MARS - Musical Applications

G. Palmieri, S. Sapir

IRIS S.r.l.
Pacco La Selva 151 - 03018 Pallaro (FR) ITALY
Fax. (775) 553343 - Tel. (775) 553441 - E-mail: nicola.larosa@etekom.it

ABSTRACT

The IRIS Musical Audio Research Station (MARS) is a general purpose musical instrument for real time digital sound processing. It does not provide any compositional environment, but its compatibility with the MIDI world allows users to include it in a studio having MIDI facilities. Moreover, its programmability and its powerful communication features allow users to build stand-alone applications. This makes MARS a machine suitable to both the needs of commercial music and the requirements of contemporary music.

As a musical environment, MARS allows users to develop single sound objects called tones. Tones are specific configurations of all algorithm input variables. Specialised editors for the graphical package EDIT20 allow the user to describe static and dynamic settings of these variables. One of the main features of MARS is the capability of assigning MIDI controls to tones by means of programmable and dynamic links between parameters and gestures. MARS also allows musicians to group sets of tones in a performance environment which involves multi-algorithm processing of sounds.

Even though MARS has not yet been commercialized, it has already been used in two important musical pieces (Piero e Andromeda by S. Sciarrino, Ofanim by L. Berio) and for music pedagogy (University of Padova).

1. TONE EDITING

The MARS graphical package EDIT20 and its real time operating system RT20M are MIDI oriented (Andrenacci 92), thus all editing actions and gestural interactions conform to the standard MIDI protocol.

1.1 Parameter editing

Parameters are the single entry points of an algorithm. They have a symbolic name, a default representation unit (such as Hz, db or ms), and a state (unused, static, dynamic) that defines the way their values will be updated.

Static parameters are initialized at the beginning of a note event with a constant value. Dynamic parameters are initialized at the beginning of a note and are continuously updated during the life-time of the note, according to a conversion formula. This formula dynamically links gestural events to physical parameters. The setup of a dynamic parameter is done in three phases:

- **Formula Selection** - One of 17 predefined formulas is graphically displayed as a combination of a maximum of 3 single blocks called M.
- **Block Definition** - Each M block allows a table lookup conversion that depends on its input points, an offset, the content of the table and a scaling factor for output values.
- **Control Selection** - The MIDI event that will be used as the table pointer C of the individual blocks (Pitch Bender in the following example) is selected by the user. The PAR editor is shown in Figure 1 and the formula associated to the equal pitch parameter is:

\[ P_{ich}=M_1[(1+M_2)+M_3] \text{ with } M_1=M\text{SCALE}(Key)^*1.0; M_2=M\text{SLIDER(Cord)}^*24; M_3=M\text{DETUNE(vel-64)}^*0.06 \]

1.2 Low Frequency Oscillator editing

LFOs are defined using the specific LFO editor shown in Figure 1. One LFO definition can be associated to one or more of the algorithm LFO connections. The LFOs are triggered by a Note On message and are released by a Note Off message. Their components are the following:

- **Oscillator** - A table lookup oscillator which can be periodic or one-shot, whose parameters are the periodic waveform and whose period can be dynamically controlled by any MIDI event.
- **Envelope** - An Attack Sustain Release envelope generator, whose parameters are the profile and the duration of the attack
and release parts, and the velocity delay after a Note On (attack, release and delay can be controlled by MIDI events).

Tables used by the LFO and PAR editors can be edited by the TAB editor. The TAB editor is used to define conversion tables or amplitude waveforms using either mathematical descriptions or a set of graphical and interactive tools which manage function primitives (sin, exp, log, shift, rotate, smooth, etc.).

1.3 Envelope editing

Envelopes are generated by segments, which are defined in terms of final and slope (or duration) values. An envelope definition is the complete description of the segments and the points that determine the 3 envelope parts: Attack - This part begins at the first segment and includes a variable number of segments. It is triggered by a Note On. Loop - This part is enclosed between two segment pointers. The segments between when this part is repeated according to the number of loops set by the user. Release - This part begins at a pointer selected segment and ends at the last segment. The envelope is forced to be released when a Note Off message occurs before the envelope is finished (during the attack or loop parts or in the sustain area). The envelope editor allows one to set these parameters graphically and to assign any kind of MIDI events in order to dynamically control segment slopes. This allows envelope stretching or compression during performance.

2. PERFORMANCE EDITING

A whole orchestra (algorithm selection and duplication) can be defined by using the ALGO and ORCH editors of the EDIT20 package (Armata 1992). This orchestra can be configured for a live performance by using the 7MAP editor. Such configuration consists of selecting a set of tones for each algorithm of the current orchestra and mapping these selected tones to MIDI channels. A performance environment can contain up to 128 tones. MARS allows dynamic tone/MIDI channel mapping, by sending Program Change messages during the performance.

A performance environment is described by its Tone Map. An orchestra can be configured many times during the performance. This is realized by calling different Tone Maps that will be loaded through System Exclusive messages.

After having completed the whole configuration of an orchestra, MARS allows real time performance with that orchestra using standard MIDI messages. This feature gives the opportunity to create MIDI applications by using commercial software packages, such as sequencers or arrangers, to include the generation of musical parameters beyond MARS capabilities.

The performance may also be managed by applying, which use a quicker and more powerful communication port. This may be useful when the MIDI protocol is not powerful enough to pile complex algorithms or when it is necessary to combine the musical performance with other field applications.

Figure 1. Puř: Dynamic parameter editor and LFO: Low Frequency Oscillator editor

REFERENCES