

FINAL WRITTEN REVIEW – POLYGRAIN

Joshua Jaworski 309234557 (SID)

Final Written Review for Digital Audio Systems, DESC9115, 2016
Masters in Audio and Acoustics
Faculty of Architecture, Design and Planning, University of Sydney

ABSTRACT

Polygrain is a powerful compositional tool developed to allow time-based manipulation of an input signal. Based on the principles of granular synthesis it breaks an audio signal down into micro-segments or ‘grains’ using them as building blocks in the creation of an output signal. Through an explanation of the Digital Signal Processing (DSP) involved in its implementation and the subjective evaluation of output sounds produced by Polygrain, further development of the plug-in is justified due to its compositional validity.

1. PROBLEM DESCRIPTION

There is an overwhelming focus on textural development in most modern granulation plug-ins. These seem to dictate the concept of texture as being the accumulative product of granulations musical components. Polygrain approaches granulation from a different perspective, instead placing emphasis on the generation of rhythm through parameter manipulation. Curtis Roads, a pioneer of granular synthesis in DSP states,

“In much electronic music, rhythm often emerges as the dominant element in a flux of ever-changing parameter interactions. Indeed, rhythm is the sum total of all parameter interactions. Perceived pitch, duration, amplitude, space and timbre all function as articulators of rhythm” (Roads, 2015, p.192).

Additionally the increasing complexity of sonic output from modern digital granulators has led to their ability to operate as stand-alone individual instruments such as Chris Carlson’s ‘Borderlands’, an iPad app that creates multiple grain clouds (textures) from audio waveforms. Polygrain’s development takes a step back to focus on the almost ‘primitive’ tape splicing methods originating in Musique concrete during the mid 20th century.

2. SPECIFICATION

Built on the concept of quasi-synchronous granular synthesis (QSGS) in which grains are placed in multiple output streams at irregular intervals, Polygrain default outputs a stereo waveform consisting of two separate grain streams placed in the left and right channels respectively. This produces a wide stereo image creating rhythmic and spatial interest as a result of variations in the lengths, amplitudes and amount of grains (density) in each stream. At lower densities (1 to 20 grains per second) a gating effect occurs producing almost metric rhythms whereas high-density settings create quickly fluctuating tones.

This rhythmic effect sonically mirrors that which can be achieved by a noise gate or sequencer however the ability for grains to overlap at various densities (the amount of grains per second) and the sheer number of ‘triggered’ micro-objects when compared to a 16 or even 32 step sequencer, allows for increasingly intricate rhythmic effects to be created by the user.

3. IMPLEMENTATION

At current levels of development Polygrain operates purely in the time-domain. Its processes can be thought of as analogous to the cutting and splicing of tiny segments of analogue tape. Additionally as time-domain processing has a much lower computational complexity and therefore cost, than frequency-domain processing the future development of Polygrain into a real time 'instrument' is both feasible and desirable.

Basic user parameter control is implemented in the matlab code with the use of prompted popup menus as shown in figure 1. Fundamental to Polygrain operation, input values are listed separately for each stream. Additionally grain length and amplitude variations can be set to occur within each stream as well.

User parameter settings:

Grain selection – Grain extraction order from input signal, can be either linear (left to right) or stochastic.

Number of grains – Comparative to a density factor

Grain lengths – Length of grains in milliseconds

Grain length variation- Random variations in grain lengths

Amplitude variation- Random variations in grain amplitudes

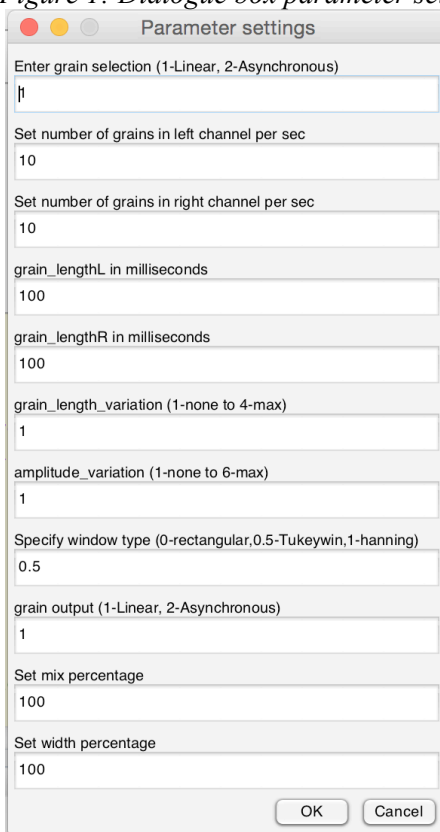
Window type – Select between Rectangular, Tukey or Hann windows

Grain output – Order in which grains are placed in the output signal

Mix – Wet and dry signal mix

Width – Decreases stereo width to mono

Figure 1: Dialogue box parameter settings (Matlab_R2015b)



In addressing concepts of Polygrain's DSP lets always think of $x(n)$ as an input signal and $y(n)$ as a output signal. Granulation of an audio file can be seen as a function of just an extraction and synthesis algorithm, both of which are universal to most, if not all granulators. The overall signal flow of Polygrain is shown below in figure 2.

Figure 2: Polygrain signal flow

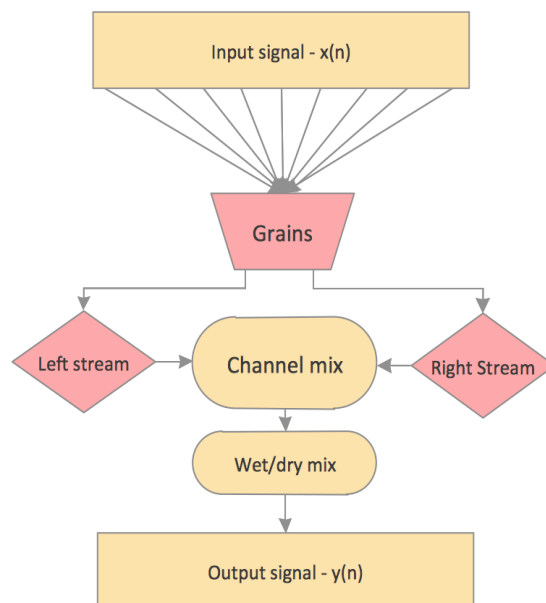


Figure 3: Grain extraction algorithm (Zolzer, Arfib, 2011)

$$G_k(i) = x(i+i_k)w_k(i)$$

The grain extraction algorithm shown in figure 3 utilizing a loop contained in i , where $i= 0, \dots, L_{k-1}$ to extract a set of grains, $G_k(i)$ from an input signal. A window function similar to those used for segmenting audio in spectrum analysis is then applied over each grains time instance of k so that artifacts produced from sudden discontinuities are avoided. The chosen window type is that of a tukey (tapered cosine) window (see appendix A) as it allows user control over the perception of frequency content in each grain. At this stage Polygrain avoids further signal processing such as time stretching or pitch shifting, as retainment of the sonic characteristics of the source material is desirable.

Figure 4: Synthesis algorithm (Zolzer, Arfib, 2011)

$$y(n) = \sum a_k g_k(n-n_k)$$

Shown in figure 4 grains are multiplied by an amplitude coefficient (a_k) and added to an output signal (stream) of predetermined length. In this development version of Polygrain audio output length is finite and set to match that of the input signal due to the limitations of Matlab as an audio processor. Future implementation will involve converting the matlab code to the programming language of C and finally into max/msp, a platform more appropriate for user controlled audio processing.

Grain length and amplitude variations are implemented through the use of randomly generated numbers within an increasing offset from a center value (see Appendix B) with grains being placed in the output signal in either a linear or stochastic order according to desired sonic results. Linear retains harmonic and melodic structure whilst stochastic can create interesting rhythmic amalgamations of the sources harmonic materials. Finally although this effect is built upon the concept of utilizing stereo width a narrowing of the signal to mono is achieved through the creation of a centre channel and subsequent amplitude mix of the stereo and mono signal.

4. EVALUATION

Granulation has long been at the forefront of experimental sound design influencing both the compositions and writings of Xenakis through to Curtis Roads. Accordingly Polygrain lends itself to exploratory sonic breath. Overall the rhythmic effects created are highly dependent on the amount of parameter difference between the left and right channels with extreme value settings producing outputs that could be considered musically unusable in many situations. The only user restriction inherent in Polygrain is that a single grains length cannot be larger than the output file length so realistically granular limitations are that of sonic taste.

Examples of Polygrain's capabilities show its validity in the field of electronic music composition. Parameter settings have been noted for each example (see appendix C) with additional waveform plots showing both the original and granulated files (see appendix D). Rhythmic movement in the stereo field is achieved through differences in density values between streams as shown in **Guitar G1** whilst mixing the dry and wet signals of a **guitar** line with appropriately matched grain lengths and densities in **Guitar G2** offers an sonic output appealing to more conventional ideals.

A simple 4 chord **piano** figure is granulated with high grain densities, length and amplitude variations and stochastic grain outputs producing **Piano G3** whose sonic characteristics can be described as fast moving continuous rhythms. Densities are reduced in **Piano G4** creating a stereo sequenced rhythmic pulse of harmonic material. Not just useful for the complete restructuring of an input source Polygrain can also be used for more subtle elaborations on a sound. **Drum 2** is already a fairly fast paced intricate loop however subtle stereo movement can be added as shown in **Drum 2 G6** by the introduction of just grain length variations and modest density settings. Alternatively a loop can be cut up into sparse and jittery rhythms like those in **drum 1 G5**.

Further product development, pre-performance assessment will include a beats per minute grain quantization allowing for ease of use in metered electronic music. The ability to create additional grain streams placed via simple amplitude modulation in various positions within the stereo field will also be implemented. Both of these will add to the rhythmic development and complexity of the plug-in.

Evaluation of Polygrain's performance for a human user should focus on its use in real life compositional situations and subsequent user feedback. The sonic exploration inherent in granulation lends itself to assessment by user feedback from online communities such as Cycling 74's forums or other user library databases (e.g. Reaktor user library), where as long as the program is stable, prototypes in the form of Graphical User Interfaces (GUI's) can be uploaded for use. Validation of this avenue of user assessment is spoken of by Curtis Roads in *Microsound*,

"Research into sound synthesis is governed by aesthetic goals as much as by scientific curiosity. Some of the most interesting synthesis techniques have resulted from applied practice, rather than from formal theory. Sound design requires taste and skill and at the experimentation stage, musical intuition is the primary guide."(Roads, 2004, p.99).

5. CONCLUSION

Polygrain's current and continuing development shows its validity as an effective tool for musical composition. It affords the user an alternative approach to granulation, focusing on the overarching aspect of rhythm as the sum of all parameter interactions. Continued development will see its user base grow along with its sonic output.

6. REFERENCES

Marshall, David. *BSc Multimedia (CM3106) PDF Lecture Notes and Slides*, retrieved 5th April 2016, <http://www.cs.cf.ac.uk/Dave/Multimedia/PDF/>

Roads, C. (2015). *Composing Electronic Music: A New Aesthetic*. Oxford University Press.

Roads, C. (2004). *Microsound*. MIT press.

MathWorks. (2016). *Tukeywin (Documentation)*. Retrieved June 2nd, 2016, <http://au.mathworks.com/help/signal/ref/tukeywin.html>

Zolzer, U., & Arfib, D. (Eds.). (2011). *DAFX: digital audio effects* (Vol. 1). Wiley.

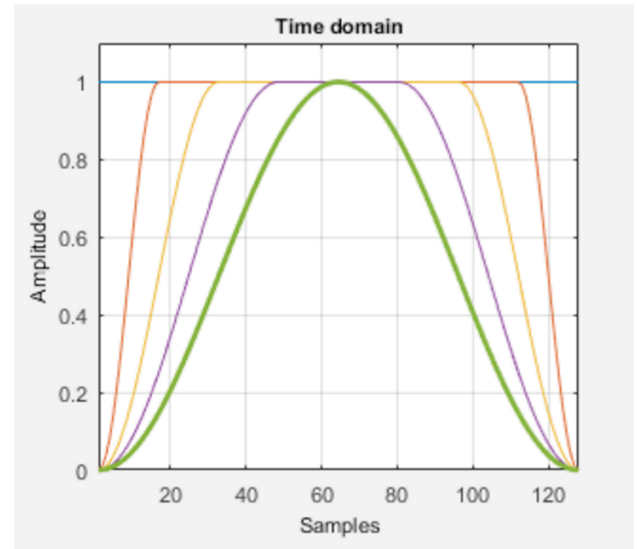
7. APPENDICES

7.1. Appendix A – Tukey window (matlab_R2015b help documentation)

Window shape is determined by user input. User input values from 0 (rectangular), 0.5 (tukey) and 1(hann).

$$w(x) = \begin{cases} \frac{1}{2} \left\{ 1 + \cos\left(\frac{2\pi}{r}[x - r/2]\right) \right\}, & 0 \leq x < \frac{r}{2} \\ 1, & \frac{r}{2} \leq x < 1 - \frac{r}{2} \\ \frac{1}{2} \left\{ 1 + \cos\left(\frac{2\pi}{r}[x - 1 + r/2]\right) \right\}, & 1 - \frac{r}{2} \leq x \leq 1 \end{cases}$$

Where x is vector, in this case an audio grain. The parameter r is the ratio of cosine-tapered section length to the entire window length with $0 \leq r \leq 1$.



7.2. Appendix B – Grain length and amplitude variation

Achieved through randomly generated numbers within an increasing offset from a center value. Deviations from a value of 1 for grain length and a value of 0.5 for amplitude variation.

Grain length:

Grain length = Mean length value

minL = Minimum length

maxL = Maximum length

$$\text{Length} = \text{Grain length} * (\text{minL} + (\text{maxL} - \text{minL}) * \text{rand}(1, \text{Density}))$$

Amplitude variation:

minA = Minimum amplitude

maxA = Maximum length

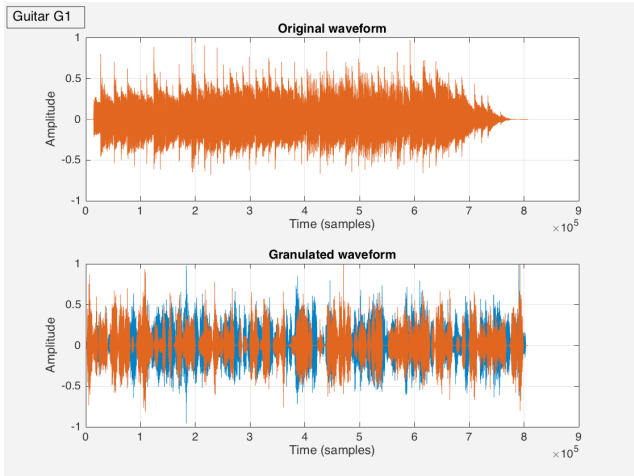
$$\text{Amplitude} = 1 * (\text{minA} + (\text{maxA} - \text{minA}) * \text{rand}(1, \text{Density}))$$

7.3. Appendix C – Example files parameter values

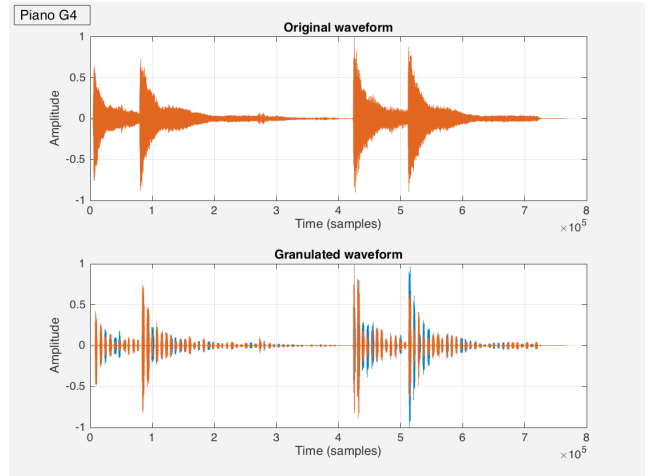
	Guitar G1	Guitar G2	Piano G3	Piano G4	Drum 1 G5	Drum 2 G6
Grain selection	1	1	1	1	1	1
Grains per sec Left	14	5	60	6	7	6
Grains per sec Right	16	7	80	6	8	4
Grain length Left (msec)	200	80	80	100	250	100
Grain length Right (msec)	170	60	75	100	300	100
Grain length variation	3	4	2	4	4	4
Amplitude variation	4	2	3	4	1	1
WinType	0.5	1	0.5	0.5	0.5	0.5
Grain output	1	1	2	1	1	2
Mix	100	70	100	100	100	100
Width	100	100	100	100	100	100

7.4. Appendix D – Input and output waveform plots

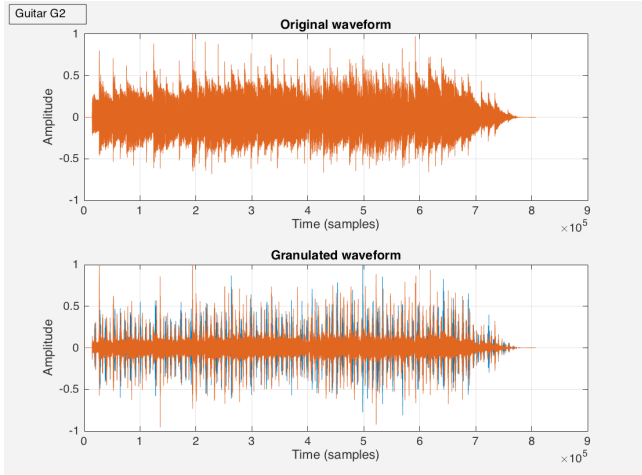
Guitar G1



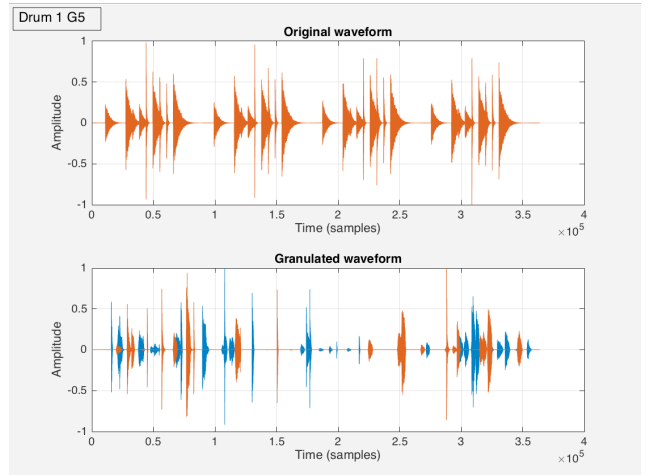
Piano G4



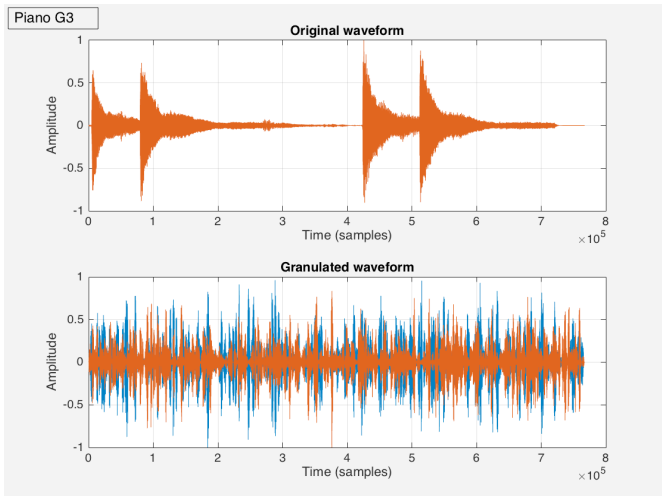
Guitar G2



Drum 1 G5



Piano G3



Drum 2 G6

