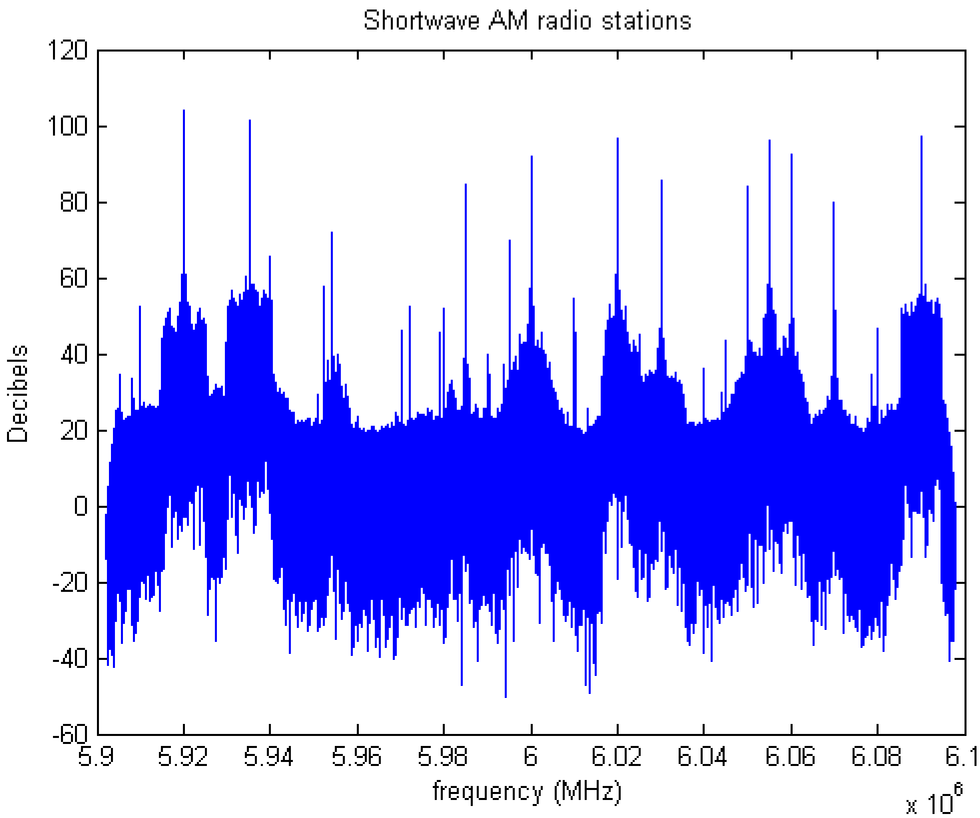
Dan Wortmann - Lab 7 - April 28th, 2014

Table of Contents

3.1

```
clc
clear

load shortwave.mat

%combine the two raw componenets
x = raw(:,1) + 1i*raw(:,2);

%take DFT
X = fftshift(fft(x));

f_LO = 6000000;
N = length(X);

freq = linspace(f_LO - Fs/2,f_LO + Fs/2,N);

figure(1)
plot(freq,20*log10(abs(X)));
xlabel('frequency (MHz)')
ylabel('Decibels')
title('Shortwave AM radio stations')
```



4.1 A simple demodulator

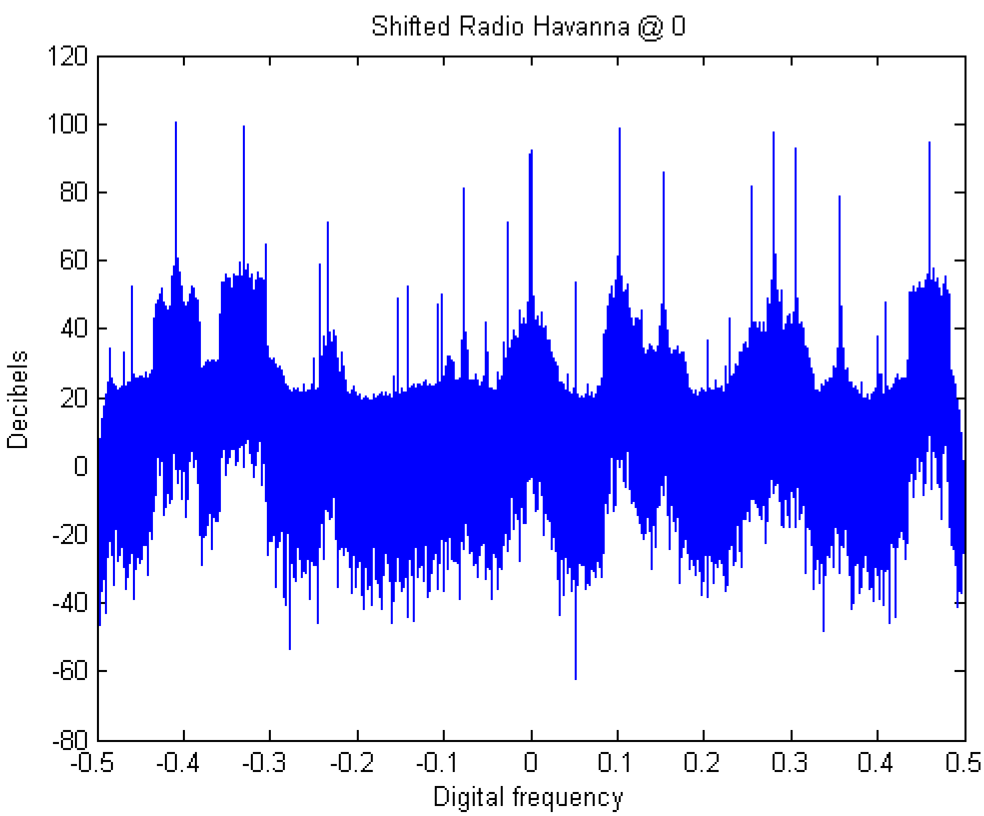
4.2 Finding the carrier frequency

```
%find the value of (f1 - fLO)/Fs

minInd = (N/2)-5000;
maxInd = (N/2)+5000;

[maxVal ,ind0] = max(abs(X(minInd:maxInd)));
f_i = freq(minInd+ind0);
```

4.3 Implementing the demodulator

```
%4.3.1
a_raw = x.*exp(-1i*2*pi*[1:N]'*(f_i - f_LO)/Fs);
Xa = fftshift(fft(a_raw));
Dfreq = linspace(-N/2,N/2-1,N)/N;

figure(2)
plot(Dfreq,20*log10(abs(Xa)));
xlabel('Digital frequency')
ylabel('Decibels')
title('Shifted Radio Havanna @ 0')
```

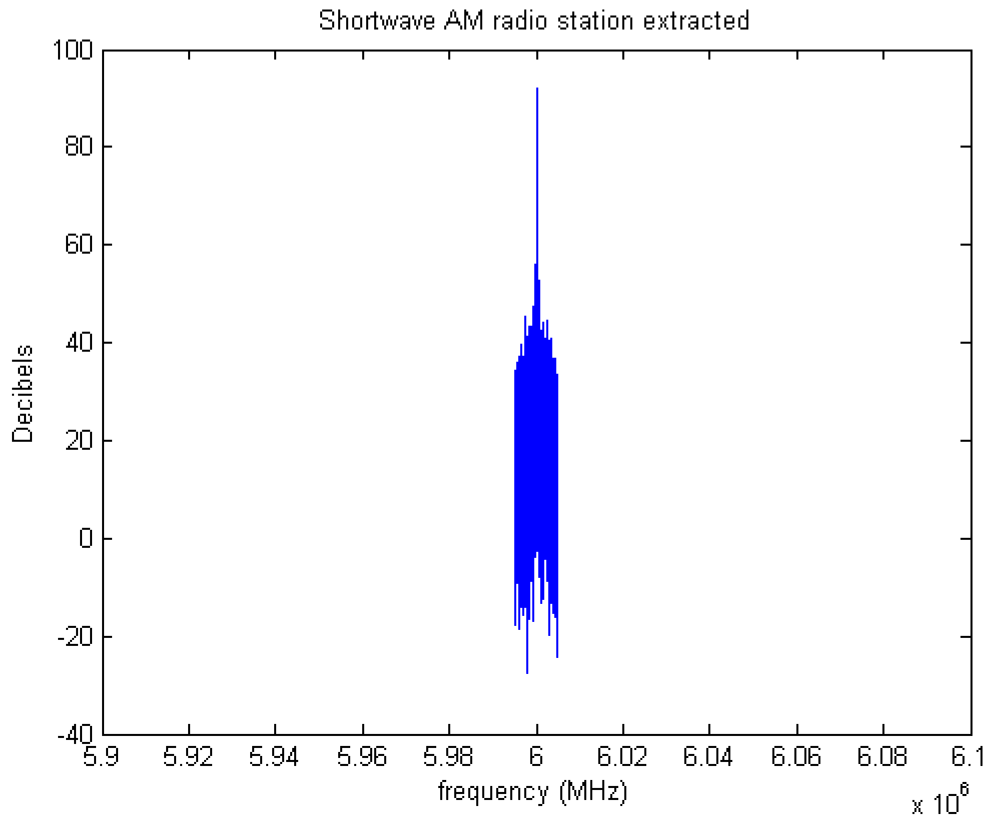
```
%4.3.2
a = real(a_raw);
soundsc(a, Fs);

%The audio seems alright for the quality however there is a distinct noice
%in the background oh a high pitched tone as well as some noticable static
%as well as some volume variation.
```

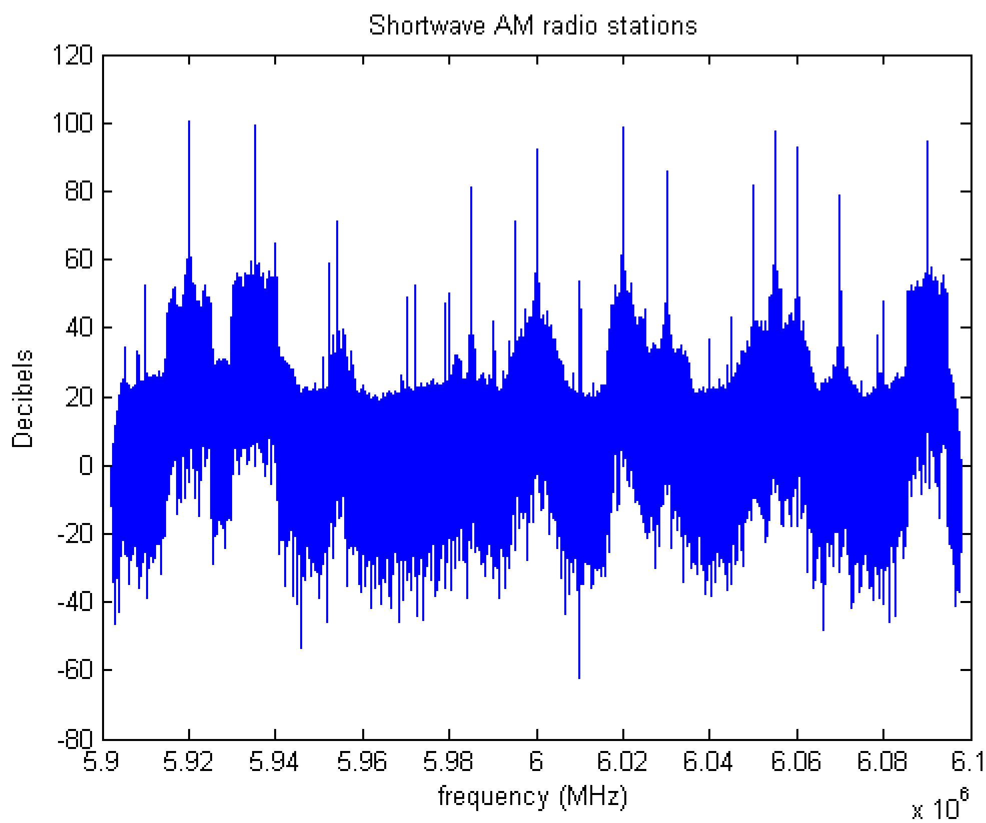


Low Pass Filtering - DFT

```
%Using the DFT
clc
clear
load shortwave.mat
%knowing the approximate width of the station
width = 4900;
f_LO = 6000000;

x = raw(:,1) + 1i*raw(:,2);
%take DFT of original raw signal
X = fftshift(fft(x));
N = length(X);

freq = linspace(f_LO - Fs/2,f_LO + Fs/2,N);
```

```
%get actual carrier frequency
minInd = (N/2)-width;
maxInd = (N/2)+width;

[maxVal ,ind0] = max(abs(X(minInd:maxInd)));
f_i = freq(minInd+ind0);
maxIndex = minInd+ind0;

%get centered signal
a_raw = x.*exp(-1i*2*pi*[1:N]'*(f_i - f_LO)/Fs);
XC = fftshift(fft(a_raw));
freqC = linspace(f_i - Fs/2,f_i + Fs/2,N);

%extract signal within the stations frequencies maxIndex+-width
stationSamples = floor((width/Fs)*N);
tempStation = XC;

%fill in zeros for the remainder of the signal to match original signal
tempStation(1:floor(N/2) - stationSamples) = 0;
tempStation(floor(N/2) + stationSamples:end) = 0;

%take inverse DFT to get a raw station
station_raw = ifft(ifftshift(tempStation));

%play back sound of real portion of station
havanna = real(station_raw);
soundsc(havanna, Fs);

figure(3)
plot(freqC,20*log10(abs(tempStation)));
xlabel('frequency (MHz)')
ylabel('Decibels')
title('Shortwave AM radio station extracted')

figure(4)
plot(freqC,20*log10(abs(XC)));
xlabel('frequency (MHz)')
ylabel('Decibels')
title('Shortwave AM radio stations')
```


Shortwave AM radio station extracted
Decibels
frequency (MHz)
x 10^6
100
80
60
40
20
0
-20
-40
5.9
5.92
5.94
5.96
5.98
6
6.02
6.04
6.06
6.08
6.1


Shortwave AM radio stations
Decibels
frequency (MHz)
x 10^6
120
100
80
60
40
20
0
-20
-40
-60
-80
5.9
5.92
5.94
5.96
5.98
6
6.02
6.04
6.06
6.08
6.1

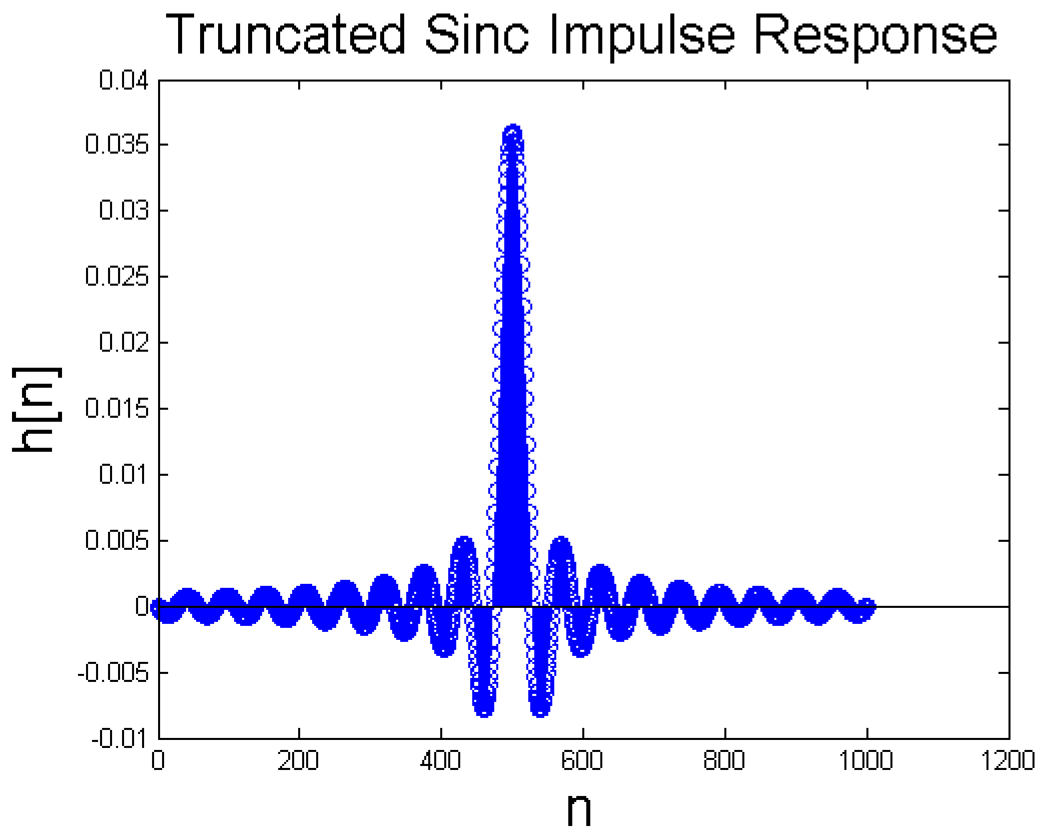
Low Pass Filtering - FIR

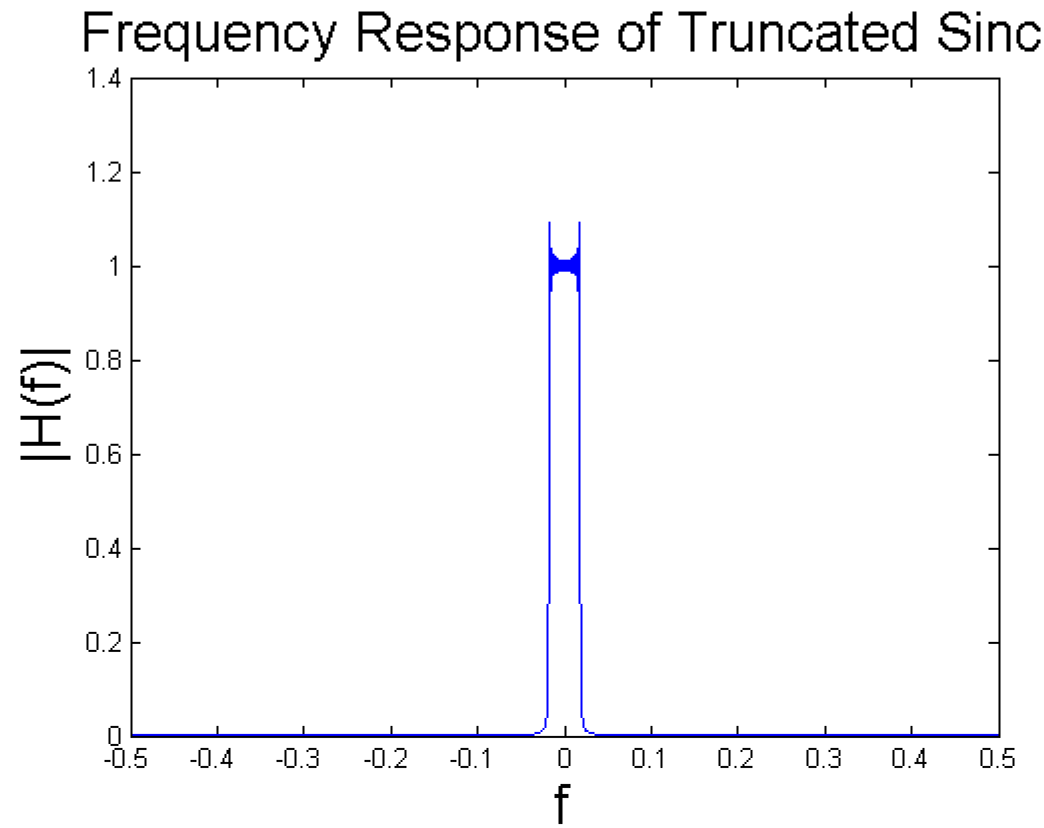
```
%In both filters I assumed the we want to filter out the width of the
%station in respect to the bandwidth of the entire signal. So by taking the
%ration of about 5kHz to Fs/2 I found that I need about 2.5% of the signal
%in each direction, and then make a filter resembling a square wave as
%closely to that as possible.

%Using FIR filters
clc
clear
load shortwave.mat

x = raw(:,1) + 1i*raw(:,2);
N = length(x);
f_i = 6.0001e6;
f_LO = 6000e3;
a_raw = x.*exp(-1i*2*pi*[1:N]'*(f_i - f_LO)/Fs);
width = 4.7e3;
L = 500;
fm = (.75*width/(Fs));
k = [0:1:2*L]+.00001;
fhat = [-.5:.001:.5];
```

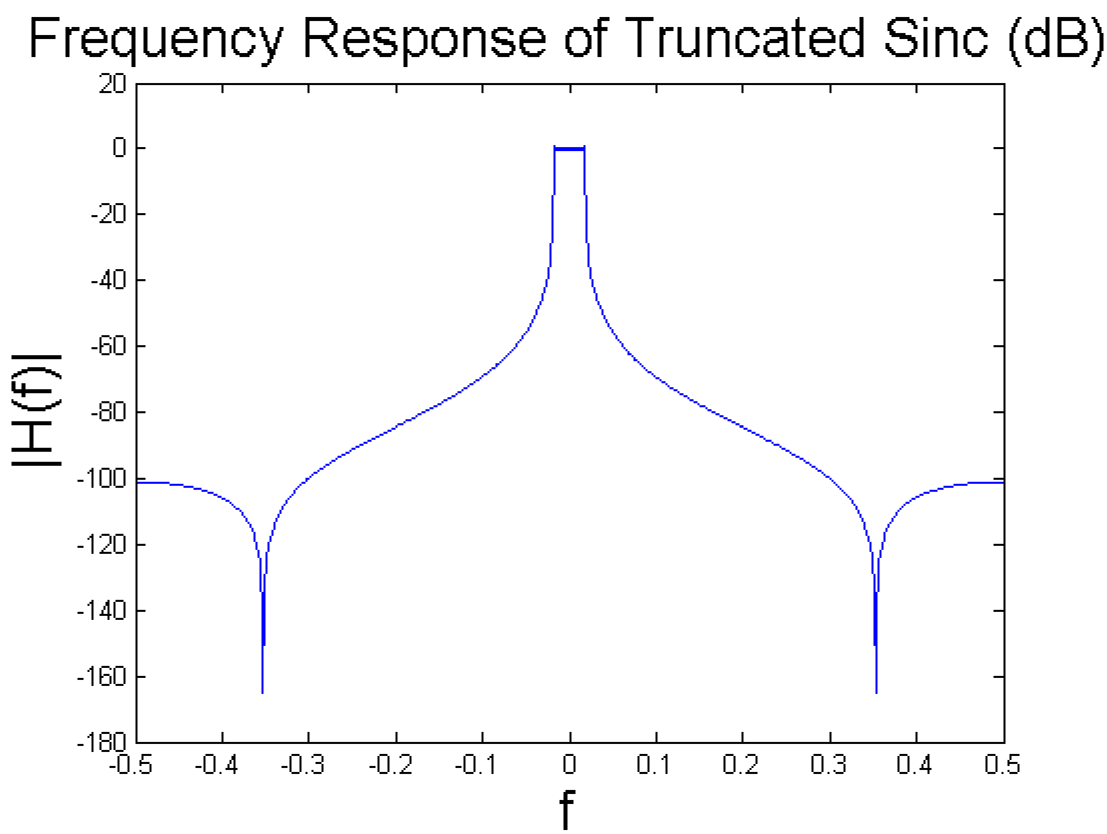
Sinc Filter%%%%%%%

```
h = sin(2*pi*fm*(k-L))./(pi*(k-L));

S_raw = conv(real(a_raw), h);
S_raw = conv(real(S_raw), h);
S_raw = conv(real(S_raw), h);
S_raw = conv(real(S_raw), h);
S_raw = conv(real(S_raw), h);


havanna = real(S_raw);
soundsc(havanna, Fs);

stem(k,h)
title('Truncated Sinc Impulse Response','fontsize',20)
xlabel('n','fontsize',20)
ylabel('h[n]','fontsize',20)

% Compute H(f)
for i =1:length(fhat)
    H(i) = h*exp(-1i*2*pi*fhat(i)*k)';
end

figure
plot(fhat,(abs(H)));
xlabel('f','fontsize',20)
ylabel('|H(f)|','fontsize',20)
```

```
title('Frequency Response of Truncated Sinc','fontsize',20)

figure
plot(fhat,20*log10(abs(H)));
xlabel('f','fontsize',20)
ylabel('|H(f)|','fontsize',20)
title('Frequency Response of Truncated Sinc (dB)','fontsize',20)
```



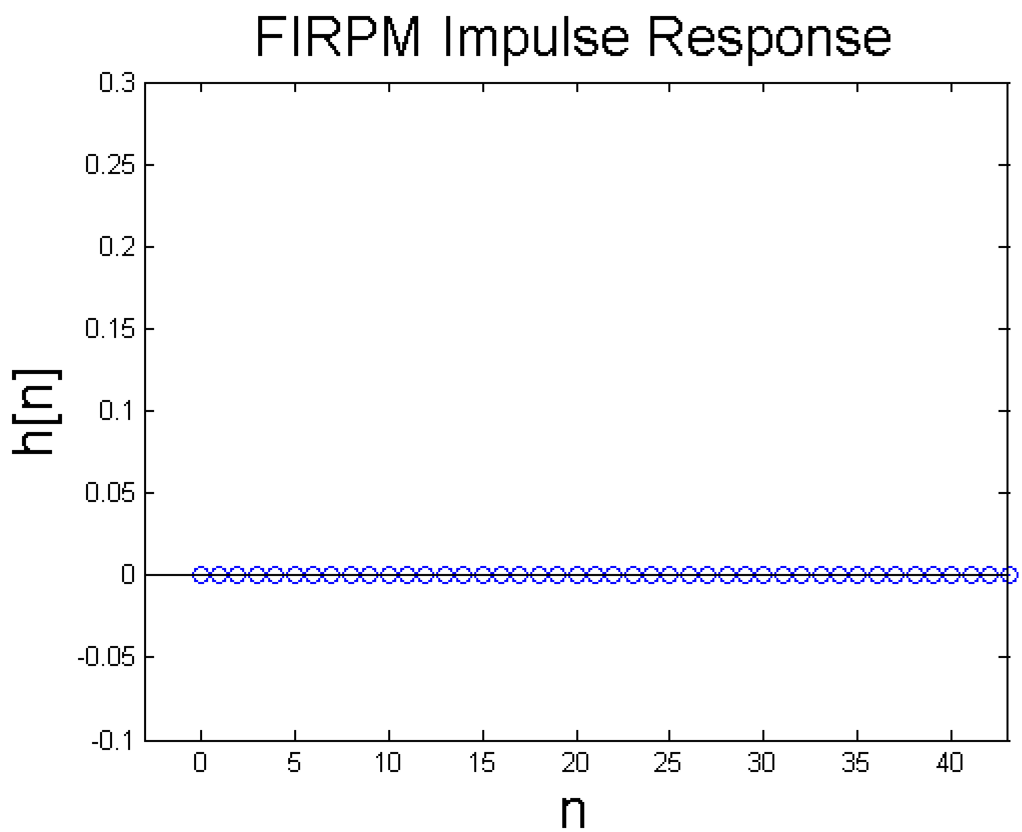

Frequency Response of Truncated Sinc
|H(f)|
f
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

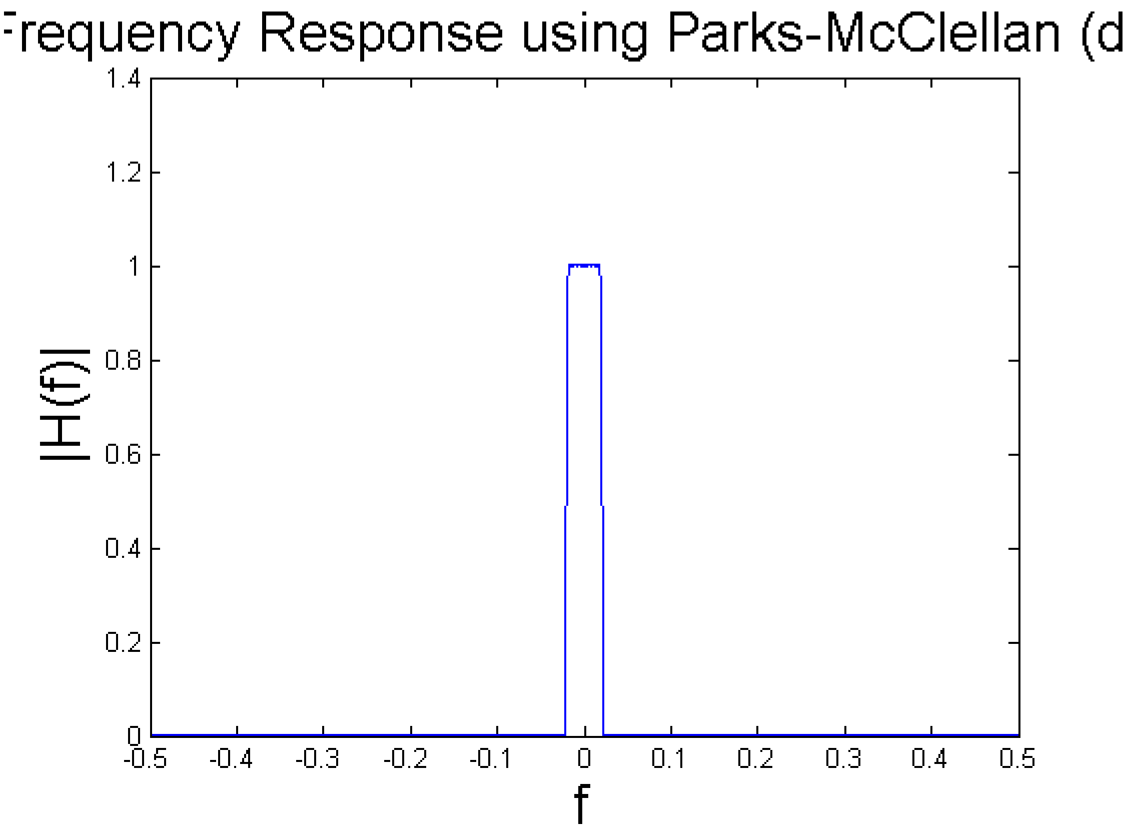

Frequency Response of Truncated Sinc (dB)
|H(f)|
f
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

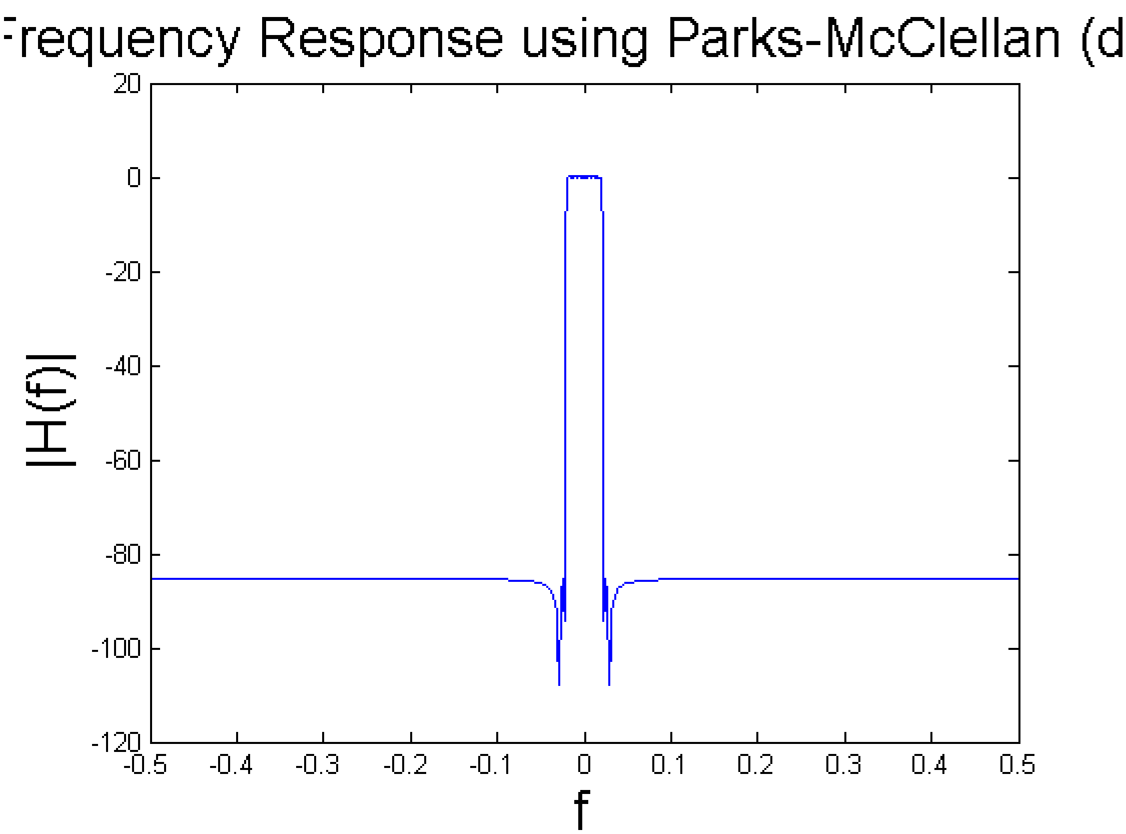
PM Filter%%%%%%%

```
f = [0 fm fm+.005 1/2]*2;
a = [1 1 0 0];
h = firpm(length(k)-1,f,a);

S_raw = conv(real(a_raw), h);
S_raw = conv(real(S_raw), h);
S_raw = conv(real(S_raw), h);
S_raw = conv(real(S_raw), h);

havanna = real(S_raw);
soundsc(havanna, Fs);

% Compute H(f)
for i =1:length(fhat)
    H(i) = h*exp(-1i*2*pi*fhat(i)*k)';
end

figure
stem(k,h)
axis([-3 43 -.1 .3])
title('FIRPM Impulse Response','fontsize',20)
xlabel('n','fontsize',20)
ylabel('h[n]','fontsize',20)

figure
plot(fhat,(abs(H)));
xlabel('f','fontsize',20)
ylabel('|H(f)|','fontsize',20)
title('Frequency Response using Parks-McClellan (dB)','fontsize',20)

figure
plot(fhat,20*log10(abs(H)));
xlabel('f','fontsize',20)
ylabel('|H(f)|','fontsize',20)
title('Frequency Response using Parks-McClellan (dB)','fontsize',20)
axis([-.5 .5 -120 20])

%%4.2
%By changing the cutoff frequency we have a noticeable change in sound
%quality and volume. For instance making the cutoff too small we have a
%very dull sounding response, also much quieter. Also as we increase the
%cutoff frequency we see an increase in stray tones and other interference
%which is much more similar to the raw signal.

%%4.3
%As we change the length of the filter we notice a significant improvement
%in the sound quality as the length increases. This can also be seen in
%parallel with the impulse response of the filters since as we increase the
%length of the filter we get a steeper peak in the middle in comparison to
%the fanning out waves. So in effect as you run the filter and audio
%together we filter out the station of choice much more easily and clearly
```

```
%in comparison to the interference that IS still there with a weak effect.

%%4.4
%The PM filter has a better higher frequency suppression just by looking at
%the Frequency response of both filters side by side. While the sinc filter
%has a noticeable and gradual decline from the middle, the PM, however, has
%an almost square wave characteristic that ensures a minimal amount of
%leakage happens from the higher frequencies.

%%4.5
%As mentioned above the PM filter has the better sound quality as it
%minimizes error and has a closer characteristic to an ideal low-pass
%filter that we could use instead of the analog filters.

%%4.6
%Cascading the filters seems to have minimal if any effect on the audio
%quality. This should be the correct behavior because as it passes the
%first filter-especially the PM- most of the higher frequencies are
%lowered. Doing the same process again has a very minimal effect. There was
%some improvement on the sinc filter after about 10 cascaded filters where
%the audio became less jittery and smoother.
```

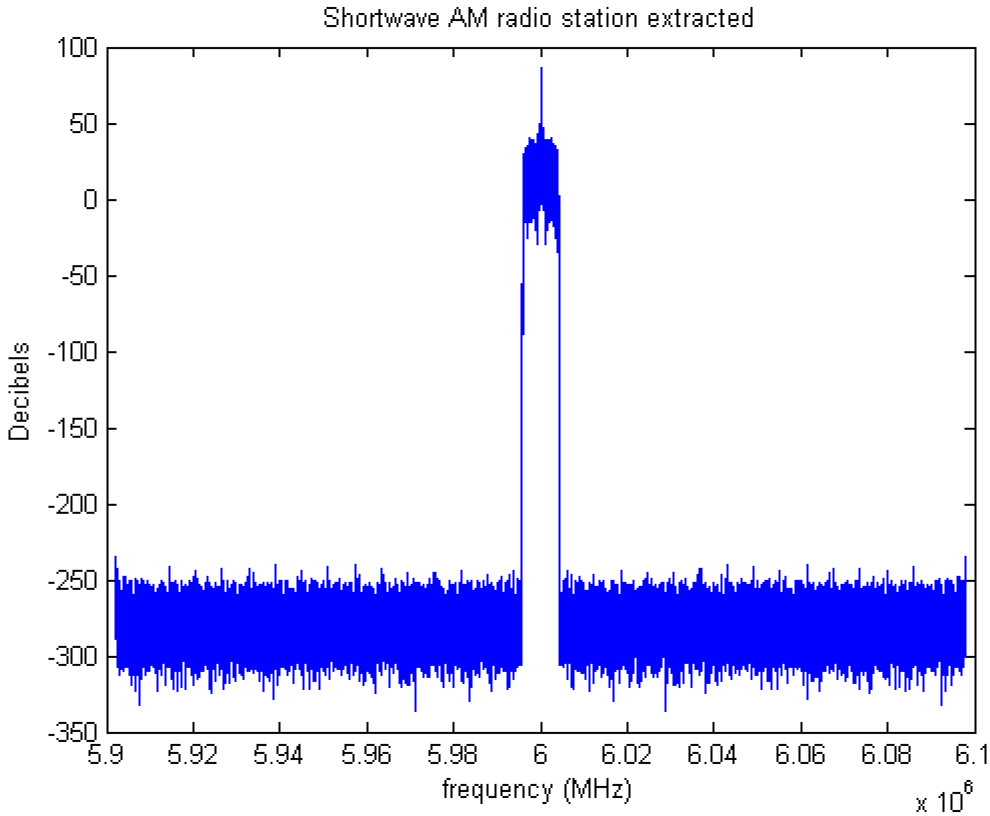


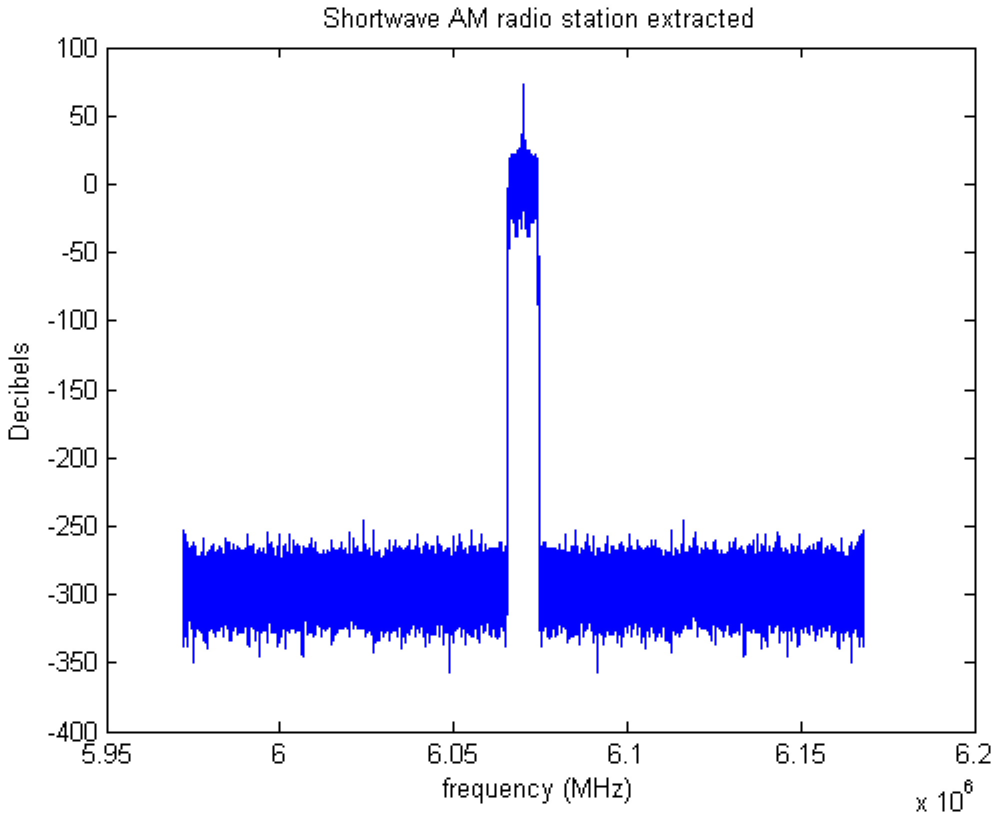

Frequency Response using Parks-McClellan (d
|H(f)|
f
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

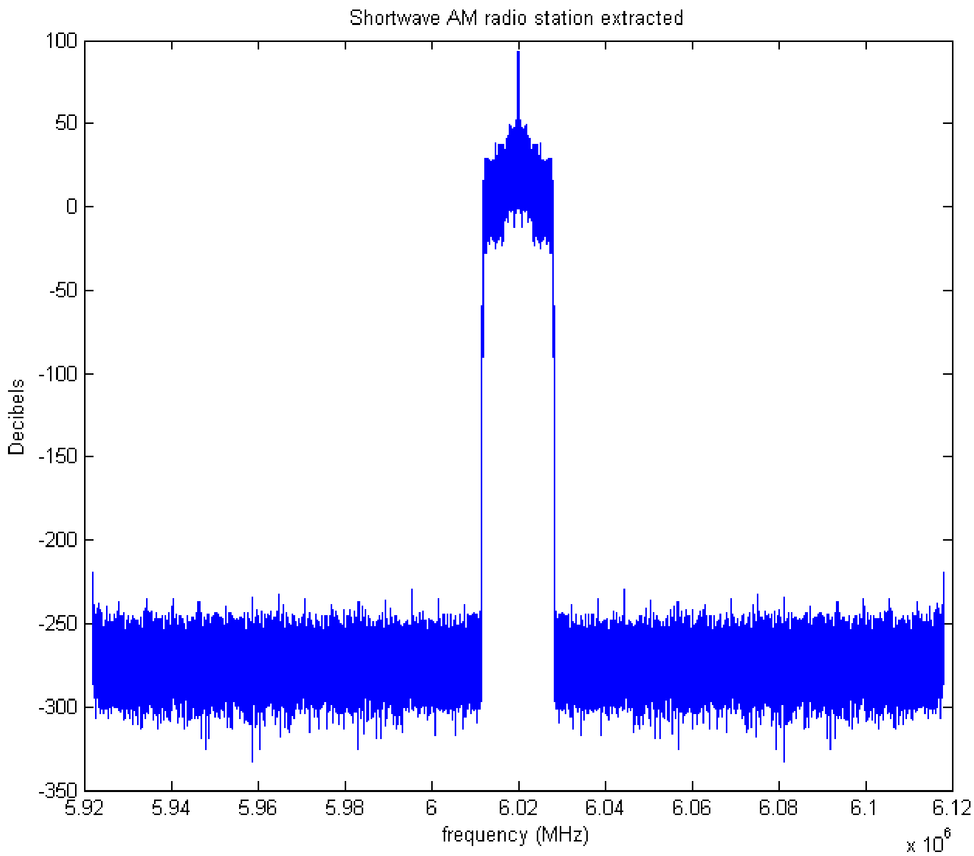

Frequency Response using Parks-McClellan (d
|H(f)|
f
20
0
-20
-40
-60
-80
-100
-120
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

4.5 Listening to the stations

```
%4.7 There are 60 thousand developers in china, he also talks about the
%spice market fluctuation and the amount of bankruptsies caused.
% @f_i of around 6.02 and a station width of 10kHz

% @f_i = 5.985 and width of 4.5e3 kHz I get a spanish station

%4.8 There is a motorhome sale even in Canada and you do not want to miss
%it for your life.
```




Shortwave AM radio station extracted
100
50
0
-50
-100
-150
-200
-250
-300
-350
-400
Decibels
5.95
6
6.05
6.1
6.15
6.2
frequency (MHz)
x 10^6




```
%In order to get the results I used this code with changes to the width and
%f_i of each station as required.
% clc
% clear
% load shortwave.mat
%
% x = raw(:,1) + 1i*raw(:,2);
% N = length(x);
% f_i = 6.07e6; /6.02e6 /6.00e6 +other stations also work
% f_LO = 6000e3;
% a_raw = x.*exp(-1i*2*pi*[1:N]'*(f_i - f_LO)/Fs);
% width = 5e3; /10e3 for british man and slightly different per station
% L = 500;
% fm = (.75*width/(Fs));
% k = [0:1:2*L]+.00001;
% fhat = [-.5:.001:.5];
%
% f = [0 fm fm+.005 1/2]*2;
% a = [1 1 0 0];
% h = firpm(length(k)-1,f,a);
%
% S_raw = conv(real(a_raw), h);
% S_raw = conv(real(S_raw), h);
% S_raw = conv(real(S_raw), h);
```

```
% S_raw = conv(real(S_raw), h);
%
% havanna = real(S_raw);
% soundsc(havanna, Fs);
%
% tempStation = fftshift(fft(S_raw));
% freqC = linspace(f_i - Fs/2,f_i + Fs/2,N+4000);
% figure(3)
% plot(freqC,20*log10(abs(tempStation)));
% xlabel('frequency (MHz)')
% ylabel('Decibels')
% title('Shortwave AM radio station extracted')
```

Published with MATLAB® R2013a