Digital Audio Systems DESC9115: Assignment 2
Final Review Vibrato Excitation Plugin

Synthetic Vibrato Plugin

Vibrato – Separate the input's fundamental excitation from the filters, applying vibrato to the fundamental sound then is adding its original signal filters to the output.

Problem Description

The given problem is how to apply a natural sounding digital vibrato into a recorded instrument digital audio sample. For this purpose we can divide the problem into two; first, to apply a vibrato into a recorded instrument sample and second to make the vibrato to sound like a real sounding like cord instrument vibrato, emulating the "natural" variations of the human player´s touch.

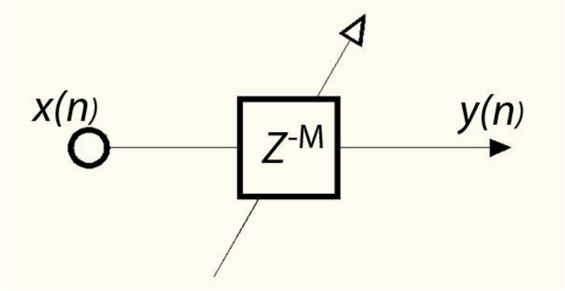

Fig1. Basic Vibrato System

The approach to solve this problem is done through matlab coding, which creates a vibrato system to apply into an audio sample input.

[y fs f]=vibrato(Modfreq,Width)

. All pass Interpolation Vibrato code

However the problem solution is meant to address the main target of emulating a "real" or human performed vibrato, therefore the original all pass interpolation formula for the vibrato has to be modified to be able to produce variations in the signal.

For this purpose a modified version of an all pass interpolation Vibrato function is suggested, it has added a fractal noise creation parameter to modulate the signal. Allowing the user to input more parameters into the vibrato.

Proposed Syntax:

synWave=delay_mod(Input,FS,Width, lpc_filter_order);

Specification

The previously described problem is addressed by a source filter approach; the original digital audio signal will go through an audio analysis method to separate the input (recorded instrumental sound) into two components;

- the *source* or excitation and
- the filter or resonant structure

Once this is done a vibrato effect shall be applied to the separated *source* giving freedom of the user to apply a modulation of the parameters of the vibrato. The vibrato´s parameter modulation could be done by the user by means of virtual knobs that increase or decrease three different parameters;

- The length of the vibrato´s time delay in milliseconds.
- The number of filters that will be split or separated from the fundamental sound(excitation)
- The amount of fractal noise created
- A low pass filter knob, so the user can limit and modify the frequency content of the modulator.

Once the vibrato envelope is created and added to the sample´s fundamental excitation, the resonant structure must be added again into the mixture to accomplish the desired output, a natural and realistic sounding vibrato effect in a recorded instrument digital audio sample.

The resulting implementation of the filters to the input excitation is meant to be done through a real time software plug- in able to run through programs such as Protools and Ableton.

The plug-in should also have a couple of windows which show in real time the graphics of the input signal and vibrato performance as described below, these could also deliver the user another way of modifying the output parameters:

- A couple of windows that show the system´s input as well as the output (these windows are just visual aids not meant to serve as parameter modifiers).
- A window showing the filter or envelope of the input signal that may allow the user to modify the envelope
- A window showing the change in the amplitude and frequency of the vibrato effect that can allow modification of the parameters by clicking and moving the sine wave.

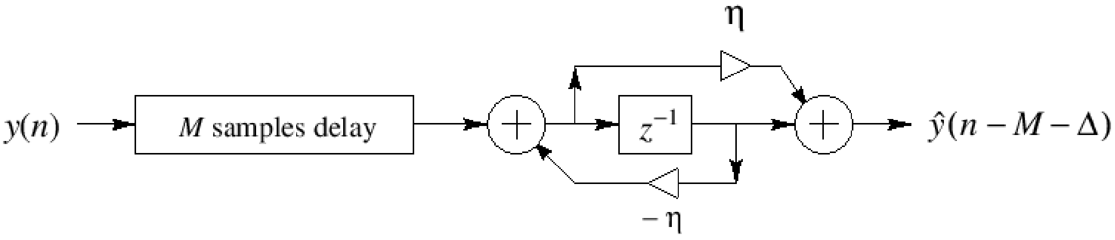

Fig2.. All pass interpolation delay line

Implementation

The way the inside of this plug –in works is generally expressed by the *matlab* syntax:

synWave=delay_mod(Input,FS,Width, lpc_filter_order);

For start, the input is separated into its fundamental noise and a series of filters whose number is determined by the user, these parameters ranges vary according to the instrument used, for example the best results for a violin are in the range of 3 to 6. Then fractal noise is created for the modulation and the low-pass filter applied to limit the frequency content of the modulator all of this just to the fundamental sound of the audio input.

`synWave=delay_mod(Input,FS,Width,       lpc_filter_order);` produces a signal with a vibrato added only to its fundamental frequency as its output. The whole arguments in the function must be determined by the user.

The formula inputs are:

- `Width` which determines width of the delay applied to the vibrato.
- `lpc_filter_order` determines the number of filter coefficients into which the input signal will be split when divided into fundamental sound and filter(s). Low values for example between 3- 6 in the violin case, perform better since more filters tend to be more noticeable at the output signal.

Example

Specification

Create a human like vibrato effect to a violin input.

```
synWave=delay_mod(Input,FS,.0002, 3);
```

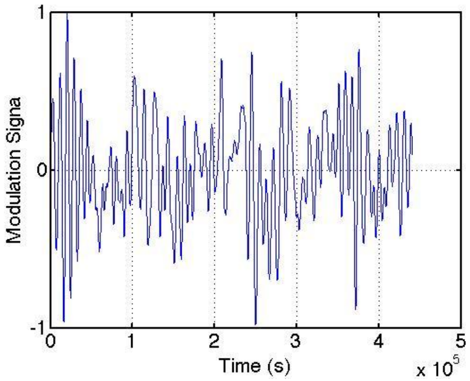

Fig3. Vibrato added modulation signal graph

Diagnostic

The result is indeed a more human like vibrato, however if the parameters are not correctly modulated you can get notice the presence of the filter at the end as the sound is fading. If the parameters are kept under the values mentioned the output is quite realistic.

Performance Assessment Criteria

The output response of the system should be assessed by a comparison method in three types of samples; one instrument played with real human made vibrato and two recorded samples with a synthetic vibrato effect applied on:

- A series of samples in a variety of instrument recordings, with real human played vibrato
- A series of samples in the same variety of instrument recordings, with a synthetic vibrato effect applied
- A series of samples in the same variety of instrument recordings, with this prototype plugin synthetic vibrato effect applied.

With the three previous series of samples ready, a listening test should be done into a group of at least 100 subjects. Who must listen to the three different series of instrument with vibrato samples. Ordered to listen each instrument with the three different vibrato effects (one natural and two synthetic) consecutively before switching to a different instrument. The subjects are asked to mark with a number each sample they hear within 1, 2 and 3, according to how realistic does the vibrato effect sound. Being 1 the most realistic or true vibrato sound, two the not so realistic effect and three the least realistic sounding sample.

So for example the test would start with a human played violin vibrato, continued with a vibrato effect and followed with a violin with this plug-in synthetic vibrato applied on.

The aim for this test results to succeed would be that the samples of this synthetic vibrato plug-in are ranged mostly into the 1 or 2 category for all instruments for the 80 % of the test results thrown by the subjects who listened the samples and from that 80% acceptance as 1 or 2, around 50% (half of those) should have been classified as 1 meaning the test subjects could hardly differentiate the samples processed through the synthetic vibrato plug-in, meaning its performance can be compared to that of a real vibrato applied by human into an instrument with its natural warmth and variation.

References

[2] Julius O. Smith III, 2010. All pass interpolation:
https://ccrma.stanford.edu/~jos/pasp/First_Order_Allpass_Interpolation.html
Accessed 7/05/2012
Zolzer Udo, “DAFX: Digital Audio Effects John Wiley & Sons, Ltd., vol. 110, no. 3, pp. 1598–1608, Mar. 2001.