Lab 15: Digital Audio Effects using Matlab

This report is in two parts. The first part uses Matlab to implement digital audio echo effects. The second part uses a high-order FIR filter to filter out an interfering signal.

Part (a)

Echo and reverberation effects are used extensively in the music industry. Matlab is used in this assignment to investigate the effects, off-line, the Matlab algorithms have, on a segment of speech. You may record your own voice in single channel mode with 8-bit resolution. The sampling frequency is 8 kHz(125us period) and you may save the voice file as a '.wav' file. Connect the speakers to the green socket and the microphone to the red socket. Both sockets are located at the back of the PC. Use the Microsoft recorder to record your voice but set the sound properties for a sample rate of 8000 Hz and mono. Save the file to the sub directory work in the Matlab directory.

Echo Filter

The difference equation for a simple echo filter

$$y(n) = x(n) + a.x(n - D)$$

Take the z-transform of this difference equation

$$Y(z)=X(z) + a.X(z)Z^{-D}$$

The transfer function for this filter:

$$H(z) = \frac{Y(z)}{X(z)} = 1 + aZ^{-D}$$

The impulse response for this filter is:

$$h(n) = \delta(n) + a.\delta(n - D)$$

Choose *D* so that the signal is delayed by 0.375s and *a* = 0.5. Plot the frequency response of the filter using the mfile freqz.m. Try smaller values of *D* and listen to your delayed signal until you can no longer distinguish a separate echo, which corresponds to reverb. Record this time.

Matlab code for implementing filter:

```
[x,fs] = wavread('c:\matlab\work\waed.wav');%read your waed.wav
% Calculate the number of samples in the file
xlen = length(x);    %Initialize all constants
a = 0.5;
delay = 0.4;
D = delay*fs;
% create a matrix , the size of x for efficient and speedier operation
y = zeros(size(x)); % filter the signal
        for i  = D + 1 : 1 : xlen;
                y(i) = x(i) + a*x(i-D);
        end;
sound(y, fs);          %play the echo signal
plot(y)                % add labels on plot…use xlabel and ylabel
f=fft(y)               % get the fast fourier transform of sound file
whos                   % checks the variables used and the size check out f
plot(abs(f))  %plot magnitude of the fft(frequency content of speech)
plot(abs(f(1:50000)))      % plots a section of the fft
```

Part (b)

Add the following commands to your existing mfile above and save under a different name. Construct the interfering signal using the following Matlab commands

```
%speechfi.m Filtering speech signal with stopband FIR filter used to
%remove interference.
echo on
% read in some speech samples
[s1,fs] = wavread('c:\matold\toolbox\waed2dem\waed1.wav');%read in
your waed.wav
figure(1)
plot(s1)
title('Speech  Samples');
fs = 8000;                     %sample rate
sound(s1, fs);                 % play sound
pause
figure(2)
ppse(s1,fs,'Speech Spectrum'); % make spectral estimate
pause
n=1:size(s1);
s2 = s1' + 10*cos(2*pi*1000/fs*n);          %add 1000Hz tone
```

```
figure(3)
plot(s2)
pause
figure(4)
ppse(s2, fs, 'Spectrum');        %look at spectrum
pause;
sound(s2, fs);                   %listen to corrupted speech signal
a=[1];
% design stop band filter
b=fir1(50, 2*[800/fs  1200/fs], 'stop');
figure(5)
freqz(b,a);                      % plot the response
title('First filter response')
pause
s3=conv(s2, b);                  %carry out filtering
figure(6)
plot(s3);
title('Speech after 1st Filter');
sound(s3, fs);                   %listen to it!
pause

b=fir1(90, [800/4000  1200/4000], 'stop');

figure(7)
freqz(b,a);                      % plot response
title('Response of second filter ');
pause;

s3 = conv(s2, b);

figure(8)
plot(s3)
sound(s3)
```