

## MATLAB Lab. FIR and IIR Filters

### Goals

Filters can be used to shape the signal spectrum in a desired way. The goals of this Lab is to learn how to design, analyze, and apply a digital filter to the data. It will also help you to learn

- how to compensate for the delay introduced by a filter,
- how to avoid distorting the signal,
- how to remove unwanted content from the signal.

### Filter Configurations

All the filter design functions in the Signal Processing Toolbox operate with normalized frequencies, so that they do not require the system sampling rate as an extra input argument. The normalized frequency is always in the interval  $0 < f < 1$ . To convert the normalized frequency back to Hertz, multiply by half the sample frequency ( $F_s/2$ ). To convert normalized frequency to angular frequency, multiply by  $\pi/\text{sample}$ .

### Experiment 1- Procedures

- (1) Generate a noisy electrocardiogram signal.

```
Fs = 500;           % sample rate in Hz
N = 500;           % number of signal samples
x = ecg(N)+0.25*randn(1,N); % noisy waveform
t = (0:N-1)/Fs;    % time vector
```

- (2) Design a 70<sup>th</sup> order FIR lowpass filter with cutoff frequency of 75 Hz to remove high frequency noise above 75 Hz. Plot the magnitude and phase responses.

```
Fn = 75/(Fs/2);    % Normalized cutoff frequency
b=fir1(70,Fn);
y=filter(b,1,x);
freqz(b)
```

- (3) Plot  $x$  and  $y$  in the same window for comparison. Observe the group delay in the signal  $y$ . Print the figures.

```
figure
subplot(2,1,1); plot(t,x)
xlabel('Time (s)')
legend('Original Noisy Signal');
grid on
title('Observe the Group Delay')
subplot(2,1,2); plot(t,y,'r')
xlabel('Time (s)')
legend('Filtered Signal');
grid on
```

- (4) Plot the group delay to verify that it is constant across all frequencies indicating that the filter is linear phase. Print the figure. Use the group delay to measure the delay of the filter.

```
figure
grpdelay(b,2048,Fs) % plot group delay
D = mean(grpdelay(b)) % filter delay in samples
```

- (5) Append  $D$  zeros at the end of the input data vector  $x$ , filter the data and compensate for the delay by shifting the output signal  $y_1$  by  $D$  samples. This effectively removes the group delay.

```
x1=[x, zeros(1,D)];
y1 = filter(b,1, x1); % Append D zeros to the input data
y1 = y1(D+1:end); % Shift data to compensate for delay
```

- (6) Plot  $x$  and  $y_1$  in the same window for comparison. Observe that the group delay is removed. Print the figures.

```
figure
subplot(2,1,1); plot(t,x)
xlabel('Time (s)')
legend('Original Noisy Signal');
title('Observe the Filtered Signal w/o Group Delay')
subplot(2,1,2); plot(t,y1,'r')
xlabel('Time (s)')
legend('Filtered Signal');
grid on
```

## Experiment 2- Procedures

- (1) Load the audio file, play the audio file.

```
load chirp;
sound(y)
pause(4)
```

- (2) Add band-limited white noise to the signal y, play the resultant audio file.

```
Nois=0.25*randn(length(y),1);
Bp=fir1(46,[0.1,0.45]);
BpNois=filter(Bp,1,Nois);
ynois=y+BpNois;
sound(ynois)
pause(4)
```

- (3) Design a 46th order FIR highpass filter with cutoff frequency of 0.45 ( $F_s/2$ ) to remove the noise band limited to 0.1-0.45 ( $F_s/2$ ). Play the filtered audio file.

```
Bh=fir1(46,0.45,'high');
yy=filter(Bh,1,ynois);
sound(yy)
```

- (4) Plot y, ynois and yy in the same window for comparison (using *subplot*). Print the figures.

```
t=(0:length(y)-1)/Fs;
figure
subplot(3,1,1);plot(t,y)
subplot(3,1,2);plot(t,ynois)
subplot(3,1,3);plot(t,yy)
```

- (5) Repeat the Experiment-2 by replacing FIR filter with IIR 6<sup>th</sup> order Butterworth filter in steps (2)-(3).