Computer Music Languages and Systems
Homework 2

Juan Camilo Albarracín, Michele Murciano, Emanuele Pavese,
Camilo Trujillo, Francesco Zese

1. INTRODUCTION

The distortion effect has been used since the earliest days of the electric guitar. Thanks to its features, it can be applied to create a wide range of sounds; musicians have implemented it for decades as a tool for artistic expression and to create their own signature sound. The distortion effect can be implemented as a self-contained effect (analog or digital) or included as a plug-in in a chain of effects inside a DAW; this second development approach will be the main focus of the document.

The following sections will deal with a complete description of the problem, the approach and methodology adopted by the team for obtaining a functional solution. In addiction, there will be a focus on the plug-in implementation process.

The last sections aim at showing the obtained result along with plug-in functionalities but also a brief conclusion about its performance and possible further developments.

2. PROBLEM DESCRIPTION AND OBJECTIVES

This project presents the opportunity for developing a distortion effect plugin that takes as input a signal produced by a musical instrument and outputs the modified waveform after having processed it. In order to gain this purpose, the software JUCE is to be used.

The following specific objectives are set in order to clarify the aim of the project and establish an appropriate orientation for the upcoming steps:

- Defining the qualitative characteristics of the resulting sound to be obtained by the distortion effect to be designed.
- Develop a plug-in effect that delivers a distorted output according to the outlined parameters.
- Define a processing method that allows the acquisition of a sound signal featuring the desired qualities.
- Implement a graphical interface that allows the user controlling the main parameters of the distortion.

3. APPROACH

An essential step to be taken, prior to the start of the implementation phase of the project, is to gain a comprehensive understanding of the various factors that give identity to a distortion effect. Once the keystone components are well defined, as well as the dynamic of their interaction within the architecture of the distortion, the development can be set in course to achieve an effect that delivers the desired and qualitatively defined objective.

In general terms, the distortion effect can be largely described by a characteristic curve and is defined as nonlinear, time invariant and memoryless. The latter two are explained because the output samples depend only on the input samples and not the time at which they are processed. The consequences of nonlinear processing over sinusoidal signals are to be specially taken into account during the development of the project, given the generation of harmonics at multiple values of the original signal frequency that takes place after the application of the transfer function, not to mention the importance carried by the input of these new frequency components at the time of defining the output's identity for the effect to be constructed. Qualities such as hard or soft clipping and symmetrical or asymmetrical behavior, also determine important details about the resulting output of the distortion. At this point, it becomes clear that the selection of the nonlinear processing function(s) to be featured in the structure of the effect, constitutes a decisive step for the development of the project and would ultimately determine whether or not the resulting distorted sound matches the desired qualities. Similarly, the method to be used in order to add the intended complexity to the input signal is crucial for guaranteeing the success of the implementation.

With the latter in mind, and in order to widen the flexibility of the tool and the achievable sound complexity range of the plugin, while maintaining a safe related computational cost, the decision is made to use the waveshaping technique as the core of the designed processing algorithm. In addition, the initiative of selecting more than one processing function for versatility purposes is acknowledged. Ideally, the outlined solution would allow the tool to perform the transformation of simple signals into harmonically rich sounds through the application of the defined nonlinear function(s) resulting in the modification of the input waveforms shapes, giving the user the possibility of obtaining distorted sounds with a considerably wide complexity range, from a typical soft clipping sigmoid function to a more exotic effect through the manipulation of the variables and additional capabilities made available by the design of the plugin. The possibility of implementing a Bitcrusher effect is also considered to be desirable as the re-quantization process would widen even more the range of possible resulting sounds while experimenting with the plugin

Regarding the user interface and controls, a general benchmark was carried out for determining which are the most widely featured parameters in commercial analog and digital distortion pedal alternatives. Following the described strategy, it was defined that the final plugin should include: a *drive* variable for allowing the user the possibility of modifying a scaling factor for the nonlinear transfer function's input signal, a *tone* control for interaction with the timbral qualities of the resulting sound and lastly, the interface should also offer the user with the opportunity of manipulating the *gain* both before and after the application of the processing function. Additionally, incorporating a *mix* element is considered appropriate as it is a commonly implemented control in the case of digital processor implementations, as the one here described.

4. IMPLEMENTATION

Having established the discussed general features regarding the plugin's architecture, the following implementation considerations were set in motion for achieving the intended objectives.

Broadly speaking, the main goal of this implementation is to create a distortion effect applying the *Waveshaping* process to an input signal resembling a perfect sinusoid. As has been pointed out, this process can be understood as adding harmonic content to the input through the application of a nonlinear transfer function to the original signal with the purpose of obtaining a somewhat *sharper* waveform. An important aspect to be considered at this point is that, to avoid aliasing induced by the generation of the unbounded series of harmonics product of the application of the nonlinear function, the input signal needs to be subjected to an *Oversampling* process before the transfer function can be applied. Through this preparation step, an important decrease in the amplitude of the aliased audible frequency components of the output signal is achieved. Additional filtering stages are also embedded in processing for purposes of highlighting the desired frequency components in concordance with the sound quality expected from the distortions' output.

As this project seeks the development of a comprehensive tool that allows the user access to useful features for creating sounds with different levels of complexity, two processing functions were selected, each one introducing different characteristics to the output signal. The denomination given to these features is *light* and *hard* distortion, based on the quality of the sound that each one of the transfer functions produce.

The first selected processing function has the following form, being d the *drive* user-controlled parameter discussed in the past section:

$$\frac{2}{\pi}\, arctan\, (dx) \qquad (1)$$

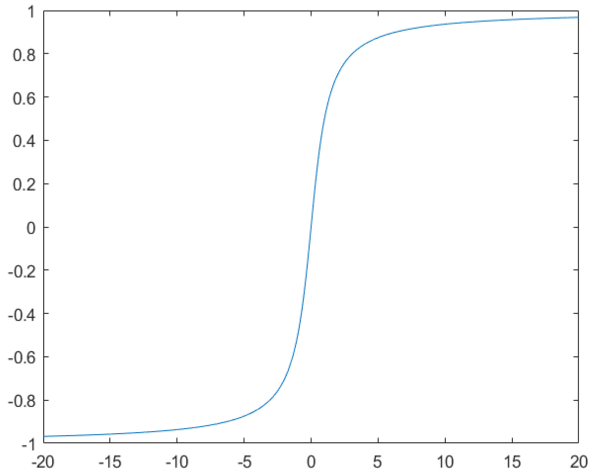

Figure 1. Light function

As shown in the figure, this processing function is odd and nonlinear. When applied to the input signal, the attainment of a new distorted and sharper waveform is expected, nevertheless, the plugin's *light* feature seeks to achieve a sound with a warm quality, for which a smooth approach to the clipping level is desired. This is the function featured for the denominated *light distortion* of the developed plugin.

On its turn, the absolute square root function (equation 2) was implemented for the *heavy distortion* functionality. In contrast with the inverse tangent function, this alternative is symmetric with respect to the vertical axis. When applying this function to the input signal, inversion of the negative components of the signal takes place as well as the increase of zero crossings, which has a substantial effect over the resulting signal's pitch and overall output sound quality.

$$\sqrt{|x|} \qquad (2)$$

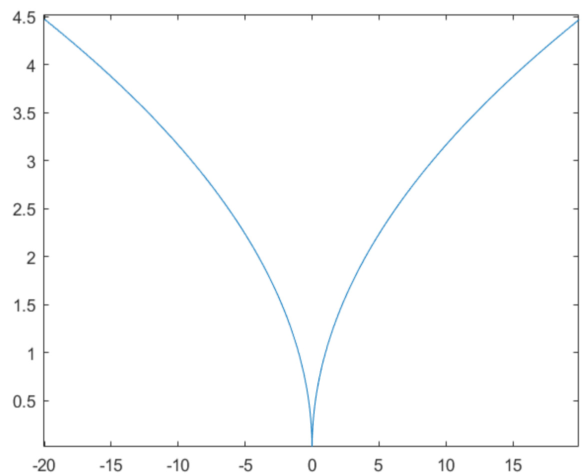

Figure 2. Hard function.

Moreover, on regards the bitcrusher effect, which is a low-resolution digital distortion effect, it was decided to use it in order to emulate the sound of early digital audio devices. The implementation of this kind of distortion includes both the bit reduction and the sample rate reduction. As regards the bit reduction, it is the process of reducing the number of values through which the amplitude of each sample will be re-quantized. The sample rate reduction concerns the increase of the reduction-factor for the sampling rate by quantizing in time resolution; the algorithm used to implement this rate reduction works like the sample&hold procedure: it takes the i-th sample and holds it for however many steps it needs to.

Signal Processing

The diagram in Figure 3 below depicts the general pipeline followed by the signal in JUCE process block. Specifically, this operation is done as a part of each block's iteration, meaning that every sample in the current buffer goes through the process described.

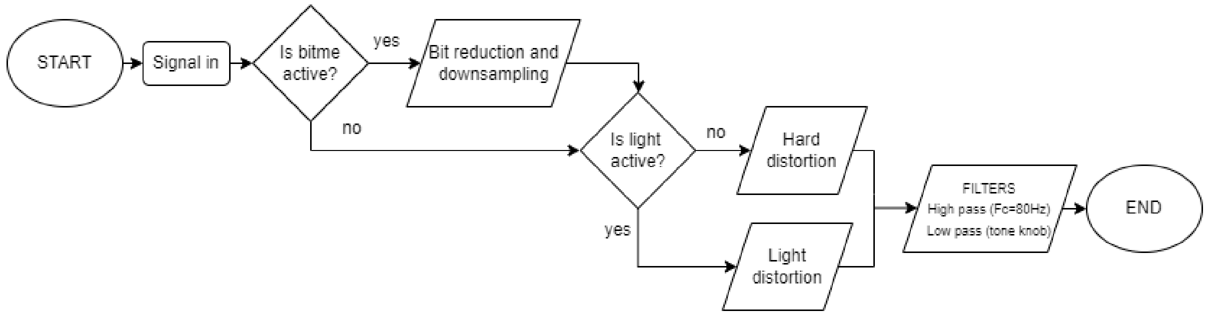

Figure 3. Signal processing diagram

5. RESULTS

Graphical User Interface

Taking into account the graphical approach described above and giving continuity to the *retro* video games theme (concept elaborated on the previous assignment as well) the graphic interface shown in Figure 4 was implemented. The GUI features an audio visualizer and several buttons and knobs described as follows:

- Pre (preamp):
  Input gain level of the signal, it also works as "parallel blend" since it also adds the original (clean) signal to the output. Useful for drums and guitar, or in context where the parallel signal processing is needed.
- Drive:
  Parameter that modifies the linear behavior of the distortion scaling the transfer function, higher values bring more distorted sounds since the linear region is reduced.
- Mix:
  Balance the blend between the clean signal and the distorted one in the output taking values between 0 and 1, using the following equation:

$$blend = input * (1.0 - mix) + distSignal * mix$$

  Which allows to obtain just the clean signal when the knob is in the minimum position and just the distorted one when it is in its maximum. The *Mix* bitcrusher knob operates under the same equation.
- Tone:
  This frequency based parameter modifies the cutoff frequency of a low-pass filter, this cutoff frequency goes from 4kHz to 6kHz. After the tests performed, the default value was set to 4550Hz.
- vol+/-
  Controls the output gain of the signal, its step is set to 1dB.

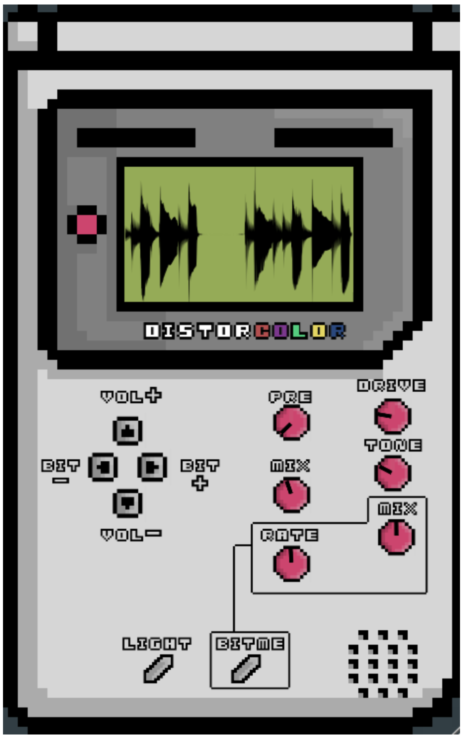

Figure 4. DistorColor GUI.

An additional knob was implemented to control the bit crusher effect when the button *bitme* is active. The rate modifies the sample rate frequency subtracting the remainder after the division of the current sample by the rate value. Regarding the bit reduction, the buttons *bit +/-* reduce the bit rate of the signal to no less than 8 bits and increment it up to 32 bits. These two parameters combined give a Lo-fi effect by downsampling and re-quantizing the signal. The *bitme* unit can be also used as a noise gate.

Plugin output

In this section an attempt to characterize the plugin is presented. For this means, a series of measures were done using the software REAPER®. Inside the DAW, a sinusoidal signal of 440 Hz was generated and a spectral analyser was placed in the plugin's output. All the measures were done keeping an output level of -6dBFs.

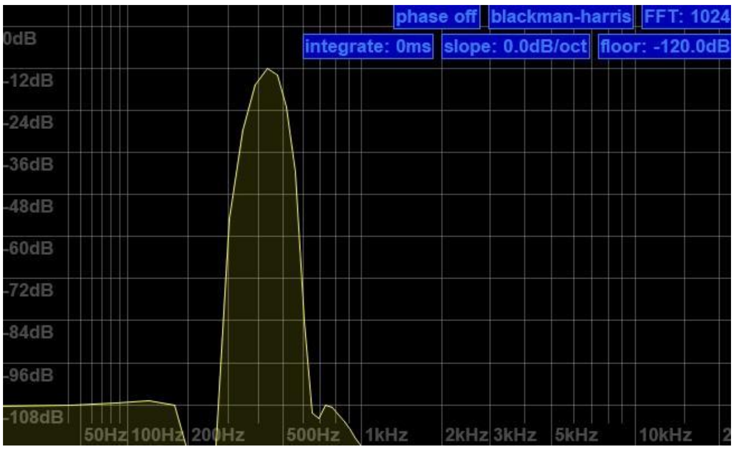

Figure 5. Input signal

Since this distortion is based on a full-wave rectifier, it is noticeable the presence of the octave harmonic and the suppression of the fundamental frequency. Additionally, the sound is enriched with even harmonics that decay thanks to the high-pass filter placed in the output.

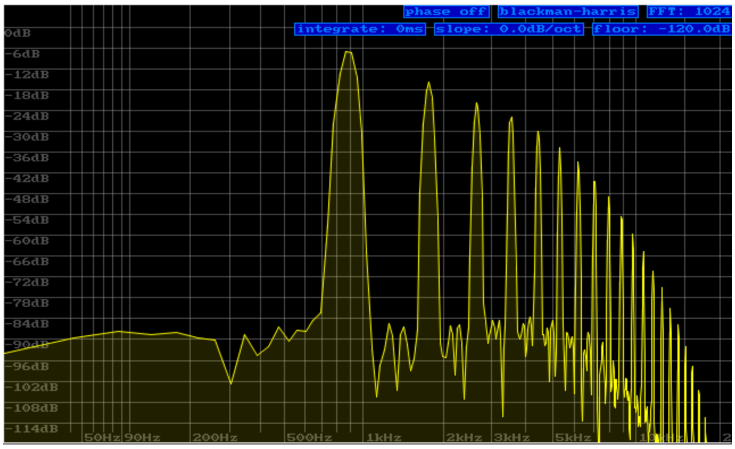

Figure 6. Heavy dist

In the light distortion case, taking into account it is based on a soft-clip distortion and an odd function a collection of odd harmonics is added to the signal, as can be seen in figure 7. In contrast with the heavy distortion, this one does not suppress the fundamental frequency and from the graph it is possible to take notice of how the 440 Hz sinusoid almost remains equal to the original one.

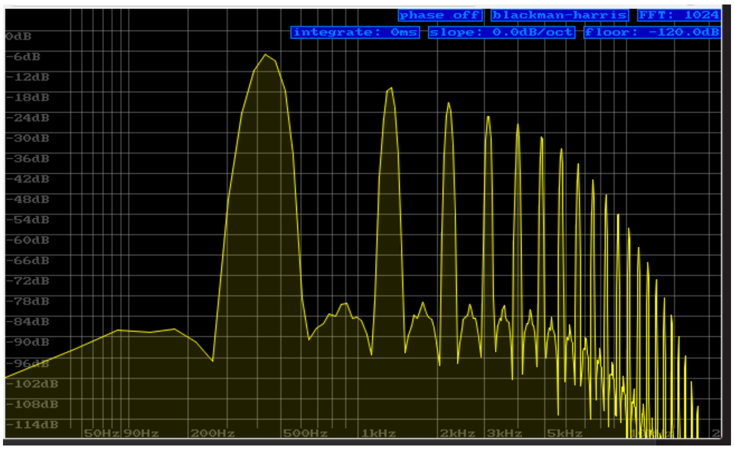

Figure 7. Light dist (softclip)

The bit crusher effect was implemented attempting to give a ‘lo-fi’ effect to the sound, and as can be seen in the figure 9, reducing the bitrate is directly related to decrease the dynamic range of the signal and has as consequence the increasing of the noise floor, directly affecting the signal-to-noise ratio.

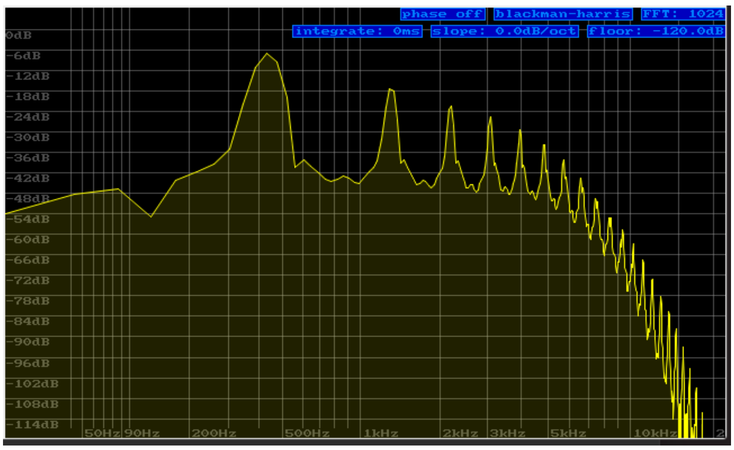

Figure 8. Light + bitcrusher

6. CONCLUSIONS

- Adding different kinds of distortions to the *DistorColor* brings to the musician and audio producers a functional tool that allows obtaining a wide variety of sounds. This was achieved thanks to the definition of qualitative characteristics done during the approach and implementation stages and through the waveshaping method, which also delivers a distorted signal with overall minimum computational cost.
- Using the bit crusher along with other kinds of distortions not only enriches the signal with harmonic distortion, but also increases the noise floor that gives a 'lo-fi' effect.
- Through the graphical user interface, it was achieved to control the main parameters that mold the output. The implementation of the audio visualizer constitutes an aid that gives to the user visual feedback of the processed signal.
- While it is true that the current development of digital alternatives has achieved a remarkable resemblance with the sound of classical analog devices, it can be argued that a much more promising endeavor, from an artistic point of view, would be investing the capabilities of digital tools for coming up with innovative concepts; rather than focusing on the emulation of other technologies, although the motivation for doing so is valid and clear from logistic and financial angles.

7. FURTHER DEVELOPMENTS

Although the objectives of the project were successfully achieved given the initial outlined scope, this section intends to propose possible next steps for the development of the *DistorColor* plugin for making it a more robust and useful tool.

- Adding presets to the plugin would contribute to a more friendly user experience, it could save time and even exemplify in a clearer manner the plugin's capabilities as well as the interaction of its functions.
- Integrating effects such as Reverb and/or Delay could play an important part in making the *DistorColor* a more flexible tool to be used creatively by performers.
- Implement the possibility of controlling the parameters through a MIDI controller.
- Broadening the plugin's compatibility spectrum to other music production software such as FL Studio.