POLITECNICO
MILANO 1863

CMLS - Homework 2

Wah-Gliù

Wah-Wah and Auto-Wah effect with distortion mixing

Di Palma Riccardo	(Person code: 10602207, ID: 988804)
Gargiulo Antonino Manuele	(Person code: 10829418, ID: 990594)
Morena Edoardo	(Person code: 10865449, ID: 996003)
Orsatti Alessandro	(Person code: 10680665, ID: 994757)
Perego Niccolò	(Person code: 10628782, ID: 992023)

Computer Music - Languages and Systems

Homework Assignment

POLITECNICO
MILANO 1863

Contents

1 Introduction

Our project focuses on the implementation of a VST plugin with the use of the `Juce` framework. We decided to start from the basic implementation of a *Wah-Wah* effect, enriching it in a unique way and forging *Wah-Gliù*. The result is a plugin that allows the user to select between 2 different modes:

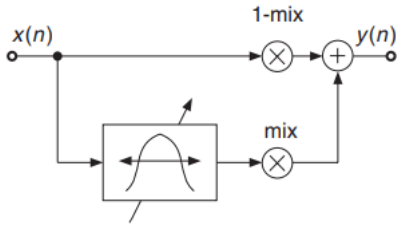
- *Wah-Wah*, where the musician operates directly the effect.
- *Auto-Wah*, where the plugin itself manages the effect, processing the data in input.

Both variants are also fitted with a parametric distortion to obtain a cool mixing between the dry signal and the effected one.

2 Background

2.1 The Wah-Wah effect

The *Wah-Wah* is an effect that alters the tone and frequency of an input signal. It is usually controlled with a pedal, that sweeps the peak response of a filter. In fact, the foot pedal allows the player to move the frequency of the resonant peak up and down, which delivers that signature crying "wah-wah" sound [1].



Schematic for a wah-wah effect

We can see that the input signal $x(n)$ is delivered to 2 different lines: the first signal remains unchanged (`dry` signal), while in the second one it is filtered (`wet` signal). Both signals are then mixed in the output stage.

2.2 Auto-Wah

Auto-wah effects, more properly called Envelope-Controlled Filters (ECF), shape the tone of an instrument (the filtered part) in response to how loud the

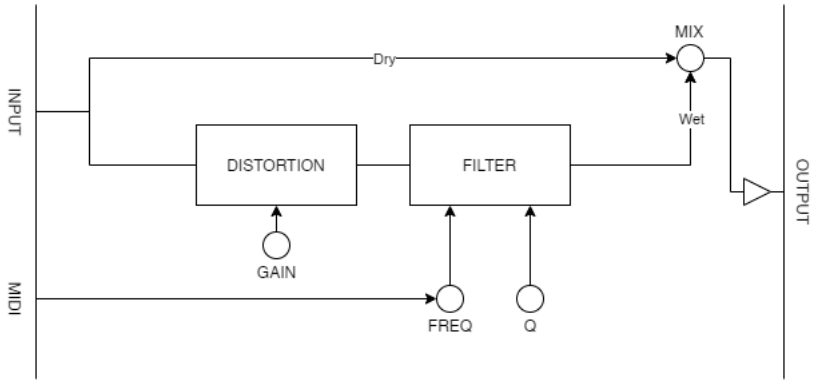
input (the envelope part) is. They are often called auto-wahs or auto-filters to distinguish them from conventional foot-operated pedals [2].

Although they can certainly mimic some of what a wah-wah does, they are substantially more flexible in many ways, and have a broader palette of tone shaping capabilities than a wah-wah. By using a sub-circuit called an *envelope detector*, the size and shape of the input signal get translated into an average change in voltage over time. This resulting signal is then used to control the audio part of the circuit. Under normal circumstances, as the sound gets louder, the filter's center frequency goes higher. As the sound gets softer the center frequency goes lower, often so low that the entire signal is cut out.

3 Implementation

3.1 Wah-Wah

The wah-wah effect is produced mostly by foot-controlled signal processors containing a bandpass filter with variable center frequency and a small bandwidth. Moving the pedal back and forth changes the bandpass center frequency. The wah-wah effect is then mixed with the direct signal.



Schematic for the wah-wah effect

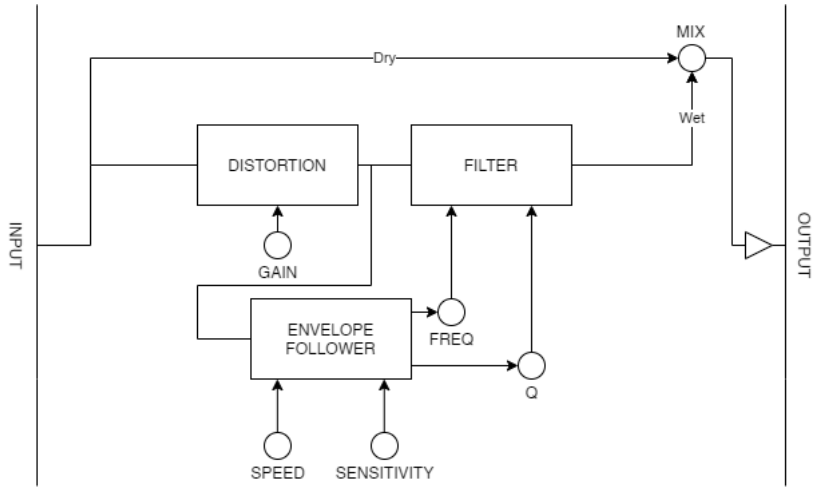
Our wah-wah mode recreates the behaviour of a real wah-wah pedal. The Q factor and the central frequency of the peaking filter are controlled directly by the user through the GUI, along with additional parameters for distorsion (see section 2.3) and volume control. The user is advised to perform the sweep of the filter's central frequency using any MIDI controller by mapping whichever physical control (e.g., expression pedal, slider, knob) to the frequency GUI dial in his preferred DAW.

The core of the audio processing is performed in the `processBlock` function of the `PluginProcessor` script. We use 2 filters, one for the right channel and one for the left channel. The `filterRight` and `filterLeft` objects are an implementation of the class `juce::dsp::IIR::Filter`. The filters coefficients are updated using the method `makePeakFilter` of the class `juce::dsp::IIR::Coefficients`, which creates a peaking filter parametrized by central frequency, resonance (Q-factor), and gain. In our case, the arguments passed to this method are the central frequency and the resonance value selected by the user, along with a fixed gain factor of 7. The sweep of the filter is limited to the range of $[400, 8000]$ Hz.

Finally the samples from the left and right channels are processed, and the resulting signal is mixed with the clean signal. The mixing process is described in the section 2.3.

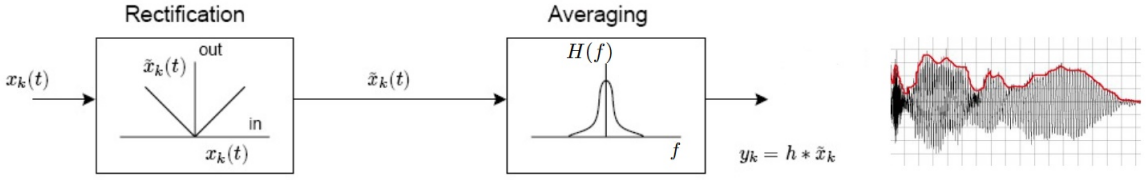
3.2 Auto-Wah

The Auto-Wah feature is implemented through an envelope follower. The main idea is to let the center frequency of a peaking filter be guided by the amplitude evolution in time of the signal. In our implementation, we want the distorted signal to be the one driving the filter, as shown in the following schematic:



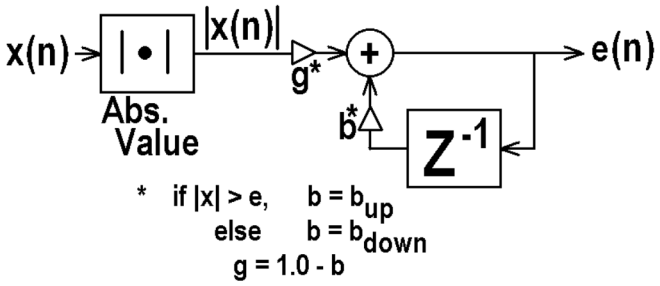
Schematic for the auto-wah effect

A conventional way to compute the envelope of a signal is to first rectify it, by means of the absolute value, and then low-pass filter it, in order to have as output a smooth amplitude trajectory that can easily be followed by the center frequency.



Envelope Follower in the analog domain

For the actual real-time implementation, in the `processBlock` function, we decided to go for a sample-by-sample approach: after rectification, each sample goes through a one pole low-pass signal dependent filter, as shown in the following image:



Real-Time Envelope Follower

In our case, we implemented a unitary delay in order to have access to the previous sample while we are processing the current one. This is accomplished thanks to an array.

Due to the filter signal dependency we can use two different time constants in order to let the output trajectory follow the attack and the release with the corresponding proper speed. We opted for two exponential time constants in order to let the filter open in a non linear fashion.

In our implementation also the bandwidth of the band pass filter is envelope-controlled. In this way, we compute at runtime the quality factor Q, making it dynamically more suitable for a low pass or a high pass filtering depending on the situation; in other words, the filter bandwidth remains constant on a logarithmic scale.

After computing the center frequency and the bandwidth we create the peaking filters (one for the right and one for the left channel) as we did in the Wah-Wah

case. Once the processing of the samples from the two channels is concluded, the wet signal is mixed with the dry one.

3.3 Distortion and Mixing

The distortion is applied to the `wet` signal before filtering. We implemented it with an **arctan** function, which codomain is $(\frac{-\pi}{2}; \frac{\pi}{2})$. A simple normalization of a factor of $\frac{2}{\pi}$ grants us a waveshape which can go from slightly distorted to heavily clipped in the range $(-1; 1)$.

We chose the **arctan** function instead of other waveshaping options, such as **sign** (pure hard clipping square wave) and **tanh** (smoother in the behaviour), just for a matter of personal taste and because it is the more balanced in reproducing a familiar distorted sound.

The distortion gain, which affects the amount of distortion introduced in the effected signal before its filtering, is retrieved from the GUI, along with the mixing coefficient `mix`, used to perform a classic dry/wet mixing operation, deriving the output y as:

$$y = x_{\text{dry}} * \texttt{mix} + x_{\text{wet}} * (1 - \texttt{mix})$$

.

4 GUI and UI

We realized the background image for our plugin in `Blender` to obtain a more realistic 3D base.

The user has a set of possible parameters to control via the Graphical User Interface of *Wah-Gliù*.

- *Q-factor*: acts on the quality factor of the peaking filter applied to the wet signal in wah-wah mode.
- *Speed*: regulates how quick the response of the envelope follower is in auto-wah mode.
- *Sensitivity*: modifies the filter *sweep*, i.e. how much the center frequency of the peaking filter follows the envelope of the signal in auto-wah mode.
- *Distortion*: distortion gain applied to the effected signal before filtering.

- *Mix*: parameter that regulates the ratio between the dry and the wet signal at the output stage.
- *Volume*: master volume of the effect.
- *Mode selection*: allows the user to switch between wah-wah and auto-wah functionalities.
- *Bypass*: grants possibility to bypass the effect, hearing the input as-is.

To offer a unique experience, we opted for the use of buttons and custom knobs, built exploiting the `Slider` class of `Juce` in a rotary fashion, and the possibility to create ad-hoc designs for GUI elements combining the use of the software `JKnobMan`, to produce a splash of the dial with updating angular steps, and an implementation of a custom `LookAndFeel_V4` class, `SliderLookAndFeel`.

The interface is updated depending on which mode of the plugin is selected, leaving untouched the shared properties between the wah-wah and the auto-wah effects. In particular, the controls for the volume, the distortion gain and the signal mixing are always present, along with the bypass and the mode selection buttons. Viceversa, the speed and sensitivity knob are only displayed in auto-wah while the Q factor and center frequency dials are shown in wah-wah mode.

It's important to remark the suggestion of using *Wah-Gliù* in a DAW, mapping all the GUI knobs to MIDI controllers. This is particularly useful for the center frequency parameter in wah-wah mode. The kind of effect we developed is usually found in guitar pedals and highly used in live performances, thus requiring parameters to be constantly changed while playing. DAWs usually block direct MIDI control access to VST effects, offering instead the possibility to assign internal parameters in all plugins to external MIDI controllers.

References

[1] U. Zölzer, editor. DAFX Digital Audio Effects. John Wiley Sons, 2002.

[2] The Technology of Auto-Wahs / Envelope-Controlled Filters,
URL: `http://www.geofex.com/article_folders/ecftech/ecftech.htm`