Course Name: Digital Signal Processing Design

Course Number and Section: **14:332:447:01**

Dereverberation: Removing Unwanted Echoes and Reverb from Recorded Audio

Part 5 of 6: Multi-Tap Delay

Submitted by: Lance Darragh

Digital Audio Effects, Multi-Tap Filter

The idea of adding a delayed, and scaled, copy of the input signal can be generalized to include any arbitrary number of delays (within processing time limitations between samples). Here we will illustrate this by creating a filter with two delays fed back into the input, along with these same delays being sent to the output but with different scaling factors, b_1 and b_2. The transfer function will be:

(32) $$H(z) = b_0 + \frac{b_1 z^{-D_1} + b_2 z^{-(D_1+D_2)}}{1 - a_1 z^{-D_1} - a_2 z^{-(D_1+D_2)}}$$

showing that the delays involved will be D_1 and $D_1 + D_2$. Again b_0 here represents the direct sound coefficient. The corresponding sample processing algorithm can be found in Appendix A.5. The graphs of the input and output signals are shown below. It should be noted that the value $|a_1| + |a_2|$ should be less than one to maintain stability within the feedback network.

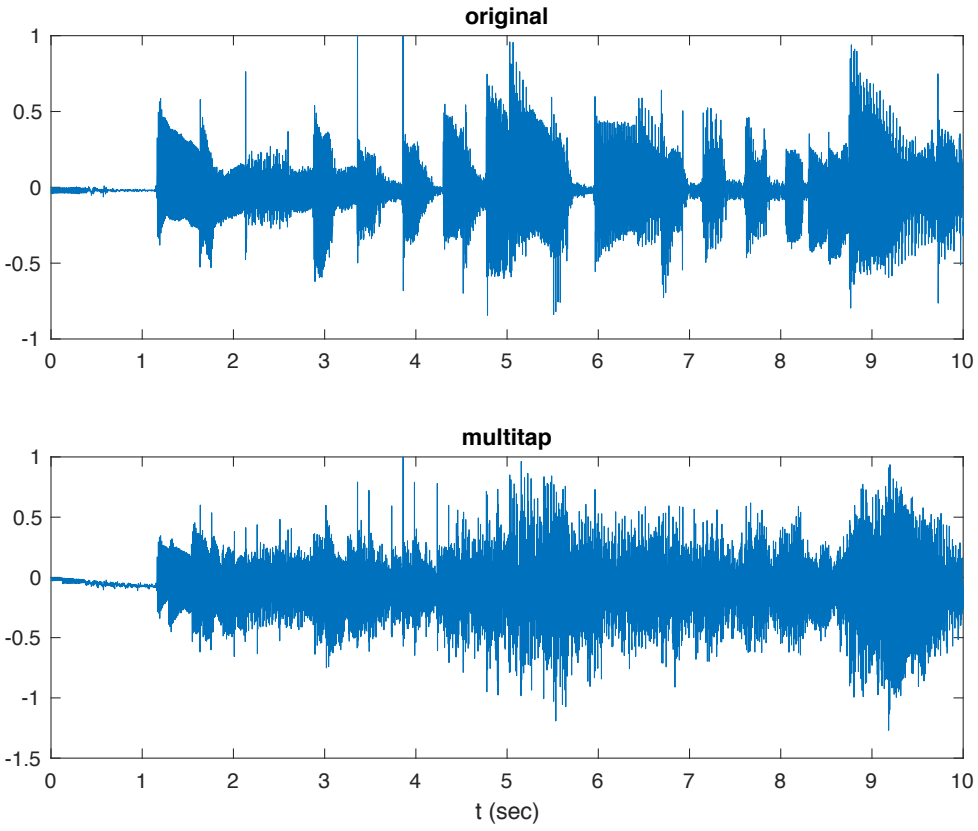

Table of Contents

Appendix A.5: Multi-tap Delay

```
% Import original audio
[s,fs] = audioread('original.wav');
% Delay values
D1 = 0.125*fs; D2 = 0.25*fs;
% Amplitude coefficients
b0 = 1; b1 = 1; b2 = 1;
a1 = 0.2; a2 = 0.4;

% Determine length of input signal
[N,k] = size(s);

% Internal delay buffer for s(n)
w = zeros(1, D1 + D2 + 1);
% Delay buffer index variable
q = 1;
% Delay buffer taps
tap1 = D1 + 1;
tap2 = D1 + D2 + 1;
% Loop through input signal
tic
for n = 1:N
    s1 = w(tap1);
    s2 = w(tap2);
    y(n) = b0*s(n) + b1*s1 + b2*s2;
    w(q) = s(n) + a1*s1 + a2*s2;
        q = q - 1;                  % Backshift index 1
    if q < 1                      % Circulate index 1
        q = D1 + D2 + 1;
    end
    tap1 = tap1 - 1;               % Backshift tap1
    if tap1 < 1                    % Circulate tap1
        tap1 = D1 + D2 + 1;
    end
        tap2 = tap2 - 1;           % Backshift tap2
    if tap2 < 1                    % Circulate tap2
        tap2 = D1 + D2 + 1;
    end
end
toc
% Normalize y(n)
ymax = max(y);
y = y/ymax;
```

```
% Check results
sound(y,fs);
```

Compare to filter Function

```
n = [b0, zeros(1,D1 - 1), b1-a1*b0, zeros(1,D2 - 1),b2 - a2*b0];
d = [1, zeros(1,D1 - 1), -a1, zeros(1,D2 - 1),-a2];

tic
yfilt = filter(n,d,s);
toc

% Normalize yfilt(n)
yfiltmax = max(yfilt);
yfilt = yfilt/yfiltmax;
sound(yfilt,fs);
```

Plot Results

```
t = 1:160000;
subplot(2,1,1)
plot(t/fs,s),title('original')
subplot(2,1,2)
plot(t/fs,y); title('multitap'), xlabel('t (sec)');
```

Output Results

```
audiowrite('Multi-Tap.wav',y,fs)
```

Published with MATLAB® R2017b