

“Timbral Inflector”

Caleb Adams

Code:

<https://github.com/CalebAdamsCMUApplication/Caleb-Adams-CMU-Application-Code-Supplement>

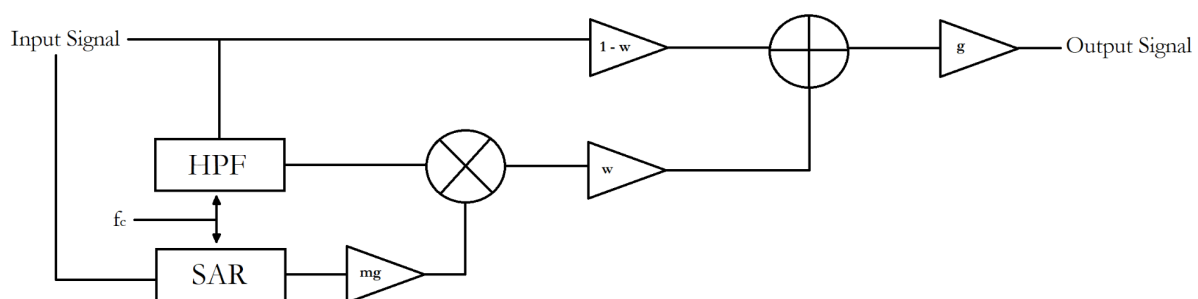
Motivation:

Timbral Inflector is a high frequency exciter that uses ring modulation to saturate the upper frequencies of an input signal. The original intentions for the device were to create a grainy, distorted, bordering-on-oversaturated sound that could be made more subtle through easier and simpler linear filtering/processing and basic reverberation. In my own work as a composer and sound designer, I find it more precise to “overdevelop” a sound before sculpting it down into its final form through subtractive techniques. I also enjoy devices that possess unique nonlinear behavior and devices that behave very differently given different inputs. Such devices remind me of the quirks of writing for acoustic instruments, which change greatly in timbre through their effective playing range. Considering all of this and my obsession with ring modulation, I landed on the current concept for *Timbral Inflector*.

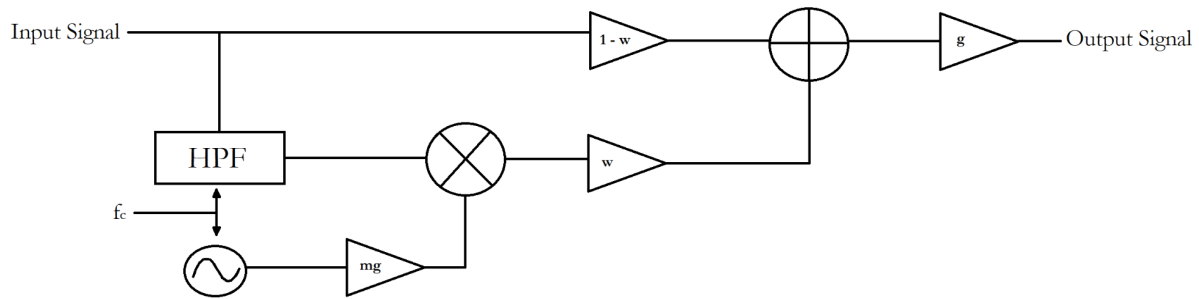
Technical Description:

Timbral Inflector itself has two operating modes: self modulation mode and normal mode. In self modulation mode, a Smith Angell Resonator is employed to obtain a modulating signal from the input signal. In normal mode, the modulating signal is a simple sinusoid; the normal mode was added to compensate for signals that are too simple to create an interesting modulator from the resonator. Both modes employ a high pass Butterworth filter to obtain the excited component of the signal, and both filters employ a biquad filter topology. The digital signal processing (DSP) diagrams for both modes are shown below.

Timbral Inflector - Self Modulating Mode Diagram



Timbral Inflector - Normal Mode Diagram



The device uses several variables changeable by the UI. In the diagram, these variables are:

- **mg** - the modulator gain, a value between 0 and 5
- **w** - the wet/dry mix of the signal, a value between 0 and 1
- **g** - the gain of the signal, a value between 0 and 5
- **fc** - the cutoff frequency of all filters, a value between 0 and 2/5 the sampling rate

Implementation:

Timbral Inflector is implemented using JUCE, an open source C++ framework for audio development. The plugin itself is compiled as a VST3, but by using JUCE, the target of compilation can easily be changed to other common audio plugin formats. In JUCE, the anatomy of a plugin is divided between the processor .h and .cpp files, where the DSP occurs, and the editor .h and .cpp files, where the UI is located. The UI consists of four dials, one for each of the four variables (**mg**, **w**, **g**, and **fc**) and a toggle button to change between the two modes of the device (self modulation and normal). The UI is shown below:

