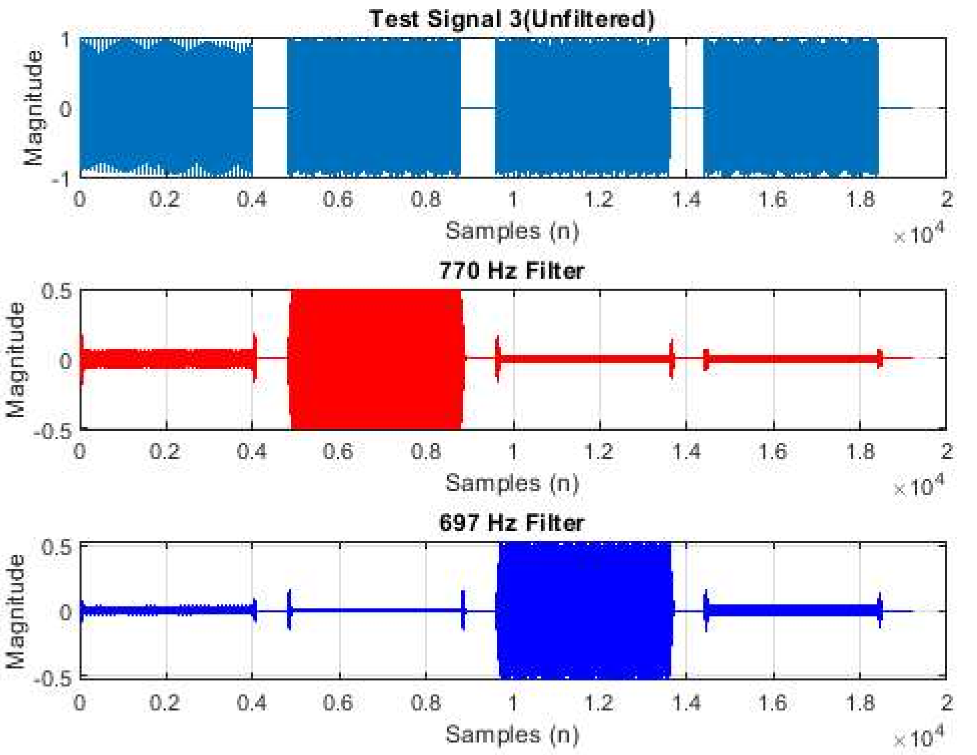
Lab 4

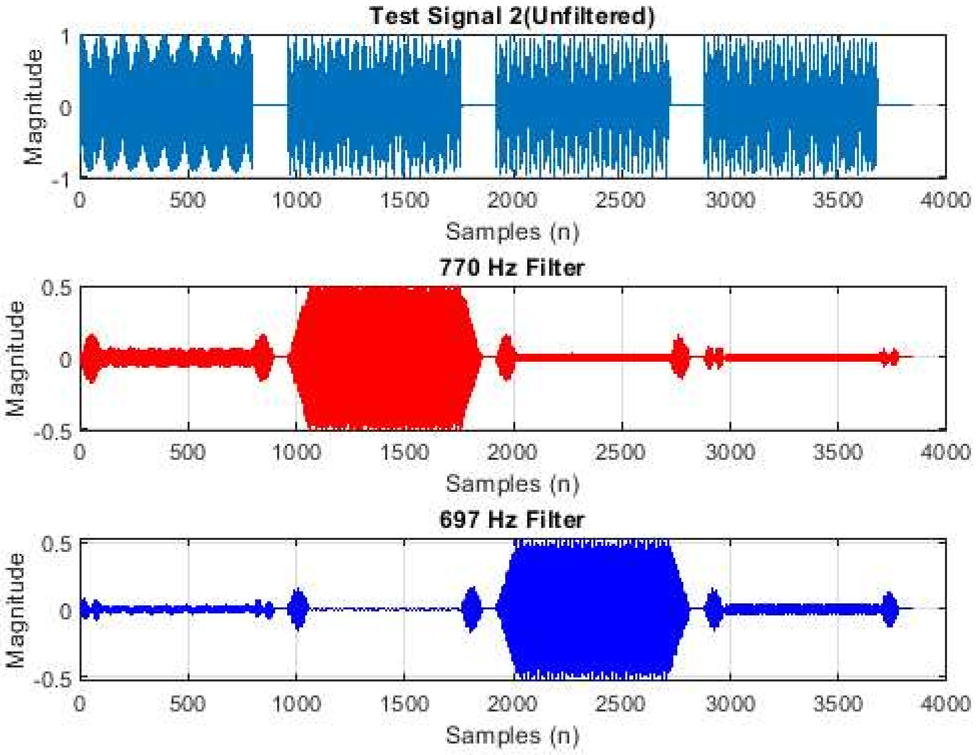
Contents

Part I:

```
dial_scu = [3 5 7 11] ;

testSig1 = my_dtmf(0.25, 0.05, 8000, dial_scu);
testSig2 = my_dtmf(0.10, 0.02, 8000, dial_scu);
testSig3 = my_dtmf(0.50, 0.10, 8000, dial_scu);
testSig4 = my_dtmf(0.25, 0.05, 8000, [1:12]);

% Create b697 and b770
N = 100;
w = 20;
fs = 8000;
fc770 = 770;
wn770 = [fc770-w, fc770+w]/(fs/2);

fc697 = 852;
wn697 = [fc697-w, fc697+w]/(fs/2);

b770 = fir1(N, wn770, rectwin(N+1) );
b697 = fir1(N, wn697, rectwin(N+1) );

y770sig3 = filter(b770, 1, testSig3);
y697sig3 = filter(b697, 1, testSig3);

figure()
subplot(3,1,1)
plot(testSig3)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 3(Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig3, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig3, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on

y770sig2 = filter(b770, 1, testSig2);
```

```
y697sig2 = filter(b697, 1, testSig2);

figure()
subplot(3,1,1)
plot(testSig2)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 2(Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig2, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig2, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on


% Our filter does well with most of the signals when we tested both
% n=200,400 with 400 having less noisy signals outputted

% It supressses most of the DTMF tones and leavees us with the tone that
% we want.

% the tones have less delay and as a result some of the noise is filtered
% through the delay  period.
```



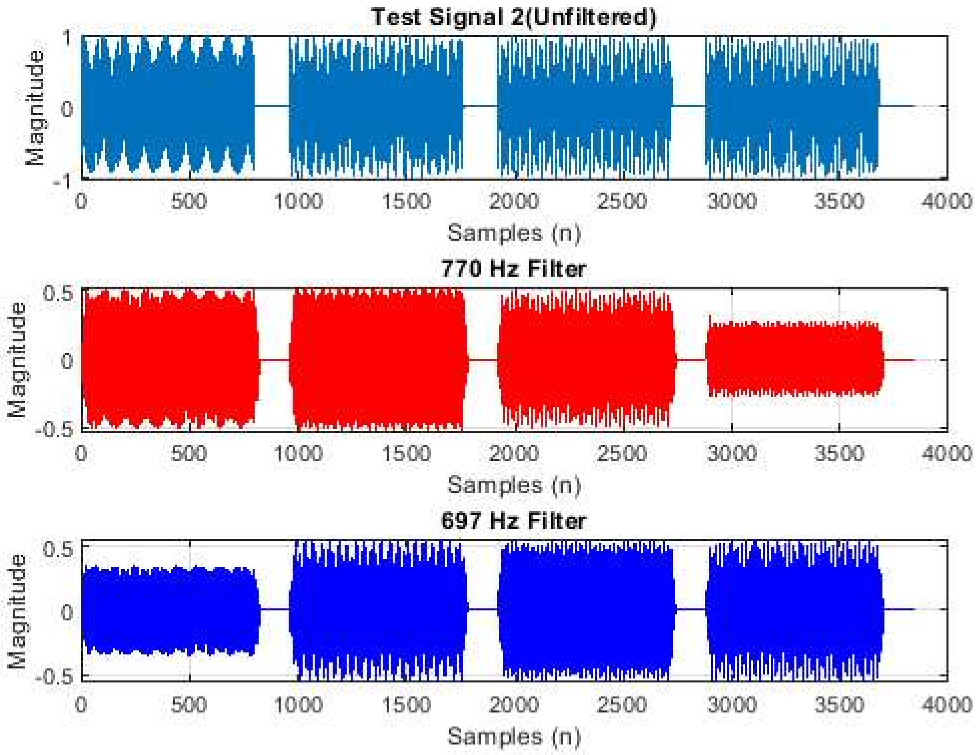


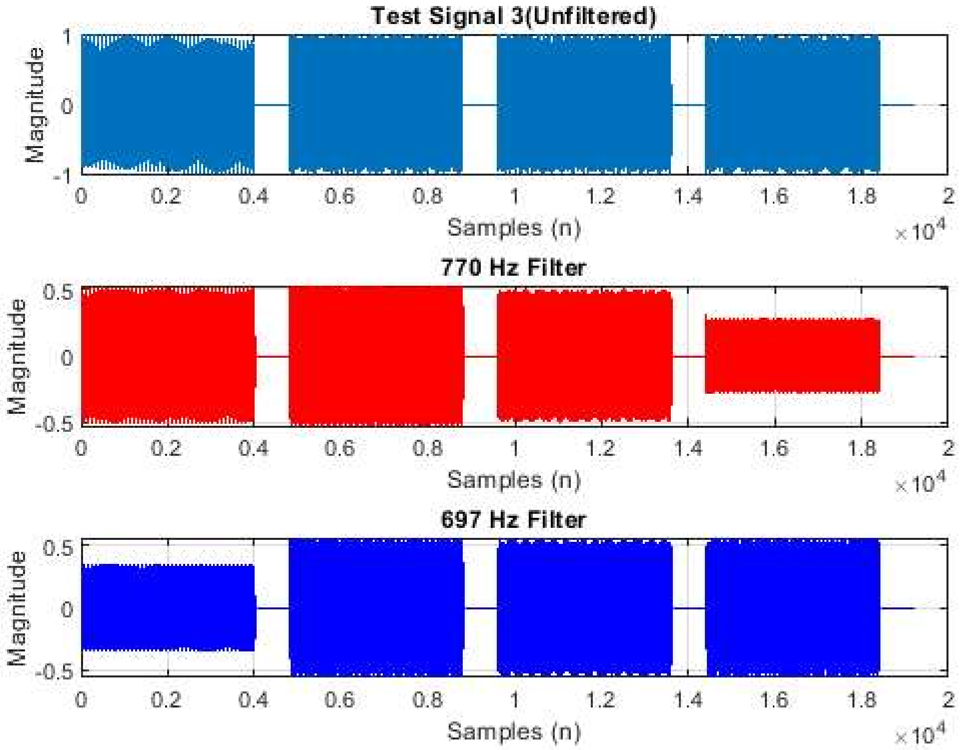
Part II:

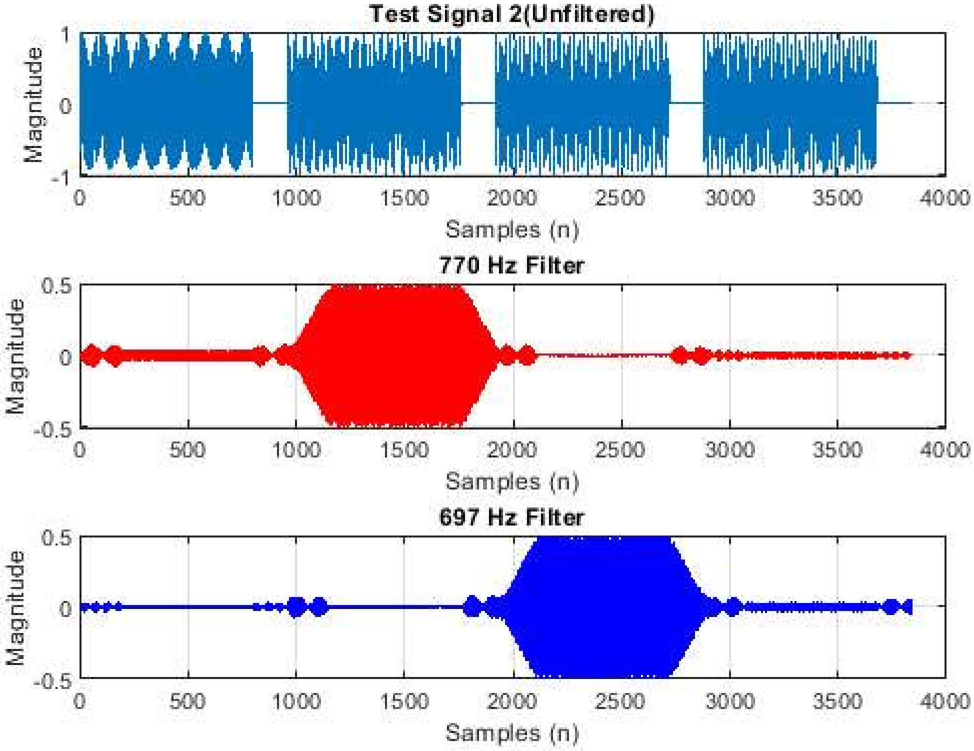
```
N = 100/4;

b770 = fir1(N, wn770, rectwin(N+1) );
```

```
b697 = fir1(N, wn697, rectwin(N+1) );

y770sig3 = filter(b770, 1, testSig3);
y697sig3 = filter(b697, 1, testSig3);
% Test Sig2

y770sig2 = filter(b770, 1, testSig2);
y697sig2 = filter(b697, 1, testSig2);

figure()
subplot(3,1,1)
plot(testSig2)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 2(Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig2, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig2, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on

figure()
subplot(3,1,1)
plot(testSig3)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 3(Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig3, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig3, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on

% How do these results compare to Step 1 results?
% they are not as effective as previously seen in step 1 because we get
% more of the unwanted tones.


%Step 2 part2
N = 100*2;
```

```
b770 = fir1(N, wn770, rectwin(N+1) );
b697 = fir1(N, wn697, rectwin(N+1) );

y770sig3 = filter(b770, 1, testSig3);
y697sig3 = filter(b697, 1, testSig3);
% Test Sig2

y770sig2 = filter(b770, 1, testSig2);
y697sig2 = filter(b697, 1, testSig2);

figure()
subplot(3,1,1)
plot(testSig2)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 2(Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig2, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig2, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on

figure()
subplot(3,1,1)
plot(testSig3)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 3 (Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig3, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig3, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on

% How do these results compare to Step 1 results? Is there any
% noticeable benefit to increasing the length of the filter?
% they are more effective than previously seen in step 1 because we get
% less of the undesired signal and noise, but not too much to the  point
% where more of the unwanted signal is being passed.
% Based on these results, would you modify your choices for N and w in your filters?
% we would want a value of around 300 with a similar value of w to get
% close to the optimal result of our wanted and unwanted tones
N = 315;

b770 = fir1(N, wn770, rectwin(N+1) );
b697 = fir1(N, wn697, rectwin(N+1) );

y770sig3 = filter(b770, 1, testSig3);
y697sig3 = filter(b697, 1, testSig3);
% Test Sig2

y770sig2 = filter(b770, 1, testSig2);
y697sig2 = filter(b697, 1, testSig2);

figure()
subplot(3,1,1)
plot(testSig2)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 2(Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig2, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig2, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on

figure()
subplot(3,1,1)
plot(testSig3)
xlabel('Samples (n)')
ylabel('Magnitude')
title('Test Signal 3 (Unfiltered)')
grid on

subplot(3,1,2)
plot(y770sig3, 'red')
xlabel('Samples (n)')
ylabel('Magnitude')
title('770 Hz Filter')
grid on

subplot(3,1,3)
plot(y697sig3, 'blue')
xlabel('Samples (n)')
ylabel('Magnitude')
title('697 Hz Filter')
grid on
```

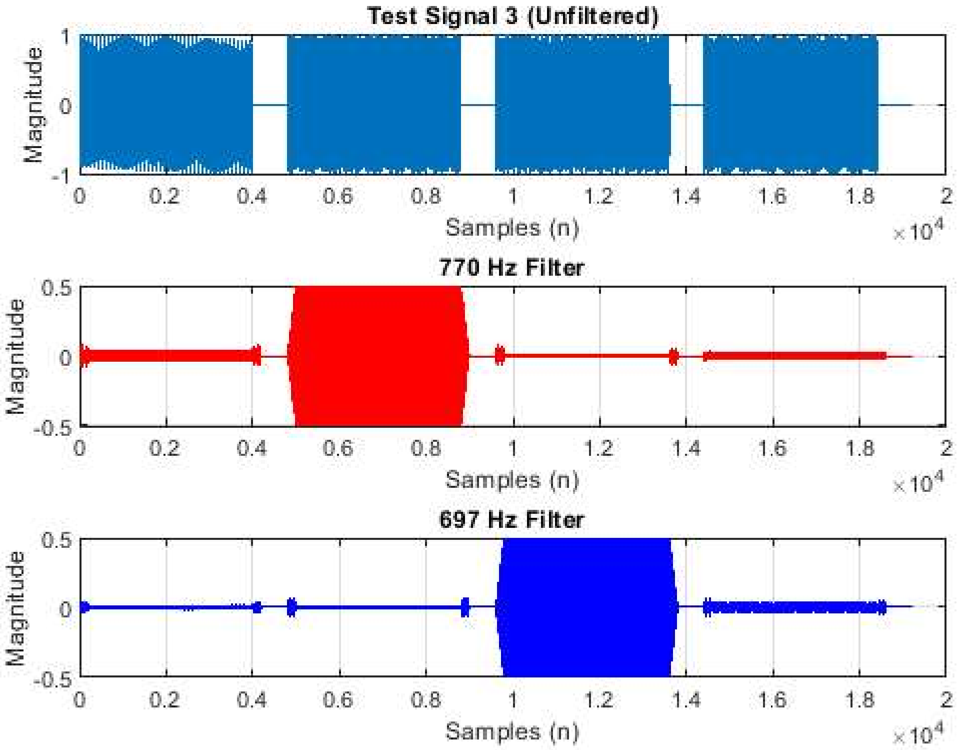

Test Signal 2(Unfiltered)
Magnitude
1
0
-1
0
500
1000
1500
2000
2500
3000
3500
4000
Samples (n)
770 Hz Filter
Magnitude
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000
3500
4000
Samples (n)
697 Hz Filter
Magnitude
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000
3500
4000
Samples (n)


Test Signal 3(Unfiltered)
Magnitude
1
0
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Samples (n)
×10^4
770 Hz Filter
Magnitude
0.5
0
-0.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Samples (n)
×10^4
697 Hz Filter
Magnitude
0.5
0
-0.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Samples (n)
×10^4


Test Signal 2(Unfiltered)

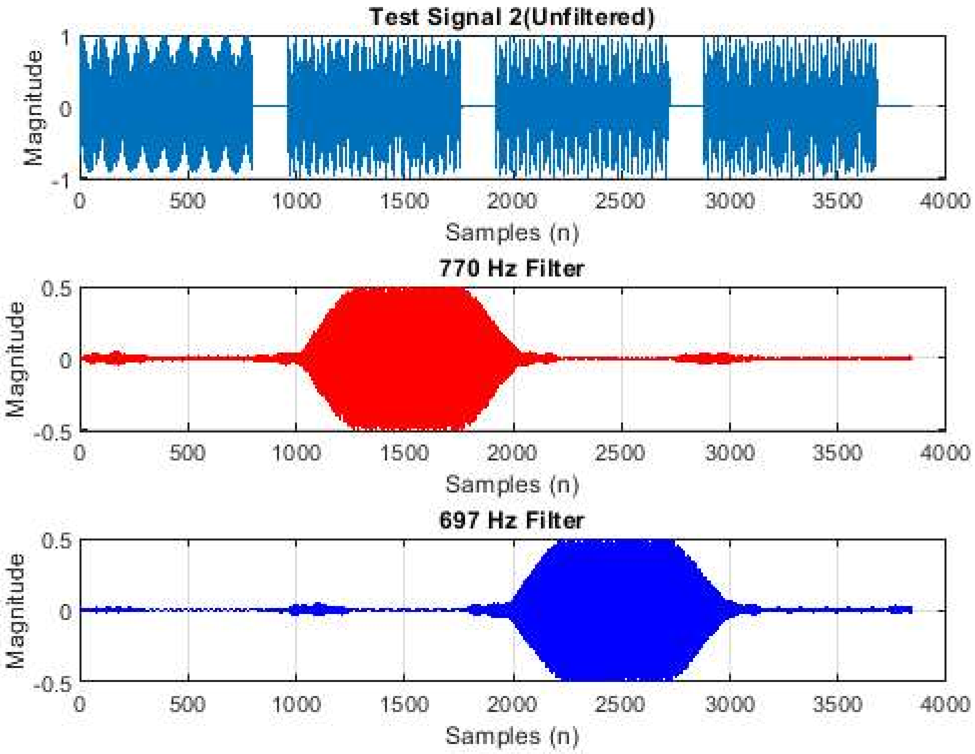
770 Hz Filter

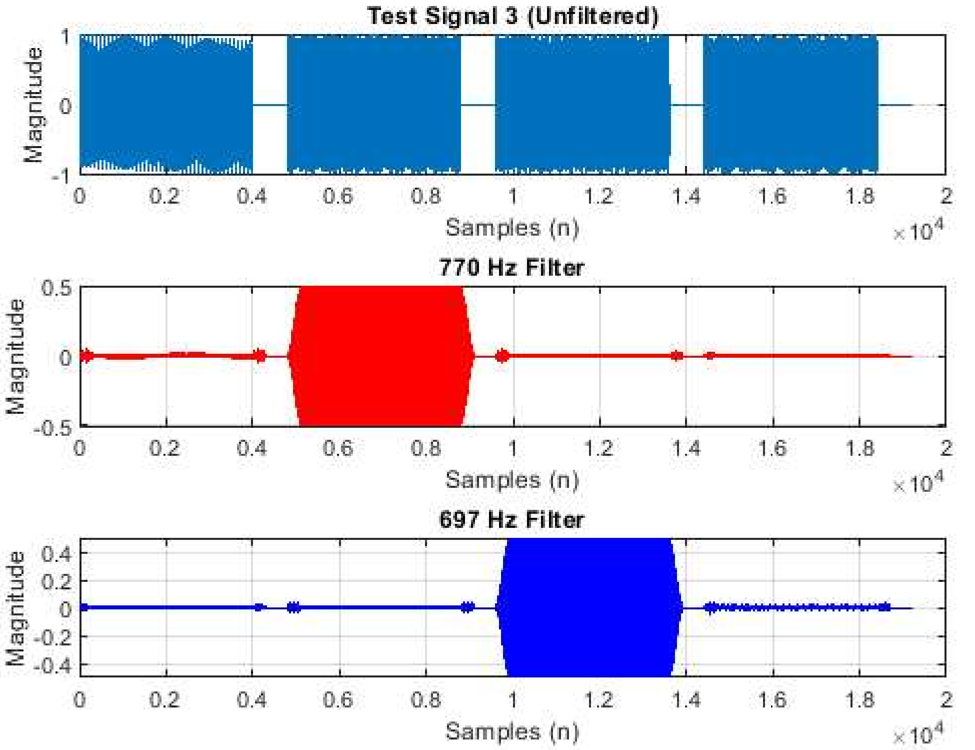
697 Hz Filter


Test Signal 3 (Unfiltered)

770 Hz Filter

697 Hz Filter

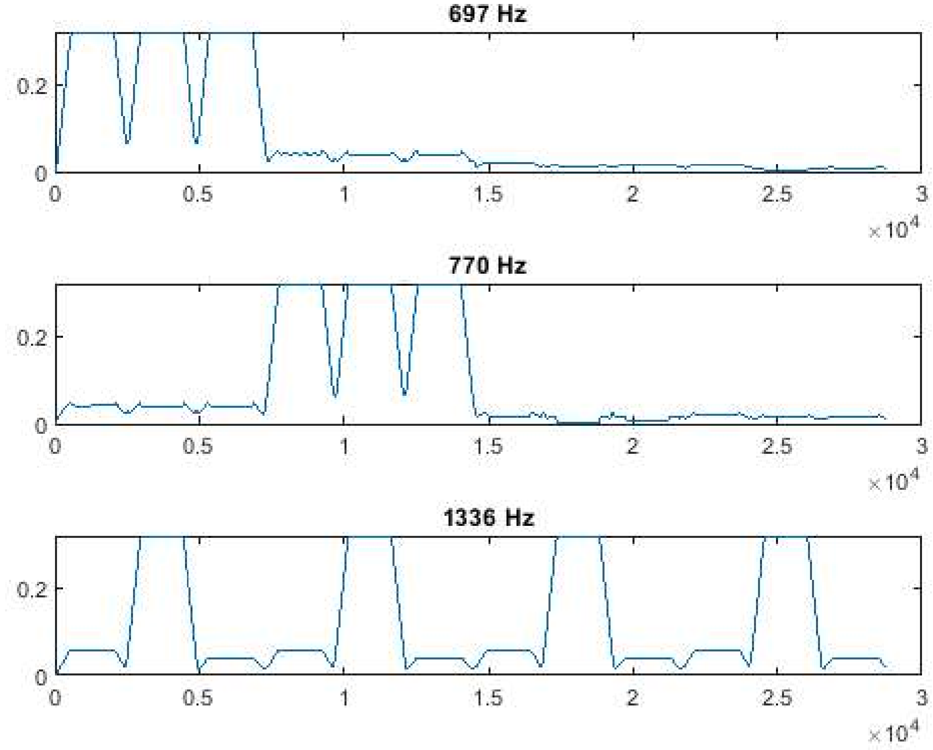




Part 3:

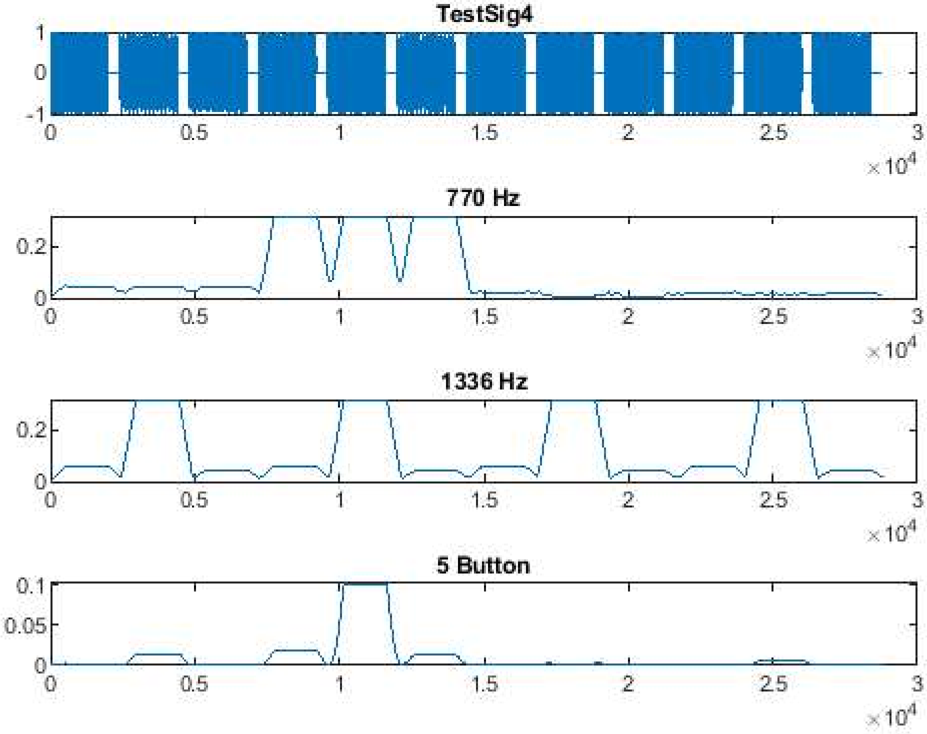
```
fs = 8000;
N = 100;
```

```
fc697 = 697;
fc770 = 770;
fc1336 = 1336;

wn697 = [fc697-w, fc697+w]/(fs/2);
wn770 = [fc770-w, fc770+w]/(fs/2);
wn1336 = [fc1336-w, fc1336+w]/(fs/2);

b770  = fir1(N, wn770,  rectwin(N+1) );
b697  = fir1(N, wn697,  rectwin(N+1) );
b1336 = fir1(N, wn1336, rectwin(N+1) );

y697sig4 = my_detector(filter(b697,  1, testSig4),500);
y770sig4 = my_detector(filter(b770,  1, testSig4),500);
y1336sig4 = my_detector(filter(b1336,  1, testSig4),500);

figure()
subplot(3,1,1)
plot(y697sig4)
title('697 Hz')

subplot(3,1,2)
plot(y770sig4)
title('770 Hz')

subplot(3,1,3)
plot(y1336sig4)
title('1336 Hz')

% Test the mydetector function using these filters on testSig4 which contains all the tones and adjust
% smoothLength so that the oscillations in the result are less than about 10% of the amplitude. What
% happens to the shape and width of the pulses in the output from mydetector when smoothLength is set to a
% large number?
% When the mydetector function's smoothLengthis set ot a large number, we
% get much less osciallation at our peaks for tones and the shape is much
% more muted with a more constant line at the amplitude.
```



5 button

```
y5 = y1336sig4 .* y770sig4;

figure()
subplot(4,1,1)
plot(testSig4)
title('TestSig4')

subplot(4,1,2)
plot(y770sig4)
title('770 Hz')

subplot(4,1,3)
plot(y1336sig4)
title('1336 Hz')

subplot(4,1,4)
plot(y5)
title('5 Button')

% How well does your detector work?
% It works really well when they are multiplied to the value that we are
% expecting.

% What is the lowest value in the pulse indicating the 5 button?
% The loswest value is about 0.01

% How wide is the pulse?
% The pulse width is about 2500*10^4

% What is the highest value in the output when the 5 button is not pressed?
% Around 0.02
% What are your reasons for selecting the values of N, w, and smoothLength that you used for your detector?
% We chose N because it was a good medium out of all the given N values, w
% seemed appropriate to reuse as was given for the frequency of this lab
% and our smoothLenght because it was a sweetspot to detcect fluctuations
% or oscillations without muting the behaviour of our signal.

% In the design process for this lab, we tried to suppress nearby row and column frequencies by a large
% factor. However, your detector might still work reliably with less attenuation of nearby DTMF
% frequencies. Considering your complete system, the DTMF tone filters followed by the mydetector
% function, followed by your decision method, how much attenuation of nearby DTMF frequencies is
% needed for reliable operation?
% Attenuation is needed to the point where there is a sweet spot of
% resembling the proper signal figure without too much oscillation or too
% little aka muted.
```



Published with MATLAB® R2020a