Exploring MASTER BOOT RECORD's Music by Designing and Building a 'Synth-Metal' Synthesizer in Max for Live.

SID: 201279673

ABSTRACT

In a typical heavy metal song, there are multiple layers that can be taken over by synthesizers instead. Artists have experimented with such techniques in music production before, but it is still far too uncommon to consider synthesizers a staple in metal. On the other hand, there are genres, such as 'synth-metal' that completely eliminate any non-synthesizer instrument from the track and rely solely on synthesizers for all the voices. In this project, a 'synth-metal' instrument built using Max 8 will be presented as a result of experimentation in order to find the qualities that make a 'synth-metal's lead, riff, and bass synthesized sounds. The design will be explained first, then the implementation using Max 8, which does require pre-requisite knowledge to fully understand. Finally, the findings of covering the 'synth-metal' song 'Config.sys' will be highlighted.

1. BACKGROUND

Heavy metal is a genre of music, mainly characterized by the presence of guitars with heavily distorted tones and bass-y rapid drumming accompanied with screaming or growling vocals [1]; such music is often fast-paced, aggressive and full of energy. These qualities are transferable to another genre known as 'synth-metal', which can be defined as heavy metal music that incorporates synthesizers, often sharing a similar structure to traditional metal, including a bass instrument, riffs, and a lead. The Italian artist MASTER BOOT RECORD (MBR) specializes in the production of such music by utilizing a purely digital approach where all the instruments are synth-made – 'the guitars aren't actual guitars – they're synth' (MASTER BOOT RECORD) [2]. Creating a 'synth-metal' synthesizer encompasses a dual-purpose: To test my personal understanding of synthesizers by recreating synth sounds from MBR's discography, and, to serve as a freely available instrument that can be a valuable experimentational tool for novice producers interested in sound design, more specifically in 'synth-metal' and related genres.

2. INTRODUCTION

The main purpose of building a synthesizer from ground-up is its use as a vessel for personal exploration and application of sound design theory to ultimately replicate MBR songs. The synthesizer would have to be as self-sufficient as possible, with very minimal additional plug-ins, to impose a more controlled environment for authentic experimentation and sound synthesis. A secondary purpose to said synthesizer would be its application as a tool useful to inexperienced electronic music producers, such as heavy metal guitarists who have taken an interest in synth music and would like to incorporate said element into their music. As per such goal, the design of the synthesizer is left rudimentary with a broad range of features pertaining to 'synth-metal'. On a macro level, it remains as complex as typical synthesizers, and on a micro level, it is rather basic. By having a wide range of features relevant to 'synth-metal' but each with minimal options, the synthesizer is well-equipped for sound design to both the experienced and the learners.

The software used to design and build the synthesizer is Max 8, or Max for Live (M4L) in Ableton. The leading reason for that is its ease of use and compatibility with most modern computers. With different objects each encompassing complex functions, the implementation of the instrument is relatively easier than coding it using C++ and JUCE (even with the use of libraries) for example, and making a user-friendly UI for the instrument is straightforward with the Max's presentation mode. Moreover, having Max be integrated into Ableton makes testing the instrument faster, and thus the workflow is more efficient.

Config.sys, a song by MBR, is going to be the judge of the resulting instrument. The song has three main synth-lines, the lead, the riff, and the bass, and by attempting to recreate the song as closely as possible in Ableton, using nothing but this instrument, it can truly be tested on the axes of functionality and viability.

To cover Config.sys, the features implemented will not only include the essential ones found on most synthesizers such as sound generators – in this case oscillators, an ADSR envelope control, a filter, and a frequency modulating LFO (low-frequency oscillator)but also more customized built-in effects such as a compressor, delay, reverb, and wide range of distortion options that can be used to sculpt the desired waveform, with a primary focus on 'waveshaping'.

3. THE DESIGN OF THE INSTRUMENT

3.1 Polyphony

The first thing to consider when making a synthesizer in Max is polyphony because of how the volume envelope and frequency modulating LFO interact with different voices. With 'synth-metal', notes often overlap due to power chords and legato, so polyphony is an essential attribute. The number of voices used is eight rather than sixteen to keep computer processing to a minimum without confining the producer using the instrument, and that should be sufficient for most synth work, unlike a piano score, where simultaneous notes played can exceed eight.

3.2 Sound Generator

Between having oscillators or a wavetable as the sound generator, the oscillators were chosen due to their ease of use and conceptualization. Having a simplified wavetable would require the use of an additional LFO to modulate the different waveforms and demands a little more technical expertise from the user. The oscillators on the other hand can produce 'synth-metal' looking waveforms with a few clicks, and minimal sound design theory. The

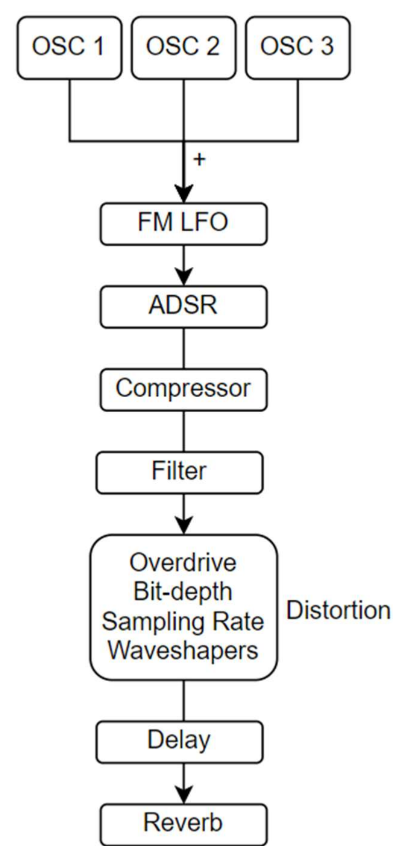

Figure 1: Synthesizer Flowchart

synthesizer has a total of three oscillators, each with a volume knob, octave knob, and a selector between four shapes: sine, triangle, saw, and square. The octave knob is imperative to additive synthesis to create a plethora of different instrument synths using the synthesizer; by detuning certain waveforms an octave or two lower, a pad synth can be derived from the lead synth. Noise is excluded from the 3x OSC because it is rather superfluous for 'synth-metal' and especially in MBR music.

3.3 Pre-processing ADSR, Compressor and Filtering

The volume ADSR envelope, compressor and filtering are all standard with minimal added features. The logic behind having these before distortion is to sculpt the base waveform before introducing effects that may completely disfigure the waveform, and with waveshaping experimentation, results can be unexpected.

3.4 Distortion

The reason distortion is so crucial to include is due to the distinct feature of 'synth-metal' which is heavy distorted riffs. Rather than including one type of distortion, the inclusion of multiple distortion effects that feed into one another can accommodate for more diverse use cases. Distortion starts off with an overdrive knob that can amplify the signal before the more selective distortion is applied. While slight changes in these parameters does not produce the largest effect, approaching the floor of either completely dismantles the signal and results in a chiptune-like sounding synth which can be audible in MBR's Ansi.sys 00:21 for example. The third and largest distortion effect is achieved via waveshaping. There are three waveshapers that feed into one another, each based on a different mathematical formula – the first being an arctan function, the second a cubic function, and the third similar to the second but with an added term to the cubic function. The specific equations and their [-1, 1] graphs are shown below in figures 2, 3 and 4.

$$waveshape\ 1 \rightarrow \frac{2}{\pi}\arctan\left(\frac{(ax)}{1+bx}\right)$$

$$waveshape\ 2 \rightarrow\ x-(ax^3)$$

$$waveshape\ 3 \rightarrow (1-a)x+(ax^3)$$

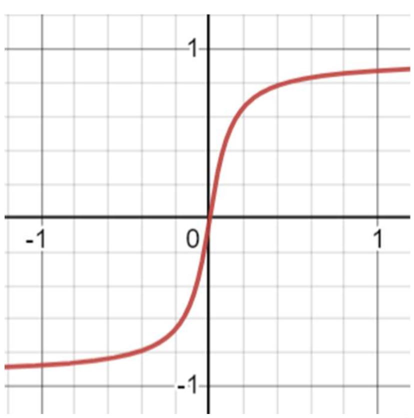

Figure 2: Waveshape 1

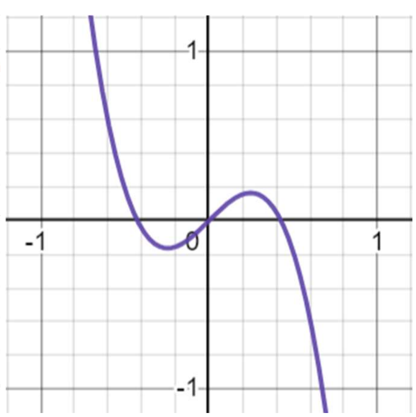

Figure 3: Waveshape 2

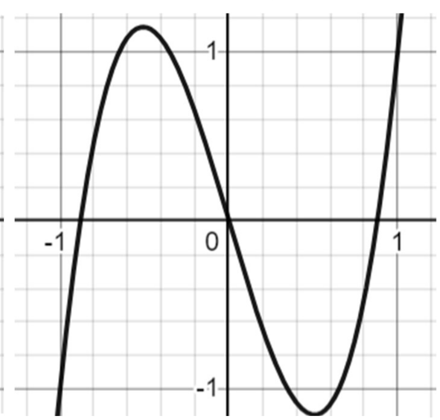

Figure 4 Waveshape 3

In each equation, x is the pre-waveshaping signal and a and b are coefficients of the waveshaper that modify the curve's characteristics, and thus the tone of the distortion, so the knobs control the a and b parameters. Considering the y-axis is the 'gain', clipping potential can be inferred by analyzing the graphs. Waveshapes 1 and 2 are contained within -1 and 1, meanwhile waveshape 3 has the tendency to clip since the signal can quickly get to values beyond 1 and -1 (with a >= 4). This data is pivotal when implementing the waveshaper knob ranges.

The first waveshape applies a girthy distortion to the sound due to its ability to modify a wider range of the signal without clipping. Waveshapes 2 and 3 are similar in shape, but serve different functions. Waveshape 2 does not clip but can attenuate the signal at certain coefficients, making for a soft sounding distortion. Waveshape 3 on the other hand, is the lateral inverse of waveshape 2, but has a larger magnitude factor multiplying it, so it can clip and create the most arbitrary distortion.

Commonly, there is a filter applied at this stage: onto the waveshape itself before it modifies the signal, or onto the signal post-waveshaping. Either method is valid, but each produces a different result, and in this synthesizer, the filter is applied to the signal for the sake of simplicity and ease of contextualization.

3.5 Delay and Reverb

As with most synthesizers, delay and reverb are staples to have. They allow the addition of an extra layer of tonality that can boast the level of stylization and customization without being too difficult to use. The order of these effects does make a difference, and in this design, the reverb is applied after the delay. This is mainly done to avoid a reverberated signal feeding back into the signal line, muddying the sound, so the delay occurs before the reverb.

4. THE IMPLEMENTATION OF THE INSTRUMENT USING MAX

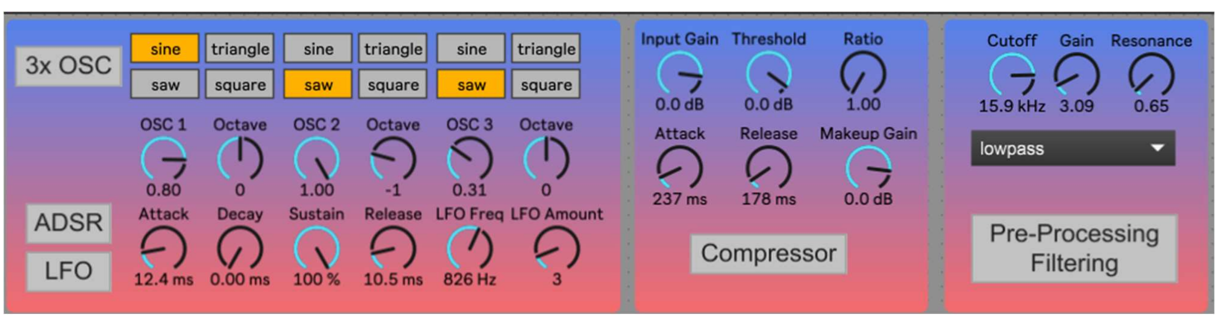

Figure 5: Synthesizer Part 1

4.1 Polyphony

In order to achieve polyphony, the objects used are poly~ and thispoly~, as per the teachings of [3]. Concisely, this method involves building the sound generator within the poly~ object and labelling it as a 'single_voice' with a parameter to set the number of max voices playing simultaneously. By connecting the mute outlet of adsr~ to trigger the thispoly~ object, the voice is 'freed' when the key is let go of. The number of voices of this synthesizer is set to eight.

4.2 3x OSC

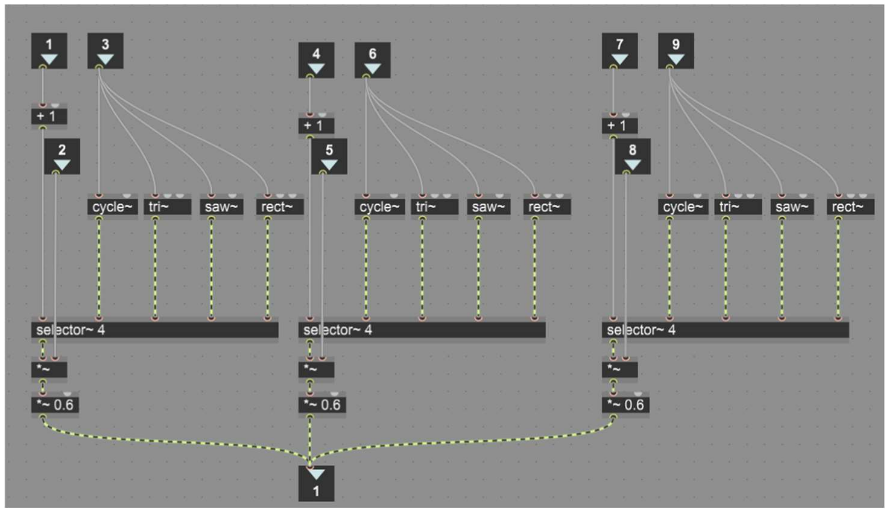

Figure 6: The Three Oscillators Sub-patch

The three-oscillator approach requires multiple wave generators that are summed up before the cumulative signal is sent to the envelope. To have an option for different waveforms with different frequencies, each oscillator must have their own set of four potential waveforms and a selector~ object to select between them, as shown by Figure 6. Before sending the oscillator frequency through inlets 3, 6 and 9, there is a option to add or subtract factors of 12 to the midi information with a knob labelled as 'octave'. Also note the attenuation of the signals by 60% before they are added to avoid unwanted clipping.

4.3 ADSR

ADSR volume control envelope is implemented using the adsr~ object.

4.4 Frequency Modulating LFO

The LFO is implemented using a cycle~ wave generator where its amplitude and frequency are LFO amount and LFO frequency. This cycle~ is added to the frequency of the oscillators.

4.5 Compressor

The compressor is fundamentally an attenuator with an 'if' clause attached to it that triggers it. The implementation of [4] has been adopted for this synthesizer due to its foundational functionality and lack of convoluted mathematics. It is built using gen~ and within gen~, the signal is taken and compared to the 'threshold' and depending on the difference between the signal gain and the 'threshold', it is attenuated by a factor of the 'ratio'. A slide object is added is smooth out the attenuation. The knobs are shown in Figure 5.

4.6 Filtering

Both filtering modules in this synthesizer are built in the same way using the combination of filtercoeff~ to control the parameters of biquad~ which is a basic two-pole filter. The implementation is based on Max's help guide. The parameters are shown in Figure 7.

4.7 Distortion

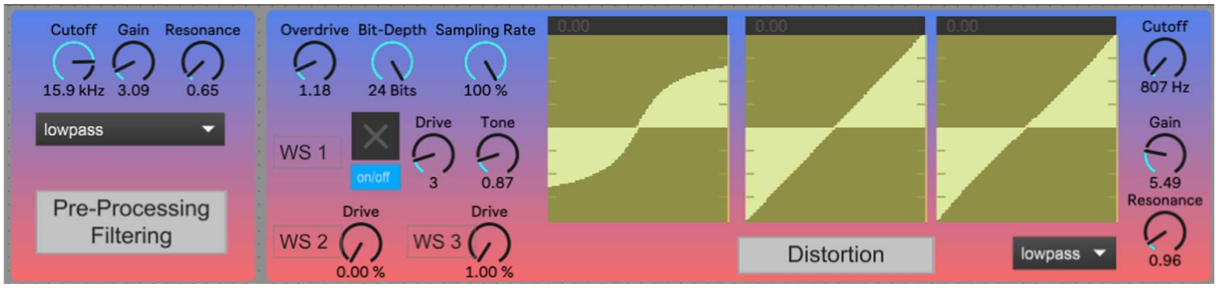

Figure 7: Synthesizer Part 2

The three types of distortion included are Overdrive, Bit-depth/Sampling rate attenuation, and waveshaping. Overdrive is straightforward and is carried out with the use of the overdrive~ object having one parameter to control. The second stage is bit-depth and sampling rate [5] added using degrade~'s two input parameters for bit-depth and sampling rate respectively. The final distortion effect is the most sophisticated method of distortion, which is waveshaping using mathematical equations. Common equations include Chebyshev's polynomials or cubic functions [6] but also arctan functions due to their range being between -1 and 1, which means it never clips. These waveshapers have been implemented using lookup~ and a buffer~. The equations are fed to a peek~ object and fill in the buffer with the designated waveform using uzi, as shown in Figure 8.

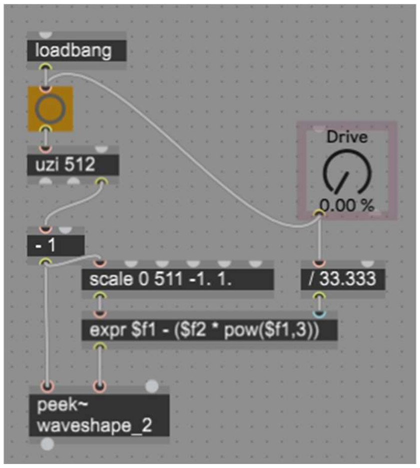

Figure 9: Waveshaper 2 Implementation

Limits of the coefficients	Range of coefficient 'a'	Range of coefficient 'b'
Waveform 1	1 → 100	0 →10
Waveform 2	0 → 3.33	N/A
Waveform 3	0 → 10	N/A

Figure 8: Ranges of the coefficients

These values have been chosen since they effectively cover the most impactful shapes these equations can have. Note that in addition to these knobs controlling the coefficients, the user can draw a custom waveshape on any of the three waveform displays, so that feature can be leveraged by more knowledgeable users.

4.8 Delay and Reverb

By using tapin~ and tapout~ objects, the original signal can be 'delayed' a certain number of milliseconds. The delayed signal can be fed back to the tapin~ object to create a feedback loop. Finally, the delayed signal and the original signal can be joined by a gain mixer.

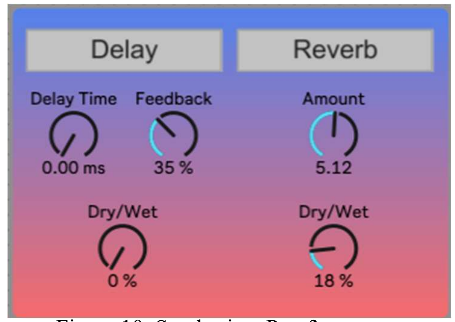

Figure 10: Synthesizer Part 3

Considering reverb does not have a built-in Max object, making it from scratch is mandatory. Once again, gen~ is a vessel powerful enough for the task of applying a set of small delays with different coefficients to simulation diffusion and reflections. The exact values are used from [7] following the guidance of [8]. The core idea of artificial reverberation is a sequence of all-pass filters with small delays connected in series, then are fed to feedback comb filters in parallel, each with a small delay too. Finally, the comb filters are summed up using a mixing matrix. The reverb parameter is basically the coefficient of the number of samples, so it can be labelled as 'tone'. Similarly to delay, the original signal and reverberated signal are mixed.

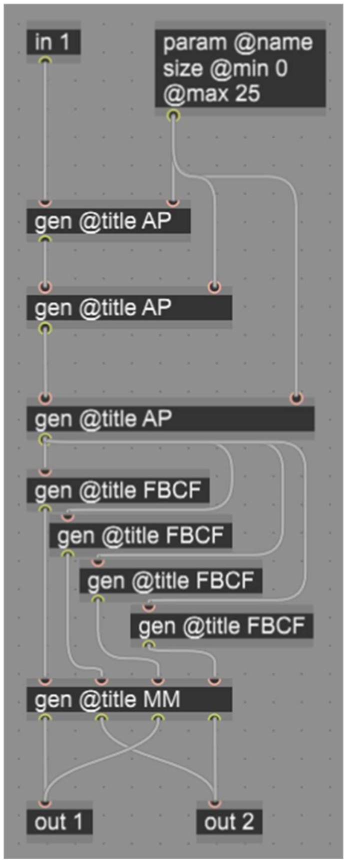

Figure 11: Aritificial Reverb

5. CONCLUSION

The Config.sys cover ultimately proved that the instrument does in fact compose usable synths in a range of different frequency ranges. Of course, this should not be the one and only deciding factor, but it is an indication of it being serviceable and adequate at the job if need be. It is a relatively simple tool capable of functioning as a medium for experimentation, study, and artistic expression.

The next refinement step would be adding a line~ object to each knob to smooth out the change of values and prevent unwanted clips and pops in the sound when adjusting any parameter. Moreover, adding a unison effect would be greatly beneficial to cover and produce a wider array of music, for example the riff synth in Bayarea.bmp. Lastly, adding a presets option and incorporating a few would be greatly valuable to the inexperienced as a starting point to their sound synthesis.

6. REFERENCES

[1] J. Arnett, "Adolescents and heavy metal music: From the mouths of metalheads," Youth & Society, vol. 23, no. 1, pp. 76-98, 1991.

[2] "Master Boot Record," Metal Blade Records, 2023. [Online]. Available: https://www.metalblade.com/us/artists/master-boot-record/. [Accessed: May 16, 2023].

[3] Cycling '74, "Max/MSP Tutorial 4a: The Buffer~ Object and Sampling Theory," YouTube, June 7, 2012. [Online]. Available: https://www.youtube.com/watch?v=y0mcWhrFrEY&ab_channel=Cycling%2774. [Accessed: May 16, 2023].

[4] Tune4Media, "What is Waveshaping? | Understanding Audio Effects," YouTube, June 27, 2020. [Online]. Available: https://www.youtube.com/watch?v=ObBldvarun0&ab_channel=tune4media. [Accessed: May 16, 2023].

[5] Sage Audio, "What is a Reverb Effect? Sage Audio," YouTube, June 12, 2020. [Online]. Available: https://www.youtube.com/watch?v=ABzkEHb3yG8&ab_channel=SageAudio. [Accessed: May 16, 2023].

[6] J. A. Moorer, "About This Document," Music and Computers: A Theoretical and Historical Approach, Columbia University Computer Music Center, 1997, ch. 4, sec. 6. [Online]. Available: http://sites.music.columbia.edu/cmc/MusicAndComputers/chapter4/04_06.php. [Accessed: May 16, 2023].

[7] J. O. Smith, "Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects," Proceedings of the IEEE, vol. 92, no. 4, pp. 664-687, Apr. 2004.

[8] Tune4Media, "What is Distortion? | Understanding Audio Effects," YouTube, June 27, 2020. [Online]. Available: https://www.youtube.com/watch?v=F_FhvxT5mLA&ab_channel=tune4media. [Accessed: May 16, 2023].