2 B3 B4 B3;
STR B3 B4 B1;END;
COM----------------------------------;
```

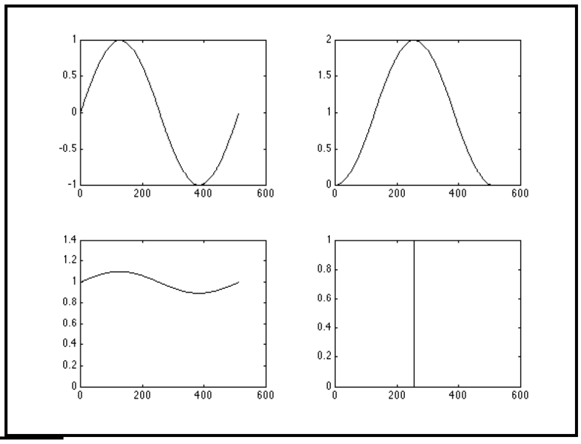


instrument definition in MUSIC V *A possible representation*

1.2 GEN instructions

INS definition depends on GEN functions which define curves. It is often the accurate definition of these curves that gives synthetic sounds a certain style or musicality. There exist various ways of creating functions (by summing sinusoids, by interpolating linear segments, exponential segments and so on....)

```
COM----------------------------------------.
GEN 0 2 1 512 1 1;
GEN 0 2 2 512 .499 0 -.499 0;
GEN 0 2 3 512 1 .1 1;
GEN 0 1 4 512 1 1  1 255 0 256 0 512;
COM-----------------------------------;
```



function generator in MUSIC V *A representation of these functions*

1.3 NOT instructions

NOT instructions provide the means of activating the Instrument However this statement might be a little misleading: for example, if the instrument is only a sinusoid wave maker, it is by simultaneously playing many notes that a sound can be obtained; whereas other Music V INS definitions can be used to produce sounds such as chords, noises, or other complex sound structures with a single NOT instruction.

```
NOT 0 1 28 1000 MI 32 .5  6.235;
NOT 2.2 1 28 1000 MI 32 .5  6.235;
NOT 4.4 1 28 1000 MI 32 .5  6.235;
NOT 6.6 1 28 1000 MI 32 .5  6.235;
NOT 8.8 1 28 1000 MI 32 .5  6.235;
NOT 11 1 28 1000 MI 32 .5 6.235;
NOT 13.2 1 28 1000 MI 32 .5 6.235;
TER 48;
```

activation of events in MUSIC V

1.4 PLF (compositional functions)

PLF functions can be used to modify or create NOT instructions. The metronome, for example, is one such function which modulates the time axis by changing the onset and the duration of notes. Other compositional functions can also be written, however, usually in the same computer language as the interpreter of Music V. Once a PLF has been written, it can be integrated by other users.

> "The PLF routines provide one of the most exciting areas for further development in the entire Music V structure. Not only do they promise the most interesting possibilities, but they also offer the greatest challenges for the composer's creativity."
>
> *The technology of computer music, Mathews and al, 1969*

1.5 Music V implementation

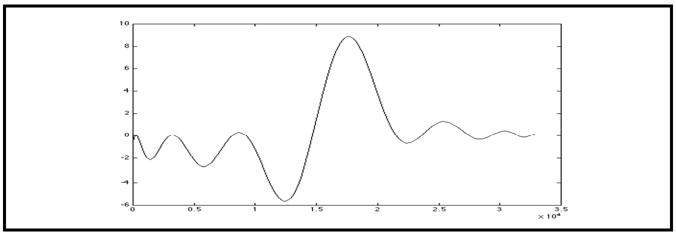
The basic principles on which Music V works are quite simple:
- Pass 1 is a syntax check and a music V score is translated into an intermediate code which expresses the operation involved and the corresponding data numerically
- Pass 2 is a sorting program, which also has possible compositional functions.
- Pass 3 is the executive program. Although the user does not have to know how to program a module to extend the language, Music V has of course been extended by many programmers and musicians for specific purposes.

Music V was initially only a sound synthesis program. But by adding new units to Music V, it has become possible to use recorded sound. The first of these new units obviously had to be a sound reader .Those providing FFT decomposition ~~,~~ or wavelet oriented unit generators have made it possible to perform some sound transformations that would otherwise be quite impossible. Introducing delay lines, for example, made it possible to design reverberation units.

Music V used to be an off-line program (since up to 24 hours of calculation were required to produce sounds lasting only a few seconds). The Risset catalogue was one of the first sound recipe books and is still a standard reference for this kind of programming. As far as France is concerned, many different versions have been developed at IRCAM, INA-GRM and CNRS-LMA. In Italy, some versions have also been developed, including even some with a graphic interface. However, work on these versions has not continued as it has in the case of Csound .

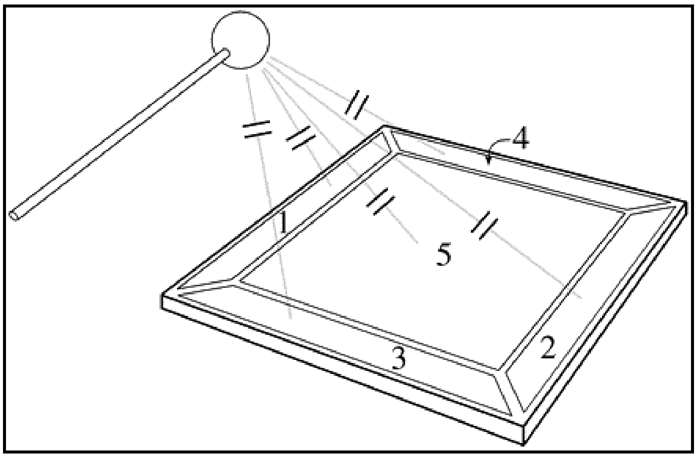
2. driving sounds using curves and gestural control

Curves are a really important part of this system. Music V programs are systematically based on making the right choice of curves. Risset's famous ever descending glissando is a typical example of the use of very precise curves. The double non-linear waveshaping procedure used by Arfib also relies on perfectly shaped index functions. One might therefore say that to make good computer music, it is necessary to have good algorithms, but extra-good functions are also required. These functions can be either time-dependent variables or relations between two variables.



A curve used in waveshaping instruments

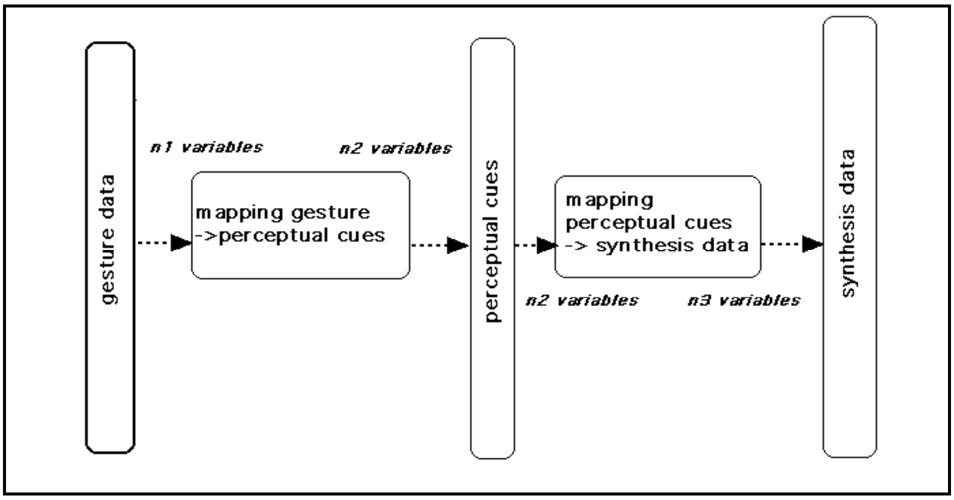
FM synthesisers such as the DX7 model prove this point: starting with the same algorithms, many different and intriguing sounds can be produced using different curves. This suggests what a wide scope thus becomes available to the user for interpreting or expressing these Music V sounds: by changing the curves, one can modulate the sound.



radio drum technology

On the other hand, gestural devices are often used to control sound computed in real-time. Linking gestures with sound consists in setting up a correspondence between specific movement parameters and sound synthesis parameters. For example, it is possible to drive an additive synthesis consisting of 64 partials with a radio drum stick giving the X,Y,Z position of the end of one stick.

Mapping gesture onto sound is a really vast topic. Very basically, there are two possible ways of performing this operation using either a time-independent one-to-one relationship or a decision making method (triggering events in terms of movements, for example). A considerable body of literature has already been written by various authors on this subject.



mapping gesture to sound via perceptual cues

One efficient (and practical) method which can be used here consists in building an intermediate set of values which can be called 'perceptual cues'. These are neither synthesis parameters nor data originating from the gestural device. They are a musical or perceptual set of data that can be linked to sound as well as to gesture. Displaying their values to provide the performers with visual feedback can also be of great help during a performance involving the use of gestural devices.

3. interpretation in computer music.

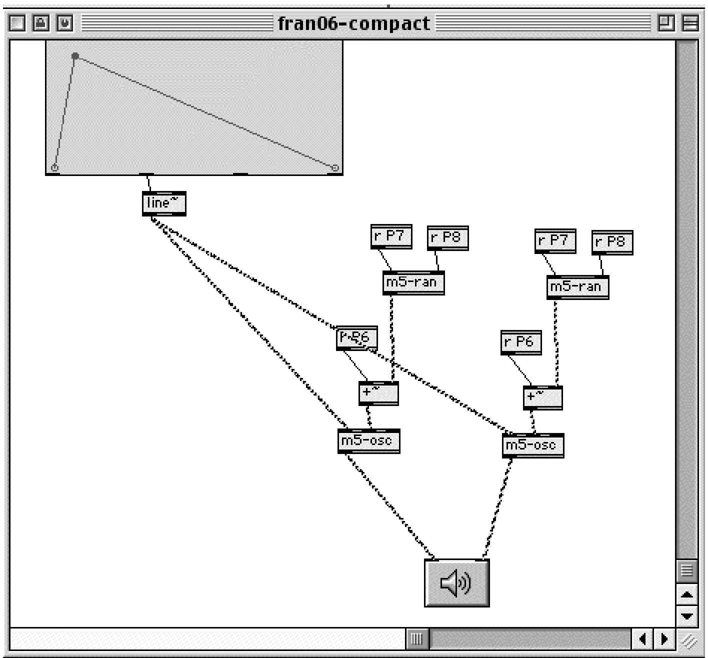
The usual way of thinking about gestural control is to start with the devices, looking at their output and their degree of freedom and trying to invent sounds or sound structures that can be controlled by the gestures. But one can also take another point of view: starting with a sound recorded on a CD and asking oneself how it might be possible to make variations on it and make it more expressive (in other words, how to interpret this sound) and what kind of devices might help to do so. This is the approach we have taken here, but an additional factor is also involved: we can have not only sounds but also the definition of how to produce the sounds in the form of a "score" (in Music V, the syntax "score" defines the design of a virtual instrument as well as its use).

Investigating how to interpret computer music written in the old way (non real-time programs) is a challenge, however, from the pragmatic point of view, where does the control need to be focused? and from the philosophical point of view, what is interpretation?

The term "Interpretation" can be defined on different levels:

- even a single sound can be interpreted, just as a violinist can perfect the emotion underlying a sound. In this case, it is mostly the timbre that can be altered by making the appropriate gestures.
- a sequence of notes can be interpreted. The analogy with MIDI is evident. In the case of MusicV- like programs, a set of NOT instructions defines the onsets, durations, and other characteristics of the sounds (for example the attack, decay, sustain and release times can be specifically changed).
- some sound creations can be more easily described in terms of "digital audio effects", or sound transformations. The example of reverberation can help to understand how a synthetic sound can be interpreted simply by changing the reverberation time or the pre-echoes of a reverberation.
- the parameters of a generating program can be re-set to produce different interpretations. The PLF functions in Music V are of this kind. Mathews in his fundamental book has already given some details on how to write and use some 'compositional functions' The example of automatic arpeggiation can help us to understand how time delays as well as frequency steps, or the choice of specific scales can be used to obtain variations on a computer music score.

4. Creative gestures in computer music.



a Max-Msp patch from "variations du Souffle du Doux"

The aim of the project named ***"Creative gestures in computer music"*** is to duplicate Music V sounds in the Max-Msp programming language with several goals in mind: first, to obtain identical sound results, and then to provide additional control modes to be able to modulate and perform these sounds live. The ultimate goal is to be able to produce music that can be either computed as such (and will thus be identical to the archived sound) or modified in real time to produce a variety of performances.

Two phases of this project have already reached an advanced stage: the "gesture rendering" of an entire computer music piece, and the constitution of a museum of computer music sounds.

4.1 the gesture rendering of "le Souffle du Doux"

"Le Souffle du Doux" is a piece entirely written in the Music V language by one of the authors in 1979. It was recently transcribed into the Max-Msp language, so that an identical copy of the original can be obtained. The NOT instructions are transformed into lists that are activated at the appropriate time, and the INS instructions are transformed into a Max patch using MSP objects. GEN functions are either part of the "function object" if they are linear interpolation functions, or else stored in a pre-calculated buffer.

Then different strategies are used with each sequence to modify the patches in order to make them playable with external devices by a performer. These strategies do not depend only on the device itself but also on the mapping method chosen.

These modifications can be of two kinds:
- decision gestures (such as hitting a radio drum) trigger sounds and define their parameters. The onset times and the durations and values of the other "NOT" parameters can be associated with the gesture. The noisy percussive sounds which occur in le Souffle du Doux can be played with different pitches, noise bandwidths and decay times, for example.
- modulation gestures create data that either replace the values of an existing MusicV function or act like an algebraic operation on an existing function. The gestural data are usually delivered by peripheral MIDI devices. These gestures can result in each performance having different interpretations. For example, the changes in the sound textures which can be obtained using a double distortion scheme in "le Souffle du Doux" have been mediated via either the 2D surface of the Max radio-drum, or linear and/or rotating sliders. The mapping is carried out by means of software tables that set up a correspondence between the position and the distortion parameters.

Rehearsals are required to learn how the various gestures are liable to affect and the sound output and to discover new ways of expressing the score. "Variations sur le Souffle du Doux" results from this part of the project.

4.2 A museum of computer music sounds

A set of a dozen sounds has been extracted from archives (along with their Music V definitions): they all differ as far as their musical character, timbre control, sequence of notes, use of PLF, and mode of sound processing are concerned. They include some of the archetypal examples featuring in Risset's catalogue. The musical purpose and the corresponding gestural intentions are assessed and possible lines of interpretation are defined .

For each sound, the possible mapping strategy between the synthesis parameters and the gestural data is tested using some special devices (a radio baton drum, a graphic tablet, a set of linear and rotating potentiometers plus specific gestural devices such as the "Bao Pao"). Some sounds suggest the need for new peripherals and conversely, some gestural devices suggest the need to make changes in the synthesis algorithms

Work on this project is now in progress. The results are very encouraging, because they raise new questions about the nature of gesture and about the musical functions unusual sounds are intended to fulfill.

Conclusion

The most original feature of this research project is that starting with the sounds and our knowledge of the computer music composer , we have attempted to link sound production with gestural devices. This approach has led us to re-define what interpretation itself consists of, and the answer will then be introduced into the sound synthesis algorithm. In addition, the various gestural devices tested will be compared not only in terms of their efficiency but also in terms of how appropriate they are as means of controlling sound and computer music.

References:

Some of the fundamental books and articles citing Music V:

D. Arfib, Digital synthesis of complex spectra by means of non-linear distorted sine waves, JAES vol 27, n° 10, oct 79 pp757-768

B. Maillard , Petite pédagogie sur la modulation de fréquence, Cahiers INA/GRM n°3

M.V. Mathews, The technoloy of computer music, MIT Press, Cambridge, MA, 1969

J.-C. Risset, ``An Introductory Catalog of Computer-Synthesized Sounds," tech. rep., Bell Laboratories, Murray Hill, N.J., 1969.

J.-C. Risset, Timbre Analysis by Synthesis: Representations, Imitations, and Variants for Musical Composition, in "Representations of Musical Signals", pp. 7-43. The MIT Press, 1991. G. De Poli, A. Piccialli, and C. Roads, eds.

For modern implementations and gesture control, here are some sites that also have excellent links and bibliography references.

Richard Boulanger
http://www.csounds.com/catalog/

Marcelo Wanderley
http://www.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/

Alex Mulder
http://www.cs.sfu.ca/~amulder/personal/webpages.html

Joe Paradiso
http://www.newmusicbox.org/third-person/index_oct99.html

=============