
Table of Contents

.....	1
Introduction	1
Example 1 -- Filtering two sinusoids	1
Example 2 -- Filtering out a voice	8

```
% Matt Kwan
% ES55 -- Final Project
% Due: 12/17/14
```

Introduction

Objective: Create a bandpass filter using Low Pass RC cascaded with High Pass RC, keeping total resistance and capacitance within given range.

Input: Desired frequency, largest/smallest C, largest/smallest R.

Output: A combination of capacitor and resistor values that create the desired filter using the given circuit (see schematic). Examples of the filter in action, including usage of audio.

Plots: Animation of the optimization of the filter. Bode plot of bandpass filter frequency response. Example input signals and output signals with corresponding FFTs.

Numerical Methods: animation, audioread, audioplayer, waitbar, fft, ifft, min, max, real.

Example 1 -- Filtering two sinusoids

This is a simple example to clearly see the change in FFT and signal. In this example, we will create a signal using 250Hz and 975Hz signals and filter out a portion of the 975Hz.

```
clear, clc
disp('Example 1 -- Filtering two sinusoids. Press any key to begin.')
pause;

% USER INPUT
DesiredFreq = 250; % Hertz. Common values: 120Hz - 600kHz
Cmin = 1*10.^-12; % Farads
Cmax = 20*10.^-6; % Farads
Rmin = 10; % Ohms
Rmax = 10*10.^6; % Ohms

% DETERMINING THE FILTER

% Initial values -- proportional to mins/maxes based off of common values
Vin = 1; Clow = (Cmin+Cmax)*.01; Rlow = (Rmin+Rmax)*.01; Chigh = 2*10.^-6;
Rhig = (Rmin+Rmax)*.0008;
% For accuracy, lower fstep. For speed, raise fstep.
```

```

f0lo = 5;
fstep = 10;
if(DesiredFreq >= 1000000)
    f0hi = 1.1*DesiredFreq;
else
    f0hi = 1000000;
end
count2 = 1;
while(1)
    count = 1;
    % GET BODE PLOT DATA
    Av1=zeros(1,ceil((f0hi-f0lo)/fstep)); Av2=Av1; Av3=Av1; % Pre-allocating for s
    for f=f0lo:fstep:f0hi
        [vout, vlow] = Kwan_bandpass(Vin, f, Clow, Rlow, Chigh, Rhigh);
        vout = abs(vout); vlow = abs(vlow);
        Av1(count) = vout/Vin; % (Bandpass Filter)
        Av2(count) = vlow/Vin; % (Low Pass Filter)
        Av3(count) = vout/vlow; % (High Pass Filter)
        count = count+1;
    end

    % GRAB PEAK DATA FOR TERMINAL OUTPUT
    [BandpassGain, fmax] = max(Av1);
    fmax = fmax*fstep; % Convert from index to Hz

    % PLOT BODE PLOT FOR ANIMATION
    if(count2 == 1)
        % Maximized figure automatically for better plot
        aniFigure = figure('units','normalized','outerposition',[0 0 1 1]);
    end
    subplot(1,3,3)
    f=f0lo:fstep:f0hi;
    semilogx(f, Av1, fmax, BandpassGain, 'o', DesiredFreq, ...
        Av1(ceil(DesiredFreq/fstep)), 'o');
    xlabel('Frequency (Hz) (dB)')
    ylabel('Gain')
    title('Semilog Plot of Bandpass Filter')
    legend('Gain', sprintf('Peak Gain Frequency = %f', fmax), ...
        sprintf('Desired Frequency = %f', DesiredFreq), ...
        'Location', 'SouthWest');
    subplot(1,3,2)
    semilogx(f, Av2)
    xlabel('Frequency (Hz) (dB)')
    ylabel('Gain')
    title('Semilog Plot of Low Pass Filter')

    subplot(1,3,1)
    semilogx(f, Av3)
    xlabel('Frequency (Hz) (dB)')
    ylabel('Gain')
    title('Semilog Plot of High Pass Filter')
    M(count2) = getframe(aniFigure);
    count2=count2+1;
end

```

```

    % PREPARE FOR NEXT BODE PLOT
    if(fmax < DesiredFreq)
        Rlow = (1-.01*fstep)*Rlow; % Altering the components to shift
        Clow = (1-.01*fstep)*Clow; % pass band to the right.
        Rhigh = (1-.01*fstep)*Rhigh;
        Chigh = (1-.01*fstep)*Chigh;
        if(Rlow < Rmin || Rhigh > Rmax || Clow < Cmin || Chigh > Cmax)
            disp('Components limits reached.')
            break
        end
    else
        break
    end
end

% PLOT THE BODE PLOT IT STOPPED AT
f=f0lo:fstep:f0hi;
figure;
loglog(f, Av1, fmax, BandpassGain, 'o', DesiredFreq, Av1(ceil(DesiredFreq/fstep))),
xlabel('Frequency (Hz) (dB)')
ylabel('Gain')
title('Bode Plot of Suggested Bandpass Filter')
legend('Gain', sprintf('Peak Gain Frequency = %f', fmax), ...
    sprintf('Desired Frequency = %f', DesiredFreq), ...
    'Location', 'SouthWest');
[BandpassGain, fmax] = max(Av1);
fmax = fmax*fstep; % Convert to Hz

% DISPLAY RESULTS
disp(sprintf('The peak bandpass gain is: %f', BandpassGain))
disp(sprintf('The peak bandpass gain occurs at: %f Hertz', fmax))
disp(sprintf('The desired frequency was: %f Hertz', DesiredFreq))
disp('COMPONENTS LIST:')
disp(sprintf('R.low: %f Ohms', Rlow))
disp(sprintf('R.high: %f Ohms', Rhigh))
disp(sprintf('C.low: %e Farads', Clow))
disp(sprintf('C.high: %e Farads', Chigh))
disp(' ');
disp('Press any key to watch an animation of the optimization process. (Loops 3 ti

pause
movFig = figure('units','normalized','outerposition',[0 0 1 1]); % Maximize
movie(movFig, M,3,9)
disp('Press any key to play, read, and plot an example audio file. Note: Turn on a
pause;

% PREP THE SOUND FILE
Fs = 44100; % Sampling frequency
tmax = 2; % Seconds long
% Create Initial Signals: 250Hz and 975Hz
t = 1/Fs:1/Fs:tmax; f1 = 250; f2 = 975;
s1 = 10*sin(2*pi*f1*t); % Amplitude of 10 -- dominant frequency
s2 = 2*sin(2*pi*f2*t + pi/4); % Amplitude of 2 -- 'noise'

```

```

% CREATE THE SOUND FILE
sig = .027*(s1 + s2); sig=sig'; % Multiplied by .027 because it's loud

% ANALYZE THE SIGNAL
L = size(sig, 1); % Length of signal

NFFT = 2^nextpow2(L); % Next power of 2 from length of L
sigf = fft(sig,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1); % Only care about half the frequency range

figure; % Plotting the signal
subplot(2,1,1);
plot(sig); title('Example Audio Signal: Tone');
xlabel('Time'); xlim([0 250*12]);
subplot(2,1,2);
plot(f,2*abs(sigf(1:NFFT/2+1)),'d-') % Plotting the FFT
title('Frequency Analysis of Signal using FFT')
xlabel('Frequency (Hz)'); xlim([0 1000]);
ylabel('Magnitude of Signal')

[~, indexFFT] = max(2*abs(sigf(1:NFFT/2+1)));
disp(' ');
disp(sprintf('The fundamental frequency is at: %f Hertz', f(indexFFT)));
disp(' ');

pause(.5);

% PLAY THE SIGNAL
player = audioplayer(sig, 44100);
play(player);

disp('Press any key to send the signal through the bandpass filter.')
pause

% SEND THE SIGNAL THROUGH THE BANDPASS FILTER
bpfilter = [f0lo:fstep:f0hi; Av1];
f = Fs*linspace(0,1,NFFT);
soundFile = [f; sigf(:,1)'];
sigf2 = soundFile;
note = 'Initializing waitbar...';
progress = waitbar(0,note);
% Integer > 0. For accuracy, lower stepSize. For speed, raise stepSize.
stepSize = 100;
sizeFile = size(soundFile,2);
for(freqInd=1:stepSize:sizeFile)
    percent = freqInd/size(soundFile,2)*100;
    note = sprintf('%u%% Completed',round(percent));
    waitbar(percent/100,progress,note)
    [closestVal, closestInd] = min(abs(bpfilter(1,:) - soundFile(1,freqInd))); % F
    gain = bpfilter(2, closestInd);
    for(i=1:1:stepSize) % Speeds up progress by using same gain for stepSize number
        if(freqInd+(i-1) <= sizeFile)
            sigf2(2, freqInd+(i-1)) = gain * soundFile(2, freqInd+(i-1)); % Multip
        end
    end
end

```

```

        end
    end
    close(progress)

    pause(.6);

    % PLOT THE FFT AFTER FILTERING
    figure;
    subplot(2,1,2);
    plot(sigf2(1,1:NFFT/2+1), 2*abs(sigf2(2,1:NFFT/2+1)), 'd-'); % Plotting FFT
    title('FFT after Bandpass Filter'); xlabel('Frequency (Hz)');
    ylabel('Magnitude of Signal'); xlim([0 1000]);

    % PLOT THE SIGNAL AFTER FILTERING
    sig2 = ifft(sigf2(2,:))*L.*(1/(BandpassGain));
    subplot(2,1,1);
    plot(real(sig2)); xlim([0 250*12]);
    title('Example Audio Signal after Bandpass Filter'); xlabel('Time');
    pause(.5)
    player = audioplayer(sig2, 44100);
    play(player);
    disp(' ');
    disp('End of Example 1: Tones. Press any key to move on to Example 2: Voices');
    pause;

```

Example 1 -- Filtering two sinusoids. Press any key to begin.

The peak bandpass gain is: 0.444388

The peak bandpass gain occurs at: 260.000000 Hertz

The desired frequency was: 250.000000 Hertz

COMPONENTS LIST:

R.low: 18530.220419 Ohms

R.high: 1482.417633 Ohms

C.low: 3.706041e-08 Farads

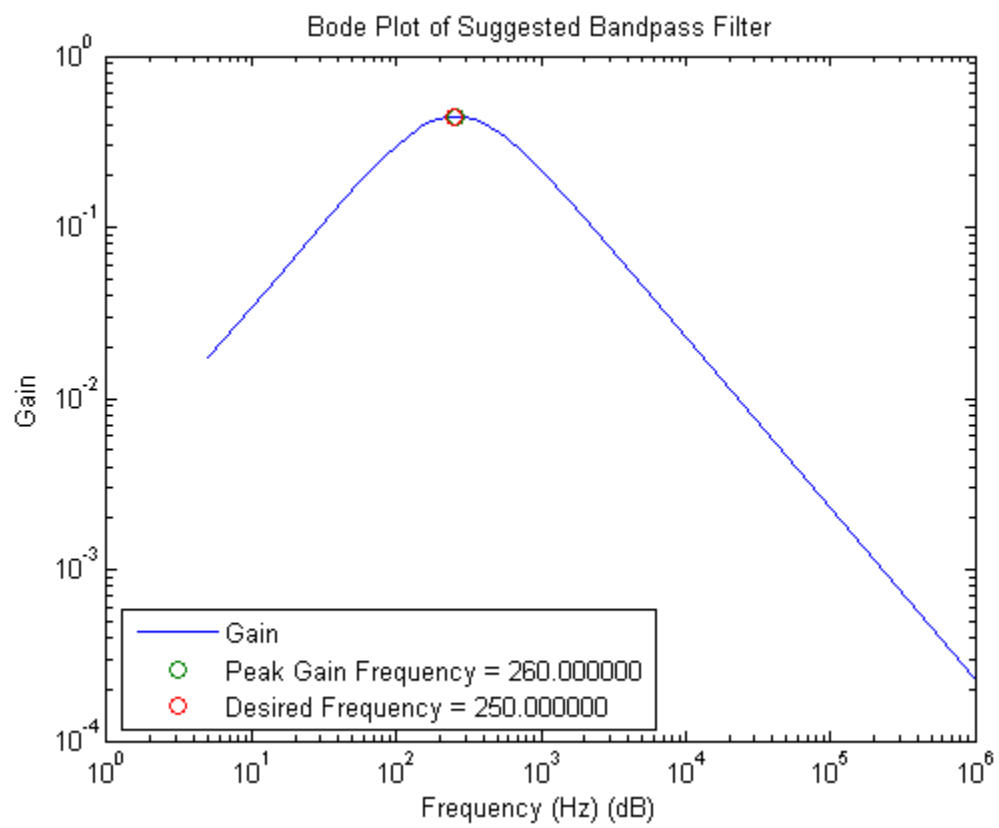
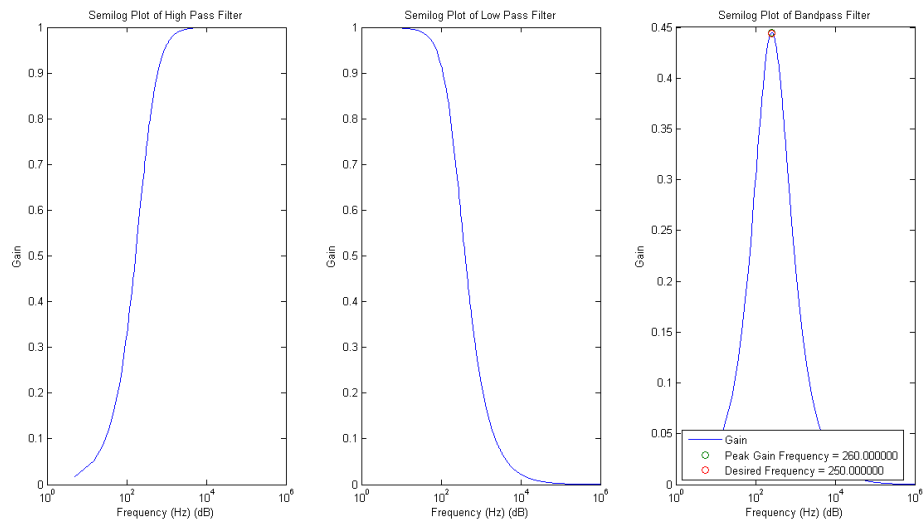
C.high: 3.706040e-07 Farads

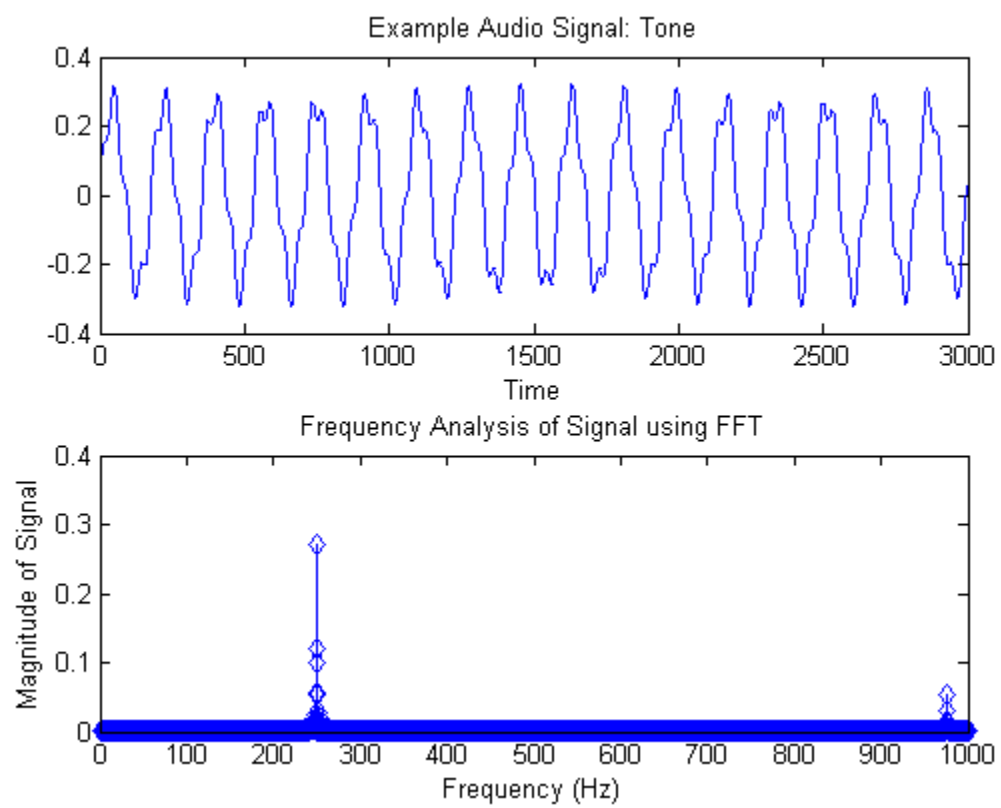
*Press any key to watch an animation of the optimization process. (Loops 3
Press any key to play, read, and plot an example audio file. Note: Turn on*

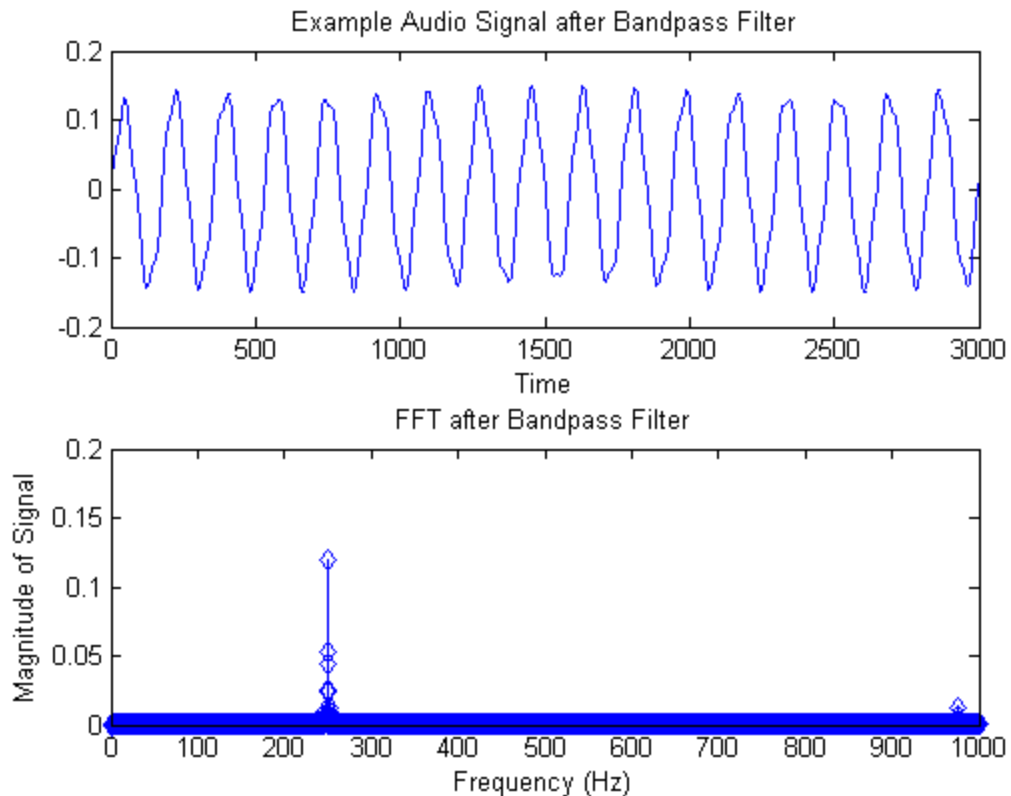
The fundamental frequency is at: 249.987030 Hertz

Press any key to send the signal through the bandpass filter.

End of Example 1: Tones. Press any key to move on to Example 2: Voices







Example 2 -- Filtering out a voice

Voices are between 85 - 400Hz. Putting a filter for only higher frequencies (in this case, 6.1kHz) significantly attenuates the signal. Thus, the voice should be quieter when played after going through a filter whose pass band is outside the range of voices. Interpreting the data can be somewhat difficult, but its audio should make it clear.

```
clear, clc
disp('Example 2 -- Filtering out a voice. Press any key to continue.')
pause;
disp(' ');

% USER INPUT
DesiredFreq = 6100; % Hertz. Common values: 120Hz - 600kHz
Cmin = 1*10.^-12; % Farads
Cmax = 20*10.^-6; % Farads
Rmin = 10; % Ohms
Rmax = 10*10.^6; % Ohms

% DETERMINING THE FILTER

% Initial values -- proportional to mins/maxes based off of common values
Vin = 1; Clow = (Cmin+Cmax)*.01; Rlow = (Rmin+Rmax)*.01; Chigh = 2*10.^-6;
Rhigh = (Rmin+Rmax)*.0008;
% For accuracy, lower fstep. For speed, raise fstep.
f0lo = 5;
```

```

fstep = 10;
if(DesiredFreq >= 1000000)
    f0hi = 1.1*DesiredFreq;
else
    f0hi = 1000000;
end
count2 = 1;
while(1)
    count = 1;
    % GET BODE PLOT DATA
    Av1=zeros(1,ceil((f0hi-f0lo)/fstep)); Av2=Av1; Av3=Av1; % Pre-allocating for s
    for f=f0lo:fstep:f0hi
        [vout, vlow] = Kwan_bandpass(Vin, f, Clow, Rlow, Chigh, Rhigh);
        vout = abs(vout); vlow = abs(vlow);
        Av1(count) = vout/Vin; % (Bandpass Filter)
        Av2(count) = vlow/Vin; % (Low Pass Filter)
        Av3(count) = vout/vlow; % (High Pass Filter)
        count = count+1;
    end

    % GRAB PEAK DATA FOR TERMINAL OUTPUT
    [BandpassGain, fmax] = max(Av1);
    fmax = fmax*fstep; % Convert from index to Hz

    % PLOT BODE PLOT FOR ANIMATION
    if(count2 == 1)
        % Maximized figure automatically for better plot
        aniFigure = figure('units','normalized','outerposition',[0 0 1 1]);
    end
    subplot(1,3,3)
    f=f0lo:fstep:f0hi;
    semilogx(f, Av1, fmax, BandpassGain, 'o', DesiredFreq, ...
        Av1(ceil(DesiredFreq/fstep)), 'o');
    xlabel('Frequency (Hz) (dB)')
    ylabel('Gain')
    title('Semilog Plot of Bandpass Filter')
    legend('Gain', sprintf('Peak Gain Frequency = %f', fmax), ...
        sprintf('Desired Frequency = %f', DesiredFreq), ...
        'Location', 'SouthWest');
    subplot(1,3,2)
    semilogx(f, Av2)
    xlabel('Frequency (Hz) (dB)')
    ylabel('Gain')
    title('Semilog Plot of Low Pass Filter')

    subplot(1,3,1)
    semilogx(f, Av3)
    xlabel('Frequency (Hz) (dB)')
    ylabel('Gain')
    title('Semilog Plot of High Pass Filter')
    M(count2) = getframe(aniFigure);
    count2=count2+1;

    % PREPARE FOR NEXT BODE PLOT

```

```

        if(fmax < DesiredFreq)
            Rlow = (1-.01*fstep)*Rlow; % Altering the components to shift
            Clow = (1-.01*fstep)*Clow; % pass band to the right.
            Rhigh = (1-.01*fstep)*Rhigh;
            Chigh = (1-.01*fstep)*Chigh;
            if(Rlow < Rmin || Rhigh > Rmax || Clow < Cmin || Chigh > Cmax)
                disp('Components limits reached.')
                break
            end
        else
            break
        end
    end
end

% PLOT THE BODE PLOT IT STOPPED AT
f=f0lo:fstep:f0hi;
figure;
loglog(f, Av1, fmax, BandpassGain, 'o', DesiredFreq, Av1(ceil(DesiredFreq/fstep)),
xlabel('Frequency (Hz) (dB)')
ylabel('Gain')
title('Bode Plot of Suggested Bandpass Filter')
legend('Gain', sprintf('Peak Gain Frequency = %f', fmax), ...
    sprintf('Desired Frequency = %f', DesiredFreq), ...
    'Location', 'SouthWest');
[BandpassGain, fmax] = max(Av1);
fmax = fmax*fstep; % Convert to Hz

% DISPLAY RESULTS
disp(sprintf('The peak bandpass gain is: %f', BandpassGain))
disp(sprintf('The peak bandpass gain occurs at: %f Hertz', fmax))
disp(sprintf('The desired frequency was: %f Hertz', DesiredFreq))
disp('COMPONENTS LIST:')
disp(sprintf('R.low: %f Ohms', Rlow))
disp(sprintf('R.high: %f Ohms', Rhigh))
disp(sprintf('C.low: %e Farads', Clow))
disp(sprintf('C.high: %e Farads', Chigh))
disp(' ');
disp('Press any key to watch an animation of the optimization process. (Loops 3 ti

pause
movFig = figure('units','normalized','outerposition',[0 0 1 1]); % Maximize
movie(movFig, M,3,9)
disp('Press any key to play, read, and plot an example audio file. Note: Turn on a
pause;
% READ IN THE SOUND FILE
sig = audioread('Kwan_Hello.wav');

% ANALYZE THE SIGNAL
Fs = 44100; % Sampling frequency
T = 1/Fs; % Sample time
L = size(sig, 1); % Length of signal
t = (0:L-1)*T; % Time vector

NFFT = 2^nextpow2(L); % Next power of 2 from length of L

```

```

sigf = fft(sig,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1); % Only care about half the frequency range

figure; % Plotting the signal
subplot(2,1,1);
plot(sig); title('Example Audio Signal "Hello, my name is Matt"');
xlabel('Time');
subplot(2,1,2);
plot(f,2*abs(sigf(1:NFFT/2+1))) % Plotting the FFT
title('Frequency Analysis of Signal using FFT')
xlabel('Frequency (Hz)'); xlim([0 25000]);
ylabel('Magnitude of Signal')

[maxFFT, indexFFT] = max(2*abs(sigf(1:NFFT/2+1)));
disp(' ');
disp(sprintf('The fundamental frequency is at: %f Hertz', f(indexFFT)));
disp(' ');

pause(.5);

% PLAY THE SIGNAL
player = audioplayer(sig, 44100);
play(player);

disp('Press any key to send the signal through the bandpass filter.')
pause

% SEND THE SIGNAL THROUGH THE BANDPASS FILTER
bpfilter = [f0lo:fstep:f0hi; Av1];
f = Fs*linspace(0,1,NFFT);
soundFile = [f; sigf(:,1)'];
sigf2 = soundFile;
note = 'Initializing waitbar...';
progress = waitbar(0,note);
% Integer > 0. For accuracy, lower stepSize. For speed, raise stepSize.
stepSize = 100;
sizeFile = size(soundFile,2);
for(freqInd=1:stepSize:sizeFile)
    percent = freqInd/size(soundFile,2)*100;
    note = sprintf('%u%% Completed',round(percent));
    waitbar(percent/100,progress,note)
    [closestVal, closestInd] = min(abs(bpfilter(1,:) - soundFile(1,freqInd))); % F
    gain = bpfilter(2, closestInd);
    for(i=1:1:stepSize) % Speeds up progress by using same gain for stepSize numbe
        if(freqInd+(i-1) <= sizeFile)
            sigf2(2, freqInd+(i-1)) = gain * soundFile(2, freqInd+(i-1)); % Multip
        end
    end
end
close(progress)
pause(.6);

% PLOT THE SIGNAL & FFT AFTER FILTERING
figure;

```

```

subplot(2,1,2);
plot(sigf2(1,1:NFFT/2+1), 2*abs(sigf2(2,1:NFFT/2+1)));
title('FFT after Bandpass Filter'); xlabel('Frequency (Hz)');
ylabel('Magnitude of Signal'); xlim([0 25000]);

% PLAY THE SIGNAL AFTER FILTERING
sig2 = ifft([(sigf2(2,:))' (sigf2(2,:))'])*L.*(1/(BandpassGain));
subplot(2,1,1);
plot(real(sig2));
title('Example Audio Signal after Bandpass Filter'); xlabel('Time');
pause(.5)
player = audioplayer(sig2, 44100);
play(player);
disp(' ');
disp('End of Program.');
```

Example 2 -- Filtering out a voice. Press any key to continue.

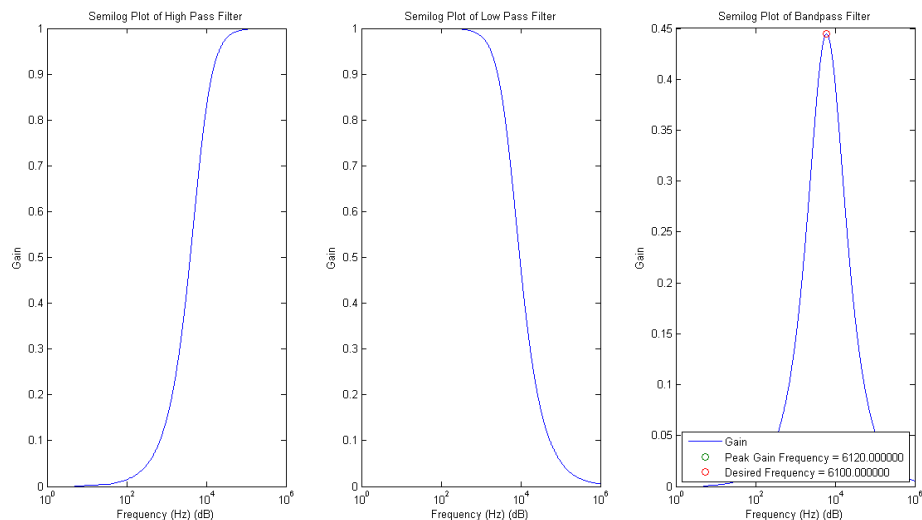
*The peak bandpass gain is: 0.444444
The peak bandpass gain occurs at: 6120.000000 Hertz
The desired frequency was: 6100.000000 Hertz
COMPONENTS LIST:
R.low: 3815.208060 Ohms
R.high: 305.216645 Ohms
C.low: 7.630409e-09 Farads
C.high: 7.630408e-08 Farads*

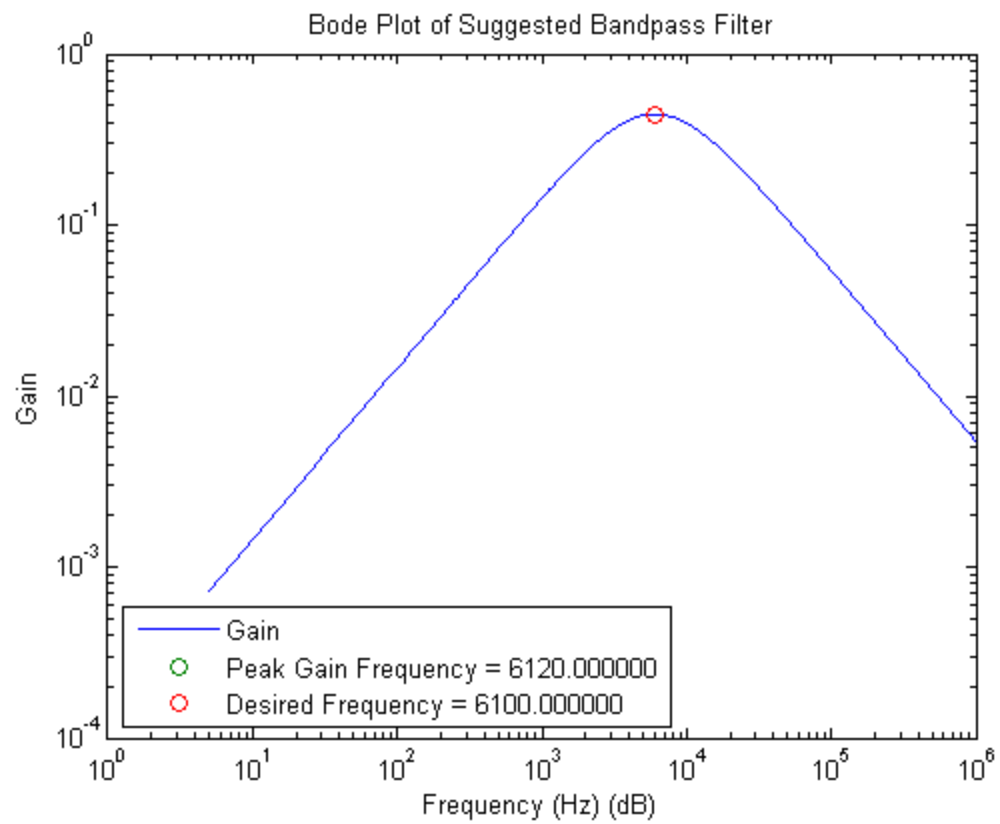
*Press any key to watch an animation of the optimization process. (Loops 3
Press any key to play, read, and plot an example audio file. Note: Turn on*

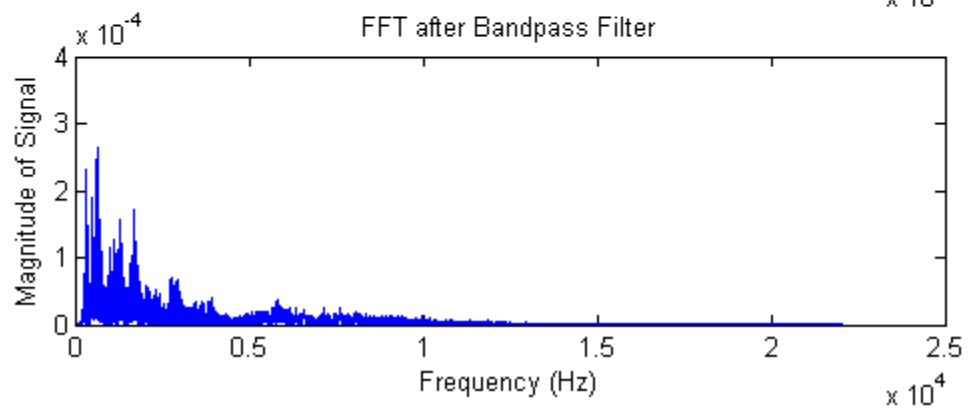
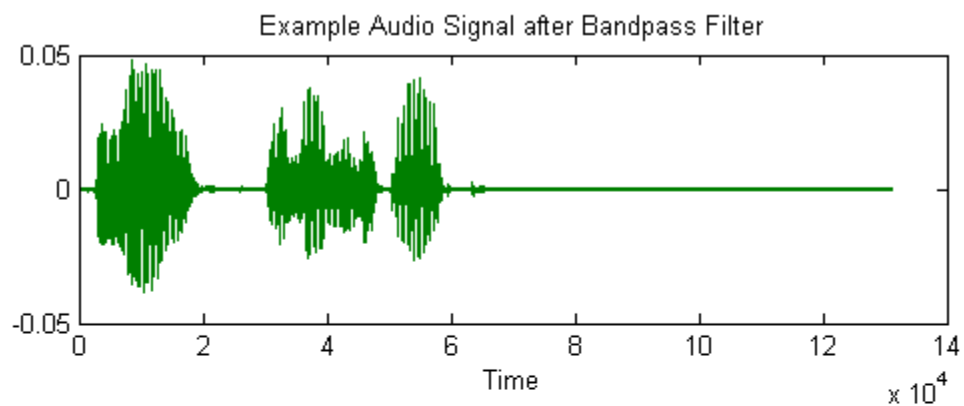
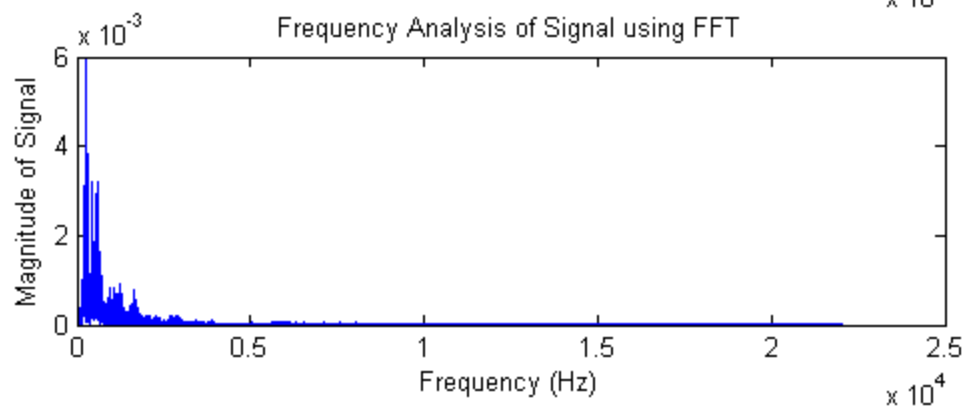
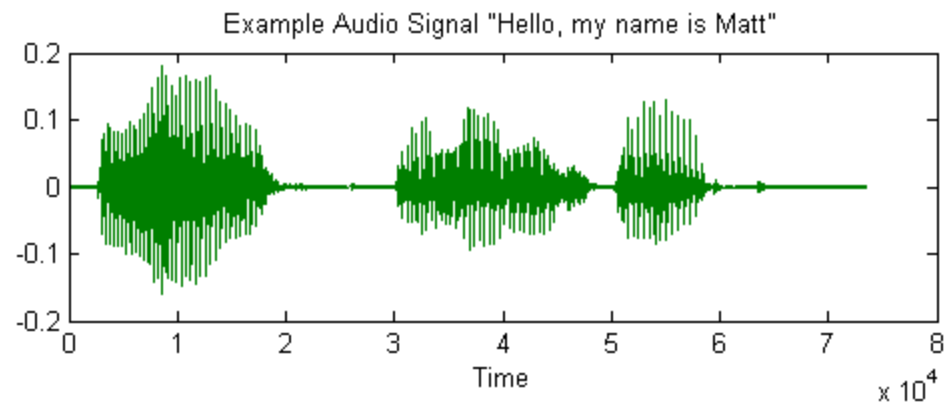
The fundamental frequency is at: 286.660767 Hertz

Press any key to send the signal through the bandpass filter.

End of Program.







Published with MATLAB® R2014a