REAL-TIME GRANULATION OF SAMPLED SOUND WITH THE DMX-1000

Barry Truax
Dept. of Communication & Centre for the Arts
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6

ABSTRACT

Granular synthesis has been proposed (Roads, 1978) as a unique method of achieving complex sounds by the generation of high densities of small "grains" on the order of magnitude of 10-20 ms duration. Recent work by the author (Truax, 1986) has shown that this technique may be implemented in real-time using the microprogrammable DMX-1000 Digital Signal Processor (Wallraff, 1979). Synthesis techniques involving additive synthesis and simple FM pairs for each grain have been implemented, with a hierarchy of control parameters directing the density, frequency range and temporal evolution of the synthesized sound textures. This paper describes an extension of this work involving real-time granulation of stored sampled sounds as source material. Although the initial implementation uses a very limited sound sample duration (150-170 ms), further work is being done to implement a continuous processing of sound samples.

I. INTRODUCTION

Granular synthesis has been proposed (Roads, 1978) as a unique method of achieving complex sounds by the generation of high densities of small "grains" on the order of magnitude of 10-20 ms duration. Two problems that must be solved for the effective use of this method are:

a) the large amount of data required to specify the sound since typically 2000 grains/second may be involved;

b) that powerful control variables are required to give the user the means to link the lower level data to macro-level compositional strategies and gestures.

Recent work by the author (Truax, 1986) has shown that this technique may be implemented in real-time using the microprogrammable DMX-1000 Digital Signal Processor (Wallraff, 1979). Synthesis techniques involving additive synthesis and simple FM pairs for each grain have been implemented in the GSX program, with a hierarchy of control parameters directing the density, frequency range and temporal evolution of the synthesized sound textures. The author's recent work *Riverrun (1986)* was exclusively realized using these techniques. This paper describes an extension of this work involving real-time granulation of stored sampled sounds as source material.

II. THE GSAMX PROGRAM

The real-time program GSAMX currently implements a sampled sound instrument for granular synthesis where each grain consists of a short segment of sampled sound with specifiable duration and offset time from the beginning of the entire sound sample. A fixed and rather short source sample is currently used, namely 4032 samples or around 150-170 ms of sound, because of the limitation of 4K onboard memory in the DMX. The duration of grains used in granular synthesis is typically less than this limit so the effect of the fixed sample size is to limit the variety of simultaneous "windows" that may be accessed from the sound material.

The synthesis instrument consists of a bank of simple envelope generators with specifiable duration and delay (in ms) between successive envelopes. Each generator produces a three-part linear envelope whose attack and decay portions are a specifiable fraction of the event duration. The attack and decay portions both default to one-quarter of the grain duration; other proportions of attack and decay from 1/2 to 1/16 the grain duration are specifiable (Fig. 1).

Twenty simultaneous "voices" of the sampled sound model are possible with the DMX-1000 as a result of the number of instructions required. Half of these voices are assigned to the left output channel, and the other half to the right. Therefore, all output from the instrument is in stereo. Given the complexity of the sound and the possibility of cancellation, the sound is much richer when synthesized in this way.

The synthesis instrument is controlled by a scheduler program on the host LSI 11/23+ (Truax, 1984) where each grain is initiated (by setting the attack ramp) and terminated (by setting the decay ramp) under clock interrupts every 1 ms. This "foreground" program level also imposes a variable delay time (in ms) until the next grain is started. The shorter the grain, the higher the overall density of grains per second (gps). The minimum grain duration that can be effectively controlled in real-time is 8 ms, hence sound densities up to 2500 gps can be achieved.

Because each grain has an attack and decay, there is no possibility of clicks or transients depending on the portion of sampled sound being used. Moreover, when the grains are unsynchronized (i.e. of variable duration) and/or when each grain starts at a different position within the sound sample, very complex textures can result from even a very simple sound source. Although the initial implementation uses a very limited sound sample duration (150-170 ms), further work is being done to implement a continuous processing of sound samples.

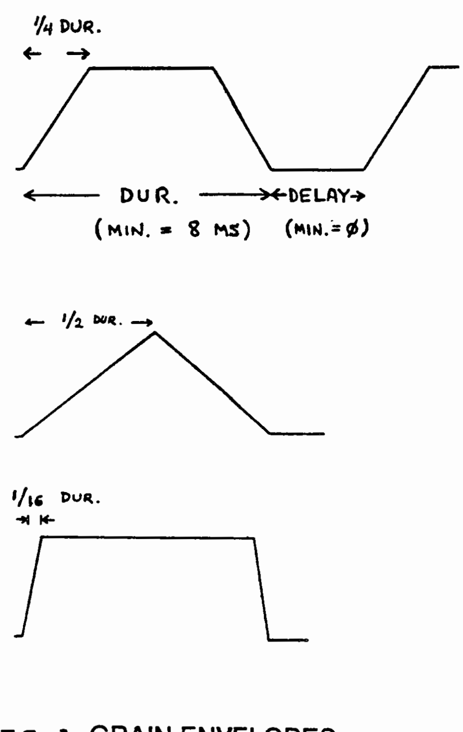

FIG. 1 GRAIN ENVELOPES

III. CONTROL VARIABLES

The control variables available to the user which determine how successive grain parameters are calculated are:

- offset number of samples and offset range
- average grain duration and duration range
- delay time between grains

- speed of output which acts as a pitch/time transposition
- number of voices sounding at transposed pitch levels
- total number of voices sounding (max.=20)

The background level of the scheduler has two functions: to service control requests from the user and to calculate new random values for the parameters (offset and grain duration) when the range of their variation is non-zero. For instance, if the average duration is 30 ms and the duration range is 10 Hz, grains will be calculated with durations between 26 and 35 Hz, distributed linearly. Any calculated duration which is negative or less than 8 ms is adjusted to this minimum value. Such recalculation of new random events occurs constantly whenever foreground activity or user controls do not demand the processor's attention. Therefore, there is no possibility of an exactly steady sound. All sounds have some degree of a granular texture because of random variation in grain specification, hence the lively and sometimes natural sounding result.

No variation in grain duration (i.e. duration range equals zero) produces an amplitude modulated signal, whereas even a small range of variation results in a stochastic texture. As predicted by Roads (1985), amplitude modulation (AM) results because each grain is immediately followed by another of the same duration; therefore the grain duration becomes the period of the modulating wave, and the grain envelope its waveform. For instance, the minimum duration of 8 ms is identical to a modulating wave of 125 Hz. Sidebands are thus produced around the centre frequency, and if that frequency is harmonically related to 125 Hz, an enriched harmonic spectrum results. It may be noted that even in the case of no variation in grain duration and frequency, the resulting sound is still not steady because each of the voices in the instrument cannot be exactly synchronized at the micro level. That is, by the time the last voice is initiated, even during the same clock interrupt, the others will have already started and therefore will be in a different phase relation to the current voice, thus producing a variable output amplitude because of cancellation and reinforcement.

The delay time between grains can also be used in conjunction with the AM effect since it changes the overall periodicity of the resulting sound. The minimum delay time is 0, and as it increases to a significant fraction of the grain duration, various modulation effects are heard. Longer delay times result in a lessening of the sense of texture (since fewer grains/second are being heard), until with very long delay times, the grain can be heard as a separate event.

Three other control variables are also available. The first is a simple speed control implemented by including "no operation" instructions in the synthesizer's program that cause it to run slower, similar to a variable speed tape recorder. The second option allows a certain number of voices in the instrument to output samples two times faster than normal (by skipping alternate samples) or two times slower than normal (by repeating every sample twice). As a result, part of the sound texture may sound an octave above or below the rest of the material. The third option turns voices on and off, thereby allowing changes in overall density. Of these three variables, only the speed may be automatically ramped; the others are toggled up and down by the user in unit increments, even while other ramps are in operation.

IV. REAL-TIME USER CONTROLS

Several modes of real-time user control are available during synthesis:

- a new value may be typed in for any parameter
- a single parameter may be changed by a specified increment
- a group of "synchronized" variables may be changed by a specified increment where the synchronization can represent a direct or inverse variation of each variable
- all parameters or only the "synchronized" parameters may be reset to any of a group of stored (preset) values
- ascending, descending or random ramps may be initiated on all of the "synchronized" variables according to a time or "ramp" value which represents the rate at which a specified increment is to be added to or subtracted from each variable; the ramp value itself may be one of the synchronized variables, thus allowing acceleration and deceleration of the ramp; ramps may also be scaled to proceed at different rates, ranging between factors of 1 through 10
- overall amplitude level may be set, and a global attack and decay initiated at any time at a predetermined rate

It should be noted that all of these control possibilities can be activated by single CRT keyboard strokes during the synthesis. All controls are compatible with each other and therefore can be executed in conjunction with them. For instance, individual parameter changes and resets can be executed during ramps; likewise, presets can be stored during a ramp. Similarly, variables can be synchronized, removed from synchronization, and the nature of their correlation (direct, inverse) switched during ramps. At any point, the user can type in a new value for any parameter (including the ramp time).

A line of control parameters is displayed on the CRT as shown in Table 1.

TABLE 1. Line of Control Variables

AMP	INC	OFFSET	OFF.RNG	DUR'N	DUR.RNG	DELAY	RAMP	SPEED	NO.VOI. TRANSP.	TOTAL NO.VOI
	1	1000	200	20	10	1	1000	210	0	19

In the example in Table 1, INC is an increment value to be added to or subtracted from any variable; OFFSET is the average offset value from the start of the sound segment (in number of samples); OFF.RNG is the range around the average offset; DUR'N is the average duration (in ms); DUR.RNG is the range of durations; DELAY is the delay time between grains (in ms); RAMP is the time (in ms) before the INCrement value is added to a variable; SPEED changes the rate of output of the samples; NO.VOI.TRANS is the number of voices having the samples synthesized faster or slower than normal; and TOTAL NO.VOI. is the number of voices sounding in the instrument. The AMP indication is the global amplitude value which only appears when requested.

Variables that are singled out for "synchronization" are so indicated with a + or - sign appearing before the number. The former indicates that with the up arrow, as well as with the ascending or random ramps, the INC value will be added to the parameter. The latter indicates that with the up arrow, ascending or random ramps, the INC value will be subtracted from the parameter, provided it does not go below permissible limits. With the descending ramp or down arrow, + variables are decremented, and - variables incremented. In general, these signs indicate direct or inverse variation during a parameter change, and they may be used in any combination. Synchronization is removed with the DELETE key. In the case of the random ramp, only a fraction of the INC value (from 0 to INC-1) will be added or subtracted according to the + or - sign. Because not all parameters change synchronously with the random ramp, its use generally produces smoother changes than the other types of ramp.

Variable rates of ramp increment are requested with the keyboard commands SHIFT 1, 2, 3, ..., 0, that is, the upper case numerals, with 0 meaning 10. Therefore, if the INC value is 2 and the rate is 5, the parameter will be incremented by 10 with + synchronization, or decremented by 10 (assuming the result is permissible) with the - synchronization. This option is particularly useful with offset and ramp values, since they often need to be changed by larger amounts than do the duration and delay values. The ramp rate (if not equal to 1) is printed immediately to the right of the number. A typical control line might therefore appear as in Table 2.

TABLE 2. Line of Synchronized Control Variables

AMP	INC	OFFSET	OFF.RNG	DUR'N	DUR.RNG	DELAY	RAMP	SPEED	NO.VOI. TRANSP.	TOTAL NO.VOI
	1	+1000 5	200	20	+10	1	-1000 2	210	0	19

The specification shown in Table 2 means that the OFFSET and DURation RaNGe values will be incremented with an up arrow or ascending ramp, and the RAMP value decremented by the same action. However, the rate of change of the offset value will be 5 times greater, and that of the ramp value 2 times greater, than that of the duration range. Therefore, after a single positive increment, the new offset value will be 1005, the duration range 11, and the ramp value 998. The variable ramp rate can be removed from a parameter by setting it to 1 (i.e. the SHIFT 1 command). Since the INC value can also be changed during an ascending, descending or random ramp, quite complex changes can be easily implemented.

Cyclic Ramps

Any ramp (ascending, descending, random) may be made cyclic with the C switch (indicated at the left beside the A, D, or Q for the ramp). This means that when the parameter value reaches the minimum value, the direction of the synchronization (+, -) is reversed to the opposite mode; likewise when the maximum value is reached the synchronization is reversed again. The minimum value is 0 for most parameters, except grain duration and ramp speed where the lowest possible value that can be realized is enforced. The current maximum value can be displayed (M) and a new value typed in for that parameter. Both minimum and maximum values can be specified in the ramp files (see below). Should the current parameter value be outside those limits, a reversal of synchronization brings it back within them.

Envelope Shape

As indicated earlier, the envelope shape is controlled by a single number (2-16) which indicates the fraction of the total grain duration devoted to the attack and decay portions. In terms of the amplitude modulation situation (little or no variation in duration range), this fraction controls the waveshape of the modulation, going from a triangle wave (attack = 1/2 duration) to approximately a rectangular wave (attack = 1/16 duration). Hence it is an important timbral variable, and its value can be changed with the up and down arrows once the current value is displayed (CNTRL Q) in the far left position.

With sampled sound as a source, the presence of an envelope prevents transients from occurring when an arbitrary group of samples are chosen. The envelope of each grain may be thought of as a fade-in and fade-out; therefore, changing the envelope shape determines the smoothness or abruptness of this effect.

V. PRESETS, OBJECTS AND SCORES

Any keyboard characters that are not reserved for special meanings can be used to indicate a set of current variables and to store or retrieve those values from temporary storage in memory. With the sampled sound model, these presets include the six variables (offset, offset range, duration, duration range, delay, and ramp value).

The preset values are retrieved whenever the keyboard character is typed and stored with CNTRL plus the same character. Storage of presets can occur during a ramp when values are caught "on the fly"; they may later be retrieved once the ramp is stopped or while it is running. To edit a preset, the desired variables are changed and the identifying character (plus CNTRL) re-typed. The user can store up to 17 such presets, and with CNTRL S, all such presets are stored in a diskfile (*.GRN) with no interruption in synthesis. CNTRL R retrieves the values stored in the file, prints them on the CRT, and makes them available to be called. These values may subsequently be edited, added to, and stored again.

Each preset can be thought of as a "sound object," similar to the timbral definition of a sound object elsewhere in the author's POD and PODX system (Truax 1977, 1985). A score editor program (PDFILG) has been created which can combine up to 160 presets into a single file, along with data about how these objects are to be scored in time. The score data that is used for each performance is:

- entry delay (in centiseconds)
- object number
- maximum amplitude

The user has the option of including a duration value (in ms) and a offset value (in number of samples) in the score specification, either of which may override the corresponding duration and average offset value in the object. Therefore each score may be performed in four different ways (with or without replaced duration or offset values). Similarly, each score may be performed with different speeds and different envelope shapes for the grains. During the performance of such a score, ramps may also be used at the same time, either manually activated or from a ramp file. Therefore, the score replaces the live performance aspect of playing the instrument by calling back presets, but it still allows considerable real-time modification during synthesis.

VI. RAMP FILES

To facilitate more precisely controlled ramps, an editor has been included that allows ramps to be specified, stored in a file (*.RAM), and recalled and implemented during synthesis (with the I command). Up to 18 sequential ramps may be specified in this way. Each ramp includes:

- ramp type (A,D,Q)
- ramp scaling (1-10) for each synchronized parameter
- synchronization (+, -, or none)
- cycle switch
- end condition (elapsed time or specific parameter value, whichever comes first)

The ramp editor displays the default ramp values when first entered, or those already stored in the current file. With the E command, the user advances to the end conditions for each ramp, which as indicated above, are either the elapsed time (in ms) or a specific parameter value that is to be reached. When both end conditions are specified, the one reached first will result in the next ramp (if any) being implemented. Note that with cyclic ramps, only the elapsed time value as an end condition allows the ramp to cycle through various values. With the M command, the user can see a list of the minimum and maximum values for each parameter during a cyclic ramp. These values may be edited and put into effect whenever a cyclic ramp is specified.

When no ramp change is desired for a certain period of time, no variable needs to be indicated as synchronized, although a ramp type (A, D, Q) still needs to be specified. Ramp files may be used with either the real-time programs (GSX and GSAMX) or in the context of a score playback, and at any time a ramp may be altered by any of the real-time user commands. Note that the speed of the ramp (controlled by the RAMP variable) is stored in the preset or object; hence the same ramp file will behave differently (i.e. at different speeds) with different objects.

The current implementation, while just a beginning, has already gone some distance in establishing an appropriate hierarchy of levels of compositional control, ranging from the control variables at the grain level, through to groups of such variables (presets), rates of change (ramps) of the control variables, and macro-level scores to determine large-scale forms.

VII. MUSICAL APPLICATIONS

The complexity and dynamic quality of granular synthesis sound makes it an attractive alternative to synthesis models based on fixed waveforms and envelopes, and to sampling instruments based on transposition of stored sound samples. Moreover, the basic unit or "quantum" of the grain is a very flexible means of manipulating sound samples, particularly because the envelope of the grain avoids transient clicks when extracting and combining sample segments. When granular synthesis is used to produce continuous textures, it has no resemblance to instrumental and other note-based music; instead, its sound world is more closely related to analog electroacoustic music, but with greater precision of control. In certain cases, the acoustic result resembles environmental sounds in terms of their inner complexity and statistical texture. However it is used, granular synthesis is clearly situated in a different psychoacoustic domain than that occupied by most computer music; it creates a unique sound world and suggests new approaches to the way music made with it is formed.

The psychoacoustic domain of high density events has recently come to be described in terms of "streaming" (McAdams and Bregman 1979) where events may be perceived as isolated, grouped into streams, or fused together depending on their frequency range and temporal density. More recently, John MacKay (1984) has described increasing densities of events as creating "a spectrum of impressions ranging from the simple 'sequence' of tones to that of a 'flow', a 'swarm', a fused 'textural band' and finally a 'massed sonority' depicting the different degree of density-determined solidity and consistency of the texture." In terms of increasing bandwidth, he describes (p. 171) "a spectrum of impressions of stratification ranging from a noise band to bandwidths with very prominent upper and lower edges but no clearly perceiveable tonal identities in the middle, to bandwidths with very prominent lower edges and mildly prominent upper edges and some fleetingly identifiable tonal content in the middle of the bandwidth."

In my own work, granular synthesis is an extension of what I have called the "stochastic texture" (Truax 1984), as realized in *Arras (1980)*, where the superimposition of many similar and spectrally related sub-events produces a clearly defined and controllable macro-level texture. The presence of any particular frequency component at the micro-level, however, can only be statistically determined. The difference with the present work is that much shorter events and higher densities can be generated such that one passes the audio-rate threshold (around 20 Hz) at the micro-event level.

However, the approach to structuring the sound and the music remains the same, namely a hierarchic organization of levels.

All sounds in my recent work *Riverrun (1986)* were generated with real-time granular synthesis, up to a maximum density of 2375 grains/second. However, in many cases, lesser densities were also used since often the progression from isolated sounds or a rapid sequence of events to a fused texture is the most interesting feature of the synthesis method. Most layers were multi-tracked with four simultaneous stereo versions, and up to four of these 8-track source tapes were later mixed. Considerable use was made of ramps applied to the synthesis variables; that is, certain parameters were made to change over time at a specific rate, sometimes with several parameters simultaneously ramped at different rates. Therefore, all sound in the piece is in a constant state of flux, much like environmental sound generally and water sound in particular, whether through the use of ramps or because of the random variation of the thousands of component grains heard in each sound.

The fundamental paradox of granular synthesis - that the enormously rich and powerful textures it produces result from its being based on the most "trivial" grains of sound - suggested a metaphoric relation to the river whose power is based on the accumulation of countless "powerless" droplets of water. The opening section of the work portrays that accumulation as individual "droplets" of sound multiply gradually into a powerful broad-band texture. The dynamic variation found in the use of ramps allows the piece to create a sound environment in which stasis and flux, solidity and movement co-exist in a dynamic balance similar to a river which is always moving yet seemingly permanent. The piece, I find, also captures some of the awe one feels in the presence of the overpowering force of such a body of water, whether in a perturbed or calm state, and as such it seems to create a different mode of listening than does conventional instrumental or electroacoustic music.

The techniques developed during the composition of *Riverrun* are suggestive of methods for working with granulated sampled sounds as described in this paper.

REFERENCES:

MacKay, J. 1984. "On the Perception of Density and Stratification in Granular Sonic Textures: An Exploratory Study." *Interface* 13:171-186.

McAdams, S., and A. Bregman. 1979. "Hearing musical streams." *Computer Music Journal* 3(4): 26-41.

Roads, C. 1978. "Automated granular synthesis of sound," *Computer Music Journal* 2(2): 61-62. Reprinted in C. Roads & J. Strawn, eds. 1985. *Foundations of Computer Music.* Cambridge, MA: MIT Press.

Truax, B. 1977. "The POD system of interactive composition programs." *Computer Music Journal* 1(3): 30-39.

Truax, B. 1982. "Timbral construction in *Arras* as a stochastic process." *Computer Music Journal* 6(3): 72-77.

Truax, B. 1984. "Models of Interactive Composition with the DMX-1000 Digital Signal Processor." In W. Buxton, ed. *Proceedings of the 1984 International Computer Music Conference*, San Francisco: Computer Music Association, 1985.

Truax, B. 1985. "The PODX system: Interactive compositional software for the DMX-1000." *Computer Music Journal* 9(1): 29-38.

Truax, B. 1986. "Real-Time Granular Synthesis with the DMX-1000." In P. Berg, ed. *Proceedings of the 1986 International Computer Music Conference*, San Francisco: Computer Music Association, 1986.

Wallraff, D. 1979. "The DMX-1000 signal processing computer," *Computer Music Journal* 3(4): 44-49. Reprinted in C. Roads & J. Strawn, eds. 1985. *Foundations of Computer Music.* Cambridge, MA: MIT Press.