Digital Audio Systems

Review #2
Tape Delay/Echo Emulation

SID: 420 180 999

Name: Jarad Avnell

Unit Code: DESC91151

Lecturers: William Martens

Due Date: 7th June, 2013
An echo is a naturally occurring phenomenon where an original audio signal is repeated after a duration of time. “A sound wave reflected by a wall will be superimposed on the sound wave at the source. If the wall is far away, such as a cliff, we will hear an echo. If the wall is close to us, we will notice the reflections through a modification of the sound color”. If reflections are continually repeated between two parallel surfaces a flutter echo will result (Zolzer, 2002. p63).

Echo and reverberation should not be confused however, as an “echo is the periodic repetition of the same sound signal”, where “reverberation is the modulation of a single sound signal to produce weaker ghost frequencies or depth” (Holmes, 2002. p81).

The effect of naturally occurring echoes can be simulated digitally using a variety of techniques. The simplest, and most common method is via the use of comb filters. Short delays can be used to simulate these echoes. “If the delay is in the range 10 to 25 ms, we will hear a quick repetition named slap back or doubling. If the delay is greater than 50 ms we will hear an echo” (Zolzer, 2002. p69). Delays differ from echoes when the time difference between the original signal and the repeated signal is longer still. Delays have the ability to repeat over extremely long periods of time, and have the capabilities of running much longer than a few milliseconds. This effect is artificial, and does not appear in nature.

The network that simulates a single delay is called the Finite Impulse Response (FIR) comb filter. This effect adds the processed signal to the input signal after a given amount of time (Zolzer, 2002. p63). Like acoustical delays, the FIR comb filter has an effect both in the time and frequency domains. “Our ear is more sensitive to the one aspect or to the other according to the range where the time delay is set”. For larger values of time we can hear an echo that is distinct from the direct signal. Below is the Matlab code for a FIR filter, as outlined by Zolzer:

```matlab
x=zeros(100,1); x(1)=1; % unit impulse signal of length 100
g=0.5;
Delayline=zeros(10,1); % memory allocation for length 10
for n=1:length(x);
    y(n)=x(n)+g*Delayline(10);
    Delayline=[x(n); Delayline(1:10-1)];
end;
(Zolzer, 2002. p64).
```

The above code simulates a single delay:
- The input signal is delayed by a given time duration, \( \tau \).
- The delayed (processed) signal is added to the input signal with the addition of amplitude gain, \( g \).
- The difference equation is: \( y(n)=x(n)+gx(n-M) \)
- The difference equation means that the output is equal to the input + (input multiplied by the gain argument and a delay argument).
- \( M \) is the time duration of the delay, and is expressed as \( \tau/\text{fs} \)
A similar function to the finite impulse response comb filter is the infinite impulse response (IIR) comb filter, which will produce an endless series of responses as opposed to the single response that a FIR will create. The input signal circulates in a delay line that is fed back to the input. Each repeat into the delay line will be attenuated. Below is the Matlab code for an IIR filter, as outlined by Zolzer:

```matlab
x=zeros(100,1); x(1)=1;  \% unit impulse signal of length 100
g=0.5;
Delayline=zeros(10,1);  \% memory allocation for length 10
for n=1:length(x);
    y(n)=x(n)+g*Delayline(10);
    Delayline=[y(n);Delayline(1:10-1)];
end;
```

(Zolzer, 2002. p65)

This above code simulates endless reflections between two surfaces:
- We get an endless series of responses, \(y(n)\) to input, \(x(n)\).
- The input signal circulates in delay line (delay time \(\tau\)) that is fed back to the input.
- Each time it is fed back it is attenuated by \(g\).
- The input can often be padded by \(c\) to compensate for high amplification of the structure. This measure is to ensure the feedback section does not exceed the volume of the input section
- The difference equation is: \(y(n)=Cx(n)+gy(n−M)\)
- Once again, \(M\) is the time duration of the delay, and is expressed as \(\tau/\text{fs}\)
- The transfer function is: \(H(z) = c1 − gz − M\)


Before digital technology where delays and echoes can be simulated using programs like Matlab, delays and echoes were often created using tape recorders. To create echo with a tape recorder, the playback output signal of the machine is fed back into the input, or record head section of the same machine. “When this connection is made and the tape recorder is simultaneously recording and playing back, the sound being played is immediately recorded by the record head.” Continuing in this manner without interruption creates the echo effect. The duration of the echo or delay is determined by the distance that the tape must travel from the record head to the playback, and the strength or persistence of the echo (or how many repeats you hear) is determined by the amplitude of the playback signal being fed back into the recorder. The strength of the delay is determined by the signal gain of the record head. A stronger feedback signal results in a longer the sequence of repeats (Holmes, 2002. p81). To achieve longer delay times obviously requires larger gaps between the record and playback heads, and is often achieved by recording and rerecording a sound using a combination of tape recorders using a single length of tape between them. “A sound is recorded on the first machine and played back on the second, creating a long delay between the first occurrence of the sound and its repetition on the second machine. If the sound being played back on the second machine was simultaneously recorded by the first machine, an extended echo effect was created with long delays between successive, degenerating repetitions” (Holmes, 2002. p82).
The most important sonic characteristic of an analogue tape delay is the degradation of sound quality after each repeat, which serves to be the primary difference between a digital delay and an analogue tape delay. It can be seen above that creating an analogue tape echo or delay is a little more complicated than repeating an exact replication of an input signal using FIR and IIR comb filtering. The easiest way to emulate tape delay digitally is by incorporating a filter in the feedback section of a delay, which will serve to degrade the repeating signal over time. Many Analogue filters have been explored in Wanhammar’s book “Analog Filters using Matlab”, and many of these filters are readily available in the Matlab’s built-in feature, the “signal processing toolbox”. Some of the appropriate filters included are: Bessel, Butterworth, Chebyshev, Elliptic, Bilinear Transformation.

Each of the above filters may be incorporated, and each will yield similar, but different results and the filters mentioned above would have their own advantages and disadvantages, generally varying in the steepness that can be achieved or the possibility that ripples will appear. When reviewing a study by Holgersson on filter design, a conclusion was made that the Butterworth filter is the most likely candidate to represent an analogue tape decay based on its smooth contour. Other filters contain anomalies and ripples either before or after the filter’s cut off point. In the case of the elliptic filter, ripples occur both before and after the cut off frequency.

The above graph shows the result of four different low-pass filters. The x-axis displays frequency and the Y-axis displays the amplitude of that frequency. (Holgersson, 2011. p207)
The design of the digital Butterworth filter is based off the analogue circuit studied by Butterworth in 1930. Butterworth was attempting to create an “ideal electrical filter [that] should not only completely reject the unwanted frequencies, but should also have uniform sensitivity for the wanted frequencies”, meaning that the frequencies situated before the cut off frequency of the filter will remain unaffected by the filters design. (Butterworth, 1930. Section 1.1).

![Diagram of Butterworth filter](image)

**Fig. 3.**

The figure above shows an analogue low-pass filter being implemented with various amounts of resistors and capacitors. The A line contour shows the least amount of resistance while E shows a larger resistance with the inclusion of capacitance. (Butterworth, 1930. Section 2.1)

The Butterworth filter is a built in feature in Matlab’s Signal Processing Toolbox, which features a number of different filter designs. The Butterworth filter can be implemented in a number of ways some of which have been examined below:
The code above calls MATLAB’s built-in Butterworth feature:

- Two syntaxes have been examined above: `[b,a]=butter(n,Wn);` and `[b,a]=butter(n,Wn,’ftype’);
- `b` represents the numerator coefficients of the filter
- `a` represents the denominator coefficients of the filter
- The `n` argument for the Butterworth filter is the order (2nd order in all of the above cases)
- `Wn` is the cut off frequency. This must be a number between 0 and 1, where 1 corresponds to the Nyquist frequency.
- In cases where ‘ftype’ is used:
  - ‘high’ is for a high-pass digital filter with normalized cut-off frequency `Wn`
  - ‘low’ is for a low-pass digital filter with normalized cut-off frequency `Wn`
  - ‘stop’ is for a order 2*n band-stop digital filter if `Wn` is a two-element vector, `Wn = [w1 w2]`. The stopband is `w1 < ω < w2`.
- The transfer function is: \( H(s) = \frac{z(s)}{p(s)} = \frac{k}{(s – p(1))(s – p(2))\ldots(s – p(n))} \) (Mathworks, [N.D.], Section 2.1)

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Command</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowpass</td>
<td><code>[b,a]=butter(2,.1);</code></td>
<td>Applies a second order filter that passes all frequencies below 0.10</td>
</tr>
<tr>
<td>Highpass</td>
<td><code>[b,a]=butter(2,.1,’high’);</code></td>
<td>Applies a second order filter that passes all frequencies above 0.10</td>
</tr>
<tr>
<td>Bandpass</td>
<td><code>[b,a]=butter(2,[.1,.2]);</code></td>
<td>Applies a second order filter that passes all frequencies between 0.10 and 0.20</td>
</tr>
<tr>
<td>Bandstop</td>
<td><code>[b,a]=butter(2,[.1,.2],’stop’);</code></td>
<td>Applies a second order filter that stops all frequencies between 0.10 and 0.20</td>
</tr>
</tbody>
</table>

(SMU, [N.D.], p2)
By implementing the Butterworth filter into the feedback section of the IIR filter mentioned above, the user can begin to emulate the decaying characteristic of repeats created by a tape delay. It has been noted that the operating parameters of magnetic recording tape will differ appreciably from one manufacturer to another, as well as between different types of tape from the same manufacturer. “Different tapes have slightly different oxide thicknesses and formulations, which affect the overall transfer characteristic of the magnetic domains. These effects are most noticeable in the high frequency response of the tape, as well as its overall sensitivity” (Kefauver, 2001. p288). With this in mind, the user may need to experiment with a number of filter parameters before reaching the desired result. Soft tape degradation can be achieved by the implementation of the Lowpass filter option, while heavier degradation on repeats can be simulated using the Bandpass option. Below is an example of the inclusion of a 2\textsuperscript{nd} order Low Pass Butterworth filter to feedback section of an IIR comb filter:

```
Delayline=butter+([y(n);Delayline(1:10-1)]2,0.1);
```

**Further Considerations:**

Listed above are just a few elements to be considered when emulating a tape delay, however before a tape delay can truly be emulated, further characteristics of tape machines must be investigated. Holmes experimented with turning up the playback section of a tape machine to the point of overload before it is rerecorded. This course of action produces echo “frizz” or repeating waves of white noise that eventually overpower the original signal (2002. p81). One suggestion for implementing this effect successfully into the code may be the addition of a gain feature in the feedback section of the delay.

Abel, Arnardottir, & Smith observed effects which can be achieved by manipulating a tape machines controls during its record and playback process. One of these effects is where the delay handle of an Echoplex is moved very quickly away from the playback head. When this happens, the tape bias signal can be Doppler shifted down into the audio band. Artists such as Jimmy Page and Tommy Bolin often employed this technique. Another interesting effect is where the Delay Handle is placed as far away from playback head as possible, it is then moved back towards the playback head at a faster rate than the tape speed. The handle will stop eventually, which means that at some point, the tape and record head are moving at the same speed. “At this point the bandwidth of the signal written to the tape is unbounded, and a “sonic boom” can result, appearing as a discontinuity at the output” (2008. Section 1.6).
References:


SMU. [N.D.]. *Matlab Filter Instructions*. Southern Methodologist University. Dallas. USA

