**DESC9115 WRITTEN ASSIGNMENT 2: FINAL REVIEW**

*Sam Johnson #420081870*

Digital Audio Systems, DESC9115, Semester 1 2012
Graduate Program in Audio and Acoustics
Faculty of Architecture, Design and Planning, The University of Sydney

## ABSTRACT

In this review I will explore the importance of a digital delay and the effects it can create in both the live and studio realms. I will look into how it this unit's DSP is created and what it looks like "under the hood".

## 1. Problem Description

The use of delay as an audio and musical effect that has been used since the coming of tape recordings. It involves the signal being taken and "stored" then played back at a later time, either once or multiple times. In modern western music, it is hard to find a song that does not use some form of delay in their mix which is why it is important to create a user friendly, great sounding, accurate delay. A real delay is caused when a sound wave reflects off a surface and because of the extra time taken to get to the receiver (the ear) it has a lagged effect. Their use in the musical realm has been taken to a new level with the use of parameter settings and modulation types, the average Joe can easily create his or her own effect to suit their specific needs. [2]

## 2. Specification

The digital delay multi FX unit that I have designed will contain various parameters including delay time in milliseconds, gain, and a modulation switcher. The effects that can be created from this are echo, flange, chorus, slap back, resonator and vibrato.

The digital delay effects are created by the use of comb filtering. Comb filtering adds a delayed signal of the input to itself causing what is known as interference where the waves will either sum or subtract. There are two main types of signal processor comb filters, the feed forward (FIR) and the feedback (IIR).

```
function y = fircomb(x, delay_in_ms, g, fs)


delay_in_seconds    =    delay_in_ms./1000;%
Converts delay time from ms to s

delay    =    round(delay_in_seconds.*fs);%
Changes delay from samples to seconds

Delayline=zeros(delay,1);%            memory
allocation for length
for n=1:length(x);
    y(n)=x(n)+g*Delayline(delay);
    Delayline=[x(n);Delayline(1:delay-1)];
end;
```

This is the code for the FIR filter I previously wrote for Lab Report 1. It will take a signal, a delay time and a gain factor, and send the signal forward so it will have just a single delay tail attached to it. [3]. The IIR filter however is written like so:

```
function y = iircomb(x, delay_in_ms, g, fs)


delay_in_seconds    =    delay_in_ms./1000;%
Converts delay time from ms to s

delay    =    round(delay_in_seconds.*fs);%
Changes delay from samples to seconds

Delayline=zeros(delay,1);      %      memory
allocation for length 10
for n=1:length(x);
    y(n)=x(n)+g*Delayline(delay);
    Delayline=[y(n);Delayline(1:delay-1)];
end;
```

The IIR filter takes the signal and feeds it back into the delay, but each time passing through a gain. If the gain is below 1 (which is necessary to prevent infinite and destructive signal growth) it will get quieter each time it feeds through the loop. Again it has the same parameters. [4]

Now to create different types of effects, this table has been constructed:

| Delay Range (ms) | Effect |
|---|---|
| 0 - 20 | Resonator |
| 25 - 50 | Slap back |
| >50 | Echo |

Each of these effects have no modulation and are just basic time delay effects which have unique sounds and textures to them. For example an echo occurs when there is a clear time difference between the input signal and the output signal. However with a resonator there is not so much a clear time delay, but more a colouration of the signal.

But this is only half of the effects the delay unit has proposed to create. That is because the rest of them have a modulation in them. As explained in Lab 2, The best way to describe modulation is to look at the Doppler effect. For example, when an observer is listening to a siren approach them, pass them, and then fade into the distance, the observer will note that the received frequency varies depending on how close the siren is. If it is approaching the frequency will be higher, as it is leaving the frequency will be lower than the emitted signal. This varying of the distance is essentially the same as varying the delay time. As we modulate the delay time, we create a periodical pitch variation, which is called vibrato. Although it doesn't use any delay, the vibrato effect is a good starting point to create the modulation for other effects. [5] It is coded like so:

```
function
y=vibrato(x,SAMPLERATE,Modfreq,Width)
ya_alt=0;
```

```
Delay=Width; % basic delay of input sample
in sec

DELAY=round(Delay*SAMPLERATE);    %    basic
delay in # samples

WIDTH=round(Width*SAMPLERATE); % modulation
width in # samples

if WIDTH>DELAY

  error('delay  greater  than  basic  delay
!!!');

  return;

end

MODFREQ=Modfreq/SAMPLERATE;   %   modulation
frequency in # samples

LEN=length(x);            % # of samples in
WAV-file
L=2+DELAY+WIDTH*2;          % length of the
entire delay
Delayline=zeros(L,1);  %  memory  allocation
for delay
y=zeros(size(x));        %  memory  allocation
for output vector

for n=1:(LEN-1)

   M=MODFREQ;
   MOD=sin(M*2*pi*n);
   TAP=1+DELAY+WIDTH*MOD;
   i=floor(TAP);
   frac=TAP-i;
   Delayline=[x(n);Delayline(1:L-1)];

y(n,1)=Delayline(i+1)*frac+Delayline(i)*(1-
frac); %Linear Interpolation

end
```

Using this as the starting point we can change the modulation type mixed in with our delay times to create other effects, as shown in this table:

| Delay Range (ms) | Modulation Type | Effect Name |
|---|---|---|
| 0 | Sinusoidal | Vibrato |
| 0 - 15 | Sinusoidal | Flanging |
| 10 - 25 | Random | Chorus |

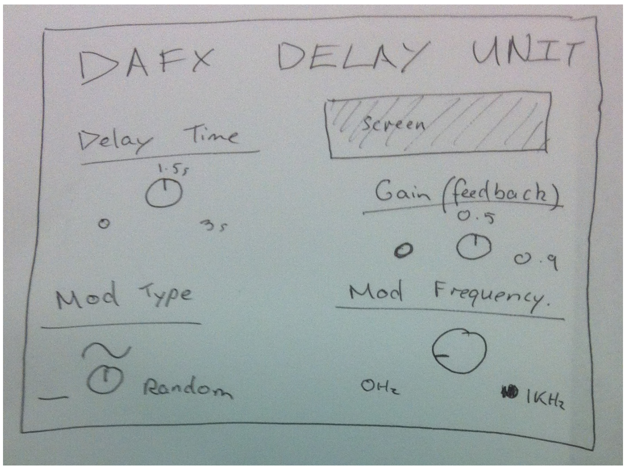
As I described in lab report 1 there will need to be a normalization section in the DSP. When using an IIR filter, it will keep feeding back to itself, as it is the nature and purpose of that filter. The gain will grow out of control if we don't stop it. So we have used what is known as the normalization coefficient (c).

$$L_2 = 1/\sqrt{1-g^2}$$
$$L_\infty = 1/(1-|g|)$$

The normalisation coefficient $c = 1/L_\infty$ stops overloading the system, where as $c = 1/L_2$ keeps the broadband signals at approximately the same loudness.

## 3. Implementation

So the signal goes into the input, passes through the signal processor and then comes out with the desired and constructed delayed effect. But practically, how is it used? Well I propose we make it both software and a hardware unit. It will be in the form of a protools plug-in in the digital domain, and both a rack unit, and guitar stomp box in the physical domain. Using pots and an on/off switch or stomp. The software would be designed to look like the hardware, but be more precise and you would be able to type in parameters as well as turn pots. It will have a screen that will tell you what the adjustable parameters are set to and using those numbers, put it through the digital algorithm. Below is a diagram of the proposed delay unit.




Using these units (software or hardware) one would be able to create their a purpose driven audio delay effect in both a live performance situation, a live mixing scenario, or to carry around with you when you're doing a recorded mix. Because of the versatility this would have a wide range target market from recording engineers, mixing engineers, performers, and live sound mixers.

## 4. Evaluation

Attached are some audio samples of an acoustic guitar going through the DSP with different parameters set showing how the different effects will sound. Using them I will evaluate my work. As you can see the DSP covers a range of effects, and unique sounds, but how will it switch between the algorithms? Well that's what the modulation pot will do. When it is set to flat, it will use the simple IIR filter. When it is set to sinusoidal it will use the vibrato algorithm, and when it is set to random it will use a modified version of the vibrato algorithm to use a random modulation.

## Conclusion

Based on these results I propose that using delay functions created with the use of IIR comb filters and modulation types is an important player in the musical domain. With a wide range of possibilities to mix and match to create your own unique sound in whatever scenario you find yourself in.

**References:**


[1] The Propagation of sound. 2012. The Propagation of sound. [ONLINE] Available at http://www.jhu.edu/virtlab/ray/acoustic.htm . [Accessed 01 April 2012].

[2] Udo Zolzer, 2002. DAFX:Digital Audio Effects. 1 Edition. Wiley, Chapter 3 Delays

[3] Feedforward Comb Filters. 2012. Feedforward Comb Filters. [ONLINE] Available at: https://ccrma.stanford.edu/~jos/waveguide/Feedforward_Comb_Filters.html . [Accessed 25 March 2012].

[4] Feedback Comb Filters. 2012. Feedback Comb Filters. [ONLINE] Available at: https://ccrma.stanford.edu/~jos/waveguide/Feedback_Comb_Filters.html . [Accessed 25 March 2012].

[5] The Vibrato Page. 2012. The Vibrato Page. [ONLINE] Available at: http://www.standingstones.com/vibrtopg.html. [Accessed 01 June 2012].