Audio Equalizer

DSP Final Project

Youssef Amr ElKady 7651 Group 2 Section 1

Ahmed Hany Ibrahim 7905 Group 1 Section 2

Code:

```
classdef Audio_Equalizer < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                    matlab.ui.Figure
        Panel                       matlab.ui.container.Panel
        TimeDomainPanel             matlab.ui.container.Panel
        AfterLabel_3                matlab.ui.control.Label
        BeforeLabel_3               matlab.ui.control.Label
        UIAxes2                     matlab.ui.control.UIAxes
        UIAxes                      matlab.ui.control.UIAxes
        gain_20k                    matlab.ui.control.Slider
        KHZSlider_4Label            matlab.ui.control.Label
        gain_14k                    matlab.ui.control.Slider
        KHZSlider_3Label            matlab.ui.control.Label
        gain_12k                    matlab.ui.control.Slider
        KHZSlider_2Label            matlab.ui.control.Label
        gain_6k                     matlab.ui.control.Slider
        KHZSliderLabel              matlab.ui.control.Label
        gain_3k                     matlab.ui.control.Slider
        HZSlider_5Label             matlab.ui.control.Label
        gain_1005                   matlab.ui.control.Slider
        HZSlider_4Label             matlab.ui.control.Label
        gain_610                    matlab.ui.control.Slider
        HZSlider_3Label             matlab.ui.control.Label
        gain_300                    matlab.ui.control.Slider
        HZSlider_2Label             matlab.ui.control.Label
        gain_170                    matlab.ui.control.Slider
        HZSliderLabel               matlab.ui.control.Label
        FreuquencyDomainPanel       matlab.ui.container.Panel
        BeforeLabel_2               matlab.ui.control.Label
        AfterLabel                  matlab.ui.control.Label
        PhaseLabel                  matlab.ui.control.Label
        MagnitudeLabel              matlab.ui.control.Label
        UIAxes5                     matlab.ui.control.UIAxes
        UIAxes6                     matlab.ui.control.UIAxes
        UIAxes4                     matlab.ui.control.UIAxes
        UIAxes3                     matlab.ui.control.UIAxes
        FrequencyBandGainsPanel     matlab.ui.container.Panel
        SoundhalfnewFSButton        matlab.ui.control.Button
        SounddoublenewFSButton      matlab.ui.control.Button
        ApplynewfiltersButton       matlab.ui.control.Button
        ResetButton                 matlab.ui.control.Button
        PlotButton                  matlab.ui.control.Button
        FiltertoPlotDropDown        matlab.ui.control.DropDown
        FiltertoPlotDropDownLabel   matlab.ui.control.Label
        samplingrateEditField       matlab.ui.control.EditField
        samplingrateEditFieldLabel  matlab.ui.control.Label
        SaveButton                  matlab.ui.control.Button
        StartButton                 matlab.ui.control.Button
        Filter_typeButtonGroup      matlab.ui.container.ButtonGroup
        IIRButton                   matlab.ui.control.RadioButton
        FIRButton                   matlab.ui.control.RadioButton
        locationEditField           matlab.ui.control.EditField
        locationEditFieldLabel      matlab.ui.control.Label
        browseButton                matlab.ui.control.Button
    end


    properties (Access = private)
        y % resampled wave
        fir_order = 40
        iir_order = 4
        fs % sampling frequency
        t % time
        fm % frequency/2
        fo % sampling rate output
        Ns % number of samples
        bandgains = ones(1,9); % gains of each band
        newfs % inputted sampling rate
        new_signal % Description
    end


    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: browseButton
        function browseButtonPushed(app, event)
           [FileName,FilePath]=uigetfile({'*.wav'});
           fullPath = [FilePath FileName];
           app.locationEditField.Value = fullPath;
           [app.y,app.fs] = audioread(app.locationEditField.Value);
           %disp(app.fs);
           app.y = app.y(:,1);
           app.y = transpose(app.y);
           app.Ns = length(app.y);
           app.t = linspace(0, app.Ns/app.fs, app.Ns);
           app.fm = app.fs/2;

        end

        % Button pushed function: StartButton
        function StartButtonPushed(app, event)

            gains = [app.gain_170.Value app.gain_300.Value app.gain_610.Value
app.gain_1005.Value app.gain_3k.Value app.gain_6k.Value app.gain_12k.Value
app.gain_14k.Value app.gain_20k.Value];
            app.bandgains = db2mag(gains);

            freq = [0,170,300,610,1005,3000,6000,12000,14000,20000];
            x =  str2double(app.FiltertoPlotDropDown.Value);


            if app.Filter_typeButtonGroup.SelectedObject == app.IIRButton
```

```
                if x == 1
                    [b, a] = butter(app.iir_order,170/(app.fs/2), 'low');
                else

                    [b, a] = butter(app.iir_order,[freq(x) 
freq(x+1)]/(app.fs/2),'bandpass');
                end
                type = ['IIR'];

            else
                if x == 1
                    b = fir1(app.fir_order,170/(app.fs/2),'low');
                else
                    b = fir1(app.fir_order,[freq(x) 
freq(x+1)]/(app.fs/2),'bandpass');
                end

                a = 1;
                type = ['FIR'];

            end


            title1 = ['Gain and Phase response of ', num2str(freq(x)), ' - ' , 
num2str(freq(x+1)), ' Hz filter'];
            title2 = ['Impulse response of ', num2str(freq(x)), ' - ' , 
num2str(freq(x+1)), ' Hz filter'];
            title3 = ['Step response of ', num2str(freq(x)), ' - ' , 
num2str(freq(x+1)), ' Hz filter'];
            title4 = ['Zeros and Poles of ', num2str(freq(x)), ' - ' , 
num2str(freq(x+1)), ' Hz filter'];
            title5 = ['Time Domain signal with (', num2str(freq(x)), ' - ' , 
num2str(freq(x+1)), ' Hz) ', type, ' filter'];
            title6 = ['Magnitude of filtered signal in frequency domain'];
            title7 = ['Phase of filtered signal in frequency domain'];

            figure;
            freqz(b, a);
            title(title1);

            figure;
            subplot(2,2,1);
            impz(b,a);
            title(title2);
            subplot(2,2,2);
            stepz(b,a);
            title(title3);


            [z,p, k]  = tf2zpk(b,a);
            subplot(2,2,[3,4]);
            zplane(z,p);
            title(title4);
```

```matlab
    filteredSignal = app.bandgains(x) * filter(b,a,app.y);
    figure;
    subplot(3,1,1)
    plot(app.t,filteredSignal);
    title(title5);
    xlabel('Time in seconds');
    ylabel('Amplitude');


    subplot(3,1,2);
    fmag = abs(fftshift(fft(filteredSignal))/app.fs);
    f_xaxis = linspace(-app.fs/2,app.fs/2,app.Ns);
    plot(f_xaxis,fmag); %filtered signal in frequency domain
    title(title6);
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');


    phase = angle(fftshift(fft(filteredSignal)));
    subplot(3,1,3)
    plot(f_xaxis,phase);
    title(title7);
    xlabel('Frequency (Hz)');
    ylabel('Phase');


    %fvtool(b, a);


end

% Callback function
function samplingrateEditFieldValueChanged(app, event)


end

% Button pushed function: ResetButton
function ResetButtonPushed(app, event)
    app.bandgains = ones(1,9);
    app.gain_170.Value = 0;
    app.gain_300.Value = 0;
    app.gain_610.Value = 0;
    app.gain_1005.Value = 0;
    app.gain_3k.Value = 0;
    app.gain_6k.Value = 0;
    app.gain_12k.Value = 0;
    app.gain_14k.Value = 0;
    app.gain_20k.Value = 0;
end

% Button pushed function: PlotButton
function PlotButtonPushed(app, event)
```

```
            %Time Domain
            plot(app.UIAxes,app.t,app.y) % elly taht 3la el shemal
            t_new =
linspace(0,length(app.new_signal)/app.newfs,length(app.new_signal));
            plot(app.UIAxes2,t_new,app.new_signal); % elly fo2 3la el yemin

            %Frequency Domain
            L1 = length(app.y);
            L2 = length(app.new_signal);
            f_xaxis1 = linspace(-app.fs/2,app.fs/2,L1);
            f_xaxis2 = linspace(-app.newfs/2,app.newfs/2,L2);

            %Frequency Magnitude
            fmag1 = abs(fftshift(fft(app.y))/app.fs); %we divide by L1 to normalize
            plot(app.UIAxes4,f_xaxis1,fmag1); % 3la el yemin taht khales

             % we divide by L2 to normalize
            fmag2 = abs(fftshift(fft(app.new_signal))/app.newfs);
            plot(app.UIAxes3,f_xaxis2,fmag2); % 3la el shemal taht khales


            %Frequency Phase
            fphase1 = angle(fftshift(fft(app.y)));
            plot(app.UIAxes6,f_xaxis1,fphase1); % 3la el yemin taht khales
            fphase2 = angle(fftshift(fft(app.new_signal)));
            plot(app.UIAxes5,f_xaxis2,fphase2); % 3la el shemal taht khales

        end

        % Button pushed function: ApplynewfiltersButton
        function ApplynewfiltersButtonPushed(app, event)
            gains = [app.gain_170.Value app.gain_300.Value app.gain_610.Value
app.gain_1005.Value app.gain_3k.Value app.gain_6k.Value app.gain_12k.Value
app.gain_14k.Value app.gain_20k.Value];
            app.bandgains = db2mag(gains);
            app.newfs = app.samplingrateEditField.Value();
            freq = [0,170,300,610,1005,3000,6000,12000,14000,20000];
            app.new_signal =zeros(1,length(app.y));

            if app.Filter_typeButtonGroup.SelectedObject == app.FIRButton
                for x = 1:9
                    if x==1
                        b = fir1(app.fir_order,170/app.fm);
                    else
                        b = fir1(app.fir_order,[freq(x)
freq(x+1)]/app.fm,'bandpass');
                    end
                    filteredSignal = filter(b,1,app.y); %filtered signal in time
domain
                    app.new_signal = app.new_signal +
(filteredSignal*app.bandgains(x));
                end
            else

                for x = 1:9
```

```
                     if x==1
                         [b,a] = butter(app.iir_order,170/app.fm);
                     else
                         [b,a] = butter(app.iir_order,[freq(x)
freq(x+1)]/app.fm,'bandpass');
                     end

                    filteredSignal = filter(b,a,app.y);
                    app.new_signal = app.new_signal +
(filteredSignal*app.bandgains(x));
                end
            end
            if strcmpi(app.newfs,'Enter new Sampling rate') == 1
                app.newfs = app.fs;
            else
                app.newfs = str2double(app.newfs);
            end
            %app.new_signal = resample(app.new_signal,app.newfs,app.fs);

        end

        % Button pushed function: SaveButton
        function SaveButtonPushed(app, event)
            audiowrite("output_file_half.wav",app.new_signal, app.newfs/2);
            audiowrite("output_file_double.wav",app.new_signal, app.newfs*2);


        end

        % Button pushed function: SounddoublenewFSButton
        function SounddoublenewFSButtonPushed(app, event)
            sound(app.new_signal, app.newfs*2);
            half_t = linspace(0,length(app.y)/(app.newfs*2),length(app.y));
            figure;
            subplot(2,1,1);
            plot(half_t,app.y);
            title("Output signal for double inputted FS");
            %Frequency Domain
            L1 = length(app.y);
            f_xaxis1 = linspace(-app.newfs,app.newfs,L1);
            %Frequency Magnitude
            fmag1 = abs(fftshift(fft(app.y))/app.newfs*2); %we divide by L1 to
normalize
            subplot(2,1,2);
            plot(f_xaxis1,fmag1);
            title("magnitude of Double fs");
        end

        % Button pushed function: SoundhalfnewFSButton
        function SoundhalfnewFSButtonPushed(app, event)
            sound(app.new_signal, app.newfs/2);
            Double_t = linspace(0,length(app.y)/(app.newfs/2),length(app.y));
            figure;
            subplot(2,1,1);
            plot(Double_t,app.y);
            title("Output signal for half inputted FS");
            %Frequency Domain
            L1 = length(app.y);
            f_xaxis1 = linspace(-app.newfs/4,app.newfs/4,L1);

            %Frequency Magnitude
            fmag1 = abs(fftshift(fft(app.y))/app.newfs/2); %we divide by L1 to
normalize
            subplot(2,1,2);
            plot(f_xaxis1,fmag1);
            title("magnitude of half fs");

        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Color = [0.902 0.902 0.902];
            app.UIFigure.Position = [100 100 1103 826];
            app.UIFigure.Name = 'MATLAB App';

            % Create browseButton
            app.browseButton = uibutton(app.UIFigure, 'push');
            app.browseButton.ButtonPushedFcn = createCallbackFcn(app,
@browseButtonPushed, true);
            app.browseButton.Position = [290 787 100 22];
            app.browseButton.Text = 'browse';

            % Create locationEditFieldLabel
            app.locationEditFieldLabel = uilabel(app.UIFigure);
            app.locationEditFieldLabel.HorizontalAlignment = 'right';
            app.locationEditFieldLabel.Position = [18 787 47 22];
            app.locationEditFieldLabel.Text = 'location';

            % Create locationEditField
            app.locationEditField = uieditfield(app.UIFigure, 'text');
            app.locationEditField.Position = [80 787 183 22];

            % Create Filter_typeButtonGroup
            app.Filter_typeButtonGroup = uibuttongroup(app.UIFigure);
            app.Filter_typeButtonGroup.Title = 'Filter_type';
            app.Filter_typeButtonGroup.Position = [657 738 123 82];

            % Create FIRButton
            app.FIRButton = uiradiobutton(app.Filter_typeButtonGroup);
            app.FIRButton.Text = 'FIR';
            app.FIRButton.Position = [11 36 58 22];
            app.FIRButton.Value = true;
            % Create IIRButton
            app.IIRButton = uiradiobutton(app.Filter_typeButtonGroup);
            app.IIRButton.Text = 'IIR';
            app.IIRButton.Position = [11 14 65 22];

            % Create StartButton
            app.StartButton = uibutton(app.UIFigure, 'push');
            app.StartButton.ButtonPushedFcn = createCallbackFcn(app,
@StartButtonPushed, true);
            app.StartButton.Position = [804 787 100 22];
            app.StartButton.Text = 'Start';

            % Create SaveButton
            app.SaveButton = uibutton(app.UIFigure, 'push');
            app.SaveButton.ButtonPushedFcn = createCallbackFcn(app,
@SaveButtonPushed, true);
            app.SaveButton.Position = [804 752 100 22];
            app.SaveButton.Text = 'Save';

            % Create samplingrateEditFieldLabel
            app.samplingrateEditFieldLabel = uilabel(app.UIFigure);
            app.samplingrateEditFieldLabel.HorizontalAlignment = 'right';
            app.samplingrateEditFieldLabel.Position = [411 787 78 22];
            app.samplingrateEditFieldLabel.Text = 'sampling rate';

            % Create samplingrateEditField
            app.samplingrateEditField = uieditfield(app.UIFigure, 'text');
            app.samplingrateEditField.Position = [504 787 141 22];
            app.samplingrateEditField.Value = 'Enter new Sampling rate';

            % Create FiltertoPlotDropDownLabel
            app.FiltertoPlotDropDownLabel = uilabel(app.UIFigure);
            app.FiltertoPlotDropDownLabel.HorizontalAlignment = 'right';
            app.FiltertoPlotDropDownLabel.Position = [432 752 76 22];
            app.FiltertoPlotDropDownLabel.Text = {'Filter to Plot'; ''};

            % Create FiltertoPlotDropDown
            app.FiltertoPlotDropDown = uidropdown(app.UIFigure);
            app.FiltertoPlotDropDown.Items = {'0-170 HZ', '170-300 HZ', '300-610 HZ',
'610-1005 HZ', '1005-3000 HZ', '3-6 KHZ', '6-12 KHZ', '12-14 KHZ', '14-20 KHZ'};
            app.FiltertoPlotDropDown.ItemsData = {'1', '2', '3', '4', '5', '6', '7',
'8', '9'};
            app.FiltertoPlotDropDown.Position = [523 752 100 22];
            app.FiltertoPlotDropDown.Value = '3';

            % Create Panel
            app.Panel = uipanel(app.UIFigure);
            app.Panel.BorderType = 'none';
            app.Panel.BackgroundColor = [0.5098 0.6 0.8392];
            app.Panel.Position = [1 1 1103 737];

            % Create FrequencyBandGainsPanel
            app.FrequencyBandGainsPanel = uipanel(app.Panel);
            app.FrequencyBandGainsPanel.BorderType = 'none';
            app.FrequencyBandGainsPanel.TitlePosition = 'centertop';
            app.FrequencyBandGainsPanel.Title = 'Frequency Band Gains';
            app.FrequencyBandGainsPanel.BackgroundColor = [0.8078 0.851 0.8118];
            app.FrequencyBandGainsPanel.FontWeight = 'bold';
            app.FrequencyBandGainsPanel.FontSize = 22;
            app.FrequencyBandGainsPanel.Position = [1 432 771 306];

            % Create PlotButton
            app.PlotButton = uibutton(app.FrequencyBandGainsPanel, 'push');
            app.PlotButton.ButtonPushedFcn = createCallbackFcn(app,
@PlotButtonPushed, true);
            app.PlotButton.FontSize = 18;
            app.PlotButton.FontWeight = 'bold';
            app.PlotButton.Position = [503 20 100 29];
            app.PlotButton.Text = 'Plot';

            % Create ResetButton
            app.ResetButton = uibutton(app.FrequencyBandGainsPanel, 'push');
            app.ResetButton.ButtonPushedFcn = createCallbackFcn(app,
@ResetButtonPushed, true);
            app.ResetButton.FontSize = 14;
            app.ResetButton.Position = [622 22 132 29];
            app.ResetButton.Text = 'Reset';

            % Create ApplynewfiltersButton
            app.ApplynewfiltersButton = uibutton(app.FrequencyBandGainsPanel,
'push');
            app.ApplynewfiltersButton.ButtonPushedFcn = createCallbackFcn(app,
@ApplynewfiltersButtonPushed, true);
            app.ApplynewfiltersButton.FontSize = 14;
            app.ApplynewfiltersButton.Position = [24 21 136 29];
            app.ApplynewfiltersButton.Text = 'Apply new filters';

            % Create SounddoublenewFSButton
            app.SounddoublenewFSButton = uibutton(app.FrequencyBandGainsPanel,
'push');
            app.SounddoublenewFSButton.ButtonPushedFcn = createCallbackFcn(app,
@SounddoublenewFSButtonPushed, true);
            app.SounddoublenewFSButton.Position = [178.5 21 139 28];
            app.SounddoublenewFSButton.Text = 'Sound double new FS';

            % Create SoundhalfnewFSButton
            app.SoundhalfnewFSButton = uibutton(app.FrequencyBandGainsPanel, 'push');
            app.SoundhalfnewFSButton.ButtonPushedFcn = createCallbackFcn(app,
@SoundhalfnewFSButtonPushed, true);
            app.SoundhalfnewFSButton.Position = [347 22 139 28];
            app.SoundhalfnewFSButton.Text = 'Sound half new FS';

            % Create FreuquencyDomainPanel
            app.FreuquencyDomainPanel = uipanel(app.Panel);
            app.FreuquencyDomainPanel.BorderType = 'none';
            app.FreuquencyDomainPanel.TitlePosition = 'centertop';
            app.FreuquencyDomainPanel.Title = 'Freuquency Domain';
            app.FreuquencyDomainPanel.BackgroundColor = [0.8118 0.851 0.8118];
            app.FreuquencyDomainPanel.FontWeight = 'bold';
            app.FreuquencyDomainPanel.FontSize = 22;
            app.FreuquencyDomainPanel.Position = [1 1 771 432];

            % Create UIAxes3
            app.UIAxes3 = uiaxes(app.FreuquencyDomainPanel);
            title(app.UIAxes3, 'Title')
            xlabel(app.UIAxes3, 'time(s)')
            ylabel(app.UIAxes3, 'magnitude')
            app.UIAxes3.XTickLabelRotation = 0;
            app.UIAxes3.YTickLabelRotation = 0;
            app.UIAxes3.ZTickLabelRotation = 0;
            app.UIAxes3.Position = [24 199 267 166];

            % Create UIAxes4
            app.UIAxes4 = uiaxes(app.FreuquencyDomainPanel);
            title(app.UIAxes4, 'Title')
            xlabel(app.UIAxes4, 'time(s)')
            ylabel(app.UIAxes4, 'magnitude')
            app.UIAxes4.XTickLabelRotation = 0;
            app.UIAxes4.YTickLabelRotation = 0;
            app.UIAxes4.ZTickLabelRotation = 0;
            app.UIAxes4.Position = [429 193 249 179];

            % Create UIAxes6
            app.UIAxes6 = uiaxes(app.FreuquencyDomainPanel);
            xlabel(app.UIAxes6, 'time(s)')
            ylabel(app.UIAxes6, 'phase')
            app.UIAxes6.XTickLabelRotation = 0;
            app.UIAxes6.YTickLabelRotation = 0;
            app.UIAxes6.ZTickLabelRotation = 0;
            app.UIAxes6.Position = [429 14 247 162];

            % Create UIAxes5
            app.UIAxes5 = uiaxes(app.FreuquencyDomainPanel);
            title(app.UIAxes5, {''; ''})
            xlabel(app.UIAxes5, 'time(s)')
            ylabel(app.UIAxes5, 'phase')
            app.UIAxes5.XTickLabelRotation = 0;
            app.UIAxes5.YTickLabelRotation = 0;
            app.UIAxes5.ZTickLabelRotation = 0;
            app.UIAxes5.Position = [25 12 266 164];

            % Create MagnitudeLabel
            app.MagnitudeLabel = uilabel(app.FreuquencyDomainPanel);
            app.MagnitudeLabel.HorizontalAlignment = 'center';
            app.MagnitudeLabel.FontSize = 18;
            app.MagnitudeLabel.Position = [262 371 193 22];
            app.MagnitudeLabel.Text = 'Magnitude';

            % Create PhaseLabel
            app.PhaseLabel = uilabel(app.FreuquencyDomainPanel);
            app.PhaseLabel.HorizontalAlignment = 'center';
            app.PhaseLabel.FontSize = 18;
            app.PhaseLabel.Position = [262 162 193 22];
            app.PhaseLabel.Text = 'Phase';
```

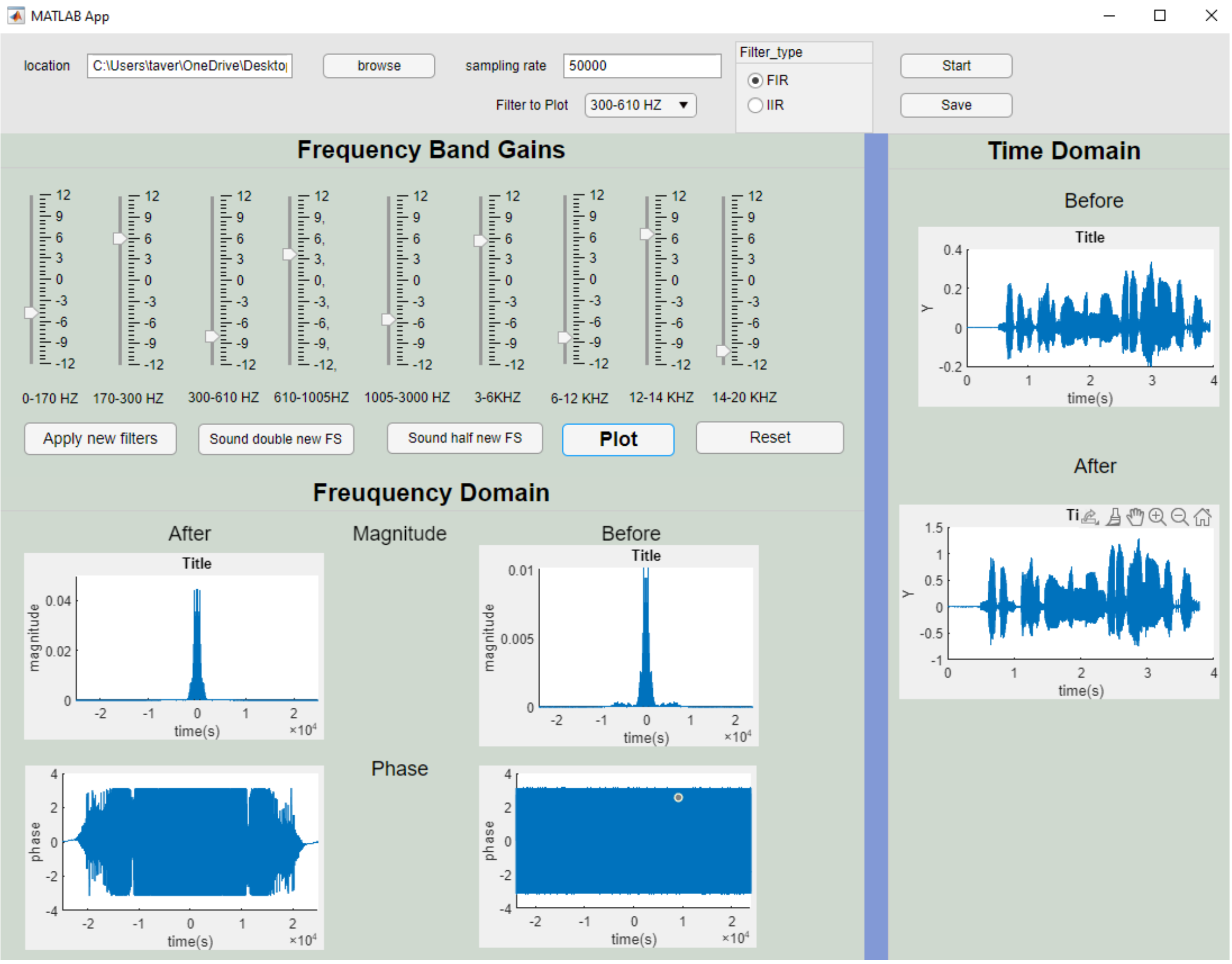
```
% Create AfterLabel
app.AfterLabel = uilabel(app.FreuquencyDomainPanel);
app.AfterLabel.HorizontalAlignment = 'center';
app.AfterLabel.FontSize = 18;
app.AfterLabel.Position = [75 371 193 22];
app.AfterLabel.Text = 'After';

% Create BeforeLabel_2
app.BeforeLabel_2 = uilabel(app.FreuquencyDomainPanel);
app.BeforeLabel_2.HorizontalAlignment = 'center';
app.BeforeLabel_2.FontSize = 18;
app.BeforeLabel_2.Position = [468 371 193 22];
app.BeforeLabel_2.Text = 'Before';

% Create HZSliderLabel
app.HZSliderLabel = uilabel(app.Panel);
app.HZSliderLabel.HorizontalAlignment = 'right';
app.HZSliderLabel.VerticalAlignment = 'bottom';
app.HZSliderLabel.Position = [17 495 55 22];
app.HZSliderLabel.Text = {'0-170 HZ'; ''};

% Create gain_170
app.gain_170 = uislider(app.Panel);
app.gain_170.Limits = [-12 12];
app.gain_170.Orientation = 'vertical';
app.gain_170.Position = [29 533 3 150];

% Create HZSlider_2Label
app.HZSlider_2Label = uilabel(app.Panel);
app.HZSlider_2Label.HorizontalAlignment = 'center';
app.HZSlider_2Label.VerticalAlignment = 'bottom';
app.HZSlider_2Label.Position = [82 495 69 22];
app.HZSlider_2Label.Text = {'170-300 HZ'; ''};

% Create gain_300
app.gain_300 = uislider(app.Panel);
app.gain_300.Limits = [-12 12];
app.gain_300.Orientation = 'vertical';
app.gain_300.Position = [108 532 3 150];

% Create HZSlider_3Label
app.HZSlider_3Label = uilabel(app.Panel);
app.HZSlider_3Label.HorizontalAlignment = 'right';
app.HZSlider_3Label.VerticalAlignment = 'bottom';
app.HZSlider_3Label.Position = [164 496 69 22];
app.HZSlider_3Label.Text = {'300-610 HZ'; ''};

% Create gain_610
app.gain_610 = uislider(app.Panel);
app.gain_610.Limits = [-12 12];
app.gain_610.Orientation = 'vertical';
app.gain_610.Position = [190 532 3 150];

% Create HZSlider_4Label
```

```
            app.HZSlider_4Label = uilabel(app.Panel);
            app.HZSlider_4Label.HorizontalAlignment = 'right';
            app.HZSlider_4Label.VerticalAlignment = 'bottom';
            app.HZSlider_4Label.Position = [241 496 72 22];
            app.HZSlider_4Label.Text = '610-1005HZ';

            % Create gain_1005
            app.gain_1005 = uislider(app.Panel);
            app.gain_1005.Limits = [-12 12];
            app.gain_1005.MajorTicks = [-12 -9 -6 -3 0 3 6 9 12];
            app.gain_1005.MajorTickLabels = {'-12,', '-9,', '-6,', '-3,', '0,', '3,', 
'6,', '9,', '12'};
            app.gain_1005.Orientation = 'vertical';
            app.gain_1005.MinorTicks = [-12 -11.4 -10.8 -10.2 -9.6 -9 -8.4 -7.8 -7.2 
-6.6 -6 -5.4 -4.8 -4.2 -3.6 -3 -2.4 -1.8 -1.2 -0.6 0 0.6 1.2 1.8 2.4 3 3.6 4.2 4.8 
5.4 6 6.6 7.2 7.8 8.4 9 9.6 10.2 10.8 11.4 12];
            app.gain_1005.Position = [259 532 3 150];

            % Create HZSlider_5Label
            app.HZSlider_5Label = uilabel(app.Panel);
            app.HZSlider_5Label.HorizontalAlignment = 'right';
            app.HZSlider_5Label.VerticalAlignment = 'bottom';
            app.HZSlider_5Label.Position = [321 496 82 22];
            app.HZSlider_5Label.Text = '1005-3000 HZ';

            % Create gain_3k
            app.gain_3k = uislider(app.Panel);
            app.gain_3k.Limits = [-12 12];
            app.gain_3k.Orientation = 'vertical';
            app.gain_3k.Position = [347 532 3 150];

            % Create KHZSliderLabel
            app.KHZSliderLabel = uilabel(app.Panel);
            app.KHZSliderLabel.HorizontalAlignment = 'right';
            app.KHZSliderLabel.VerticalAlignment = 'bottom';
            app.KHZSliderLabel.Position = [419 496 47 22];
            app.KHZSliderLabel.Text = '3-6KHZ';

            % Create gain_6k
            app.gain_6k = uislider(app.Panel);
            app.gain_6k.Limits = [-12 12];
            app.gain_6k.Orientation = 'vertical';
            app.gain_6k.Position = [429 532 3 150];

            % Create KHZSlider_2Label
            app.KHZSlider_2Label = uilabel(app.Panel);
            app.KHZSlider_2Label.HorizontalAlignment = 'center';
            app.KHZSlider_2Label.VerticalAlignment = 'bottom';
            app.KHZSlider_2Label.Position = [490 495 57 22];
            app.KHZSlider_2Label.Text = '6-12 KHZ';

            % Create gain_12k
            app.gain_12k = uislider(app.Panel);
            app.gain_12k.Limits = [-12 12];
            app.gain_12k.Orientation = 'vertical';
            app.gain_12k.Position = [504 533 3 150];

            % Create KHZSlider_3Label
            app.KHZSlider_3Label = uilabel(app.Panel);
            app.KHZSlider_3Label.HorizontalAlignment = 'right';
            app.KHZSlider_3Label.VerticalAlignment = 'bottom';
            app.KHZSlider_3Label.Position = [557 496 63 22];
            app.KHZSlider_3Label.Text = '12-14 KHZ';

            % Create gain_14k
            app.gain_14k = uislider(app.Panel);
            app.gain_14k.Limits = [-12 12];
            app.gain_14k.Orientation = 'vertical';
            app.gain_14k.Position = [577 532 3 150];

            % Create KHZSlider_4Label
            app.KHZSlider_4Label = uilabel(app.Panel);
            app.KHZSlider_4Label.HorizontalAlignment = 'right';
            app.KHZSlider_4Label.VerticalAlignment = 'bottom';
            app.KHZSlider_4Label.Position = [631 496 63 22];
            app.KHZSlider_4Label.Text = '14-20 KHZ';

            % Create gain_20k
            app.gain_20k = uislider(app.Panel);
            app.gain_20k.Limits = [-12 12];
            app.gain_20k.Orientation = 'vertical';
            app.gain_20k.Position = [645 532 3 150];

            % Create TimeDomainPanel
            app.TimeDomainPanel = uipanel(app.Panel);
            app.TimeDomainPanel.TitlePosition = 'centertop';
            app.TimeDomainPanel.Title = 'Time Domain';
            app.TimeDomainPanel.BackgroundColor = [0.8118 0.851 0.8118];
            app.TimeDomainPanel.FontWeight = 'bold';
            app.TimeDomainPanel.FontSize = 22;
            app.TimeDomainPanel.Position = [793 1 311 737];

            % Create UIAxes
            app.UIAxes = uiaxes(app.TimeDomainPanel);
            title(app.UIAxes, 'Title')
            xlabel(app.UIAxes, 'time(s)')
            ylabel(app.UIAxes, 'Y')
            app.UIAxes.XTickLabelRotation = 0;
            app.UIAxes.YTickLabelRotation = 0;
            app.UIAxes.ZTickLabelRotation = 0;
            app.UIAxes.Position = [27 494 268 160];

            % Create UIAxes2
            app.UIAxes2 = uiaxes(app.TimeDomainPanel);
            title(app.UIAxes2, 'Title')
            xlabel(app.UIAxes2, 'time(s)')
            ylabel(app.UIAxes2, 'Y')
            app.UIAxes2.XTickLabelRotation = 0;
            app.UIAxes2.YTickLabelRotation = 0;
            app.UIAxes2.ZTickLabelRotation = 0;
```

```
            app.UIAxes2.Position = [10 234 285 173];

            % Create BeforeLabel_3
            app.BeforeLabel_3 = uilabel(app.TimeDomainPanel);
            app.BeforeLabel_3.HorizontalAlignment = 'center';
            app.BeforeLabel_3.FontSize = 18;
            app.BeforeLabel_3.Position = [87 666 193 22];
            app.BeforeLabel_3.Text = 'Before';

            % Create AfterLabel_3
            app.AfterLabel_3 = uilabel(app.TimeDomainPanel);
            app.AfterLabel_3.HorizontalAlignment = 'center';
            app.AfterLabel_3.FontSize = 18;
            app.AfterLabel_3.Position = [88 430 193 22];
            app.AfterLabel_3.Text = 'After';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = Audio_Equalizer

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

FIR

Fs = 48 KHZ (original):



MATLAB App

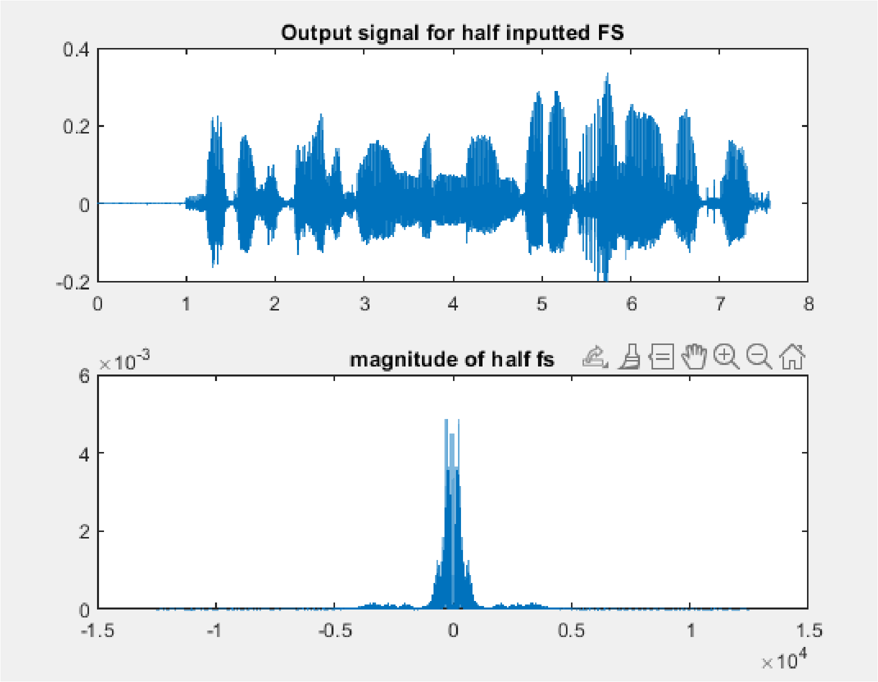
location C:\Users\taver\OneDrive\Deskto|

sampling rate 50000

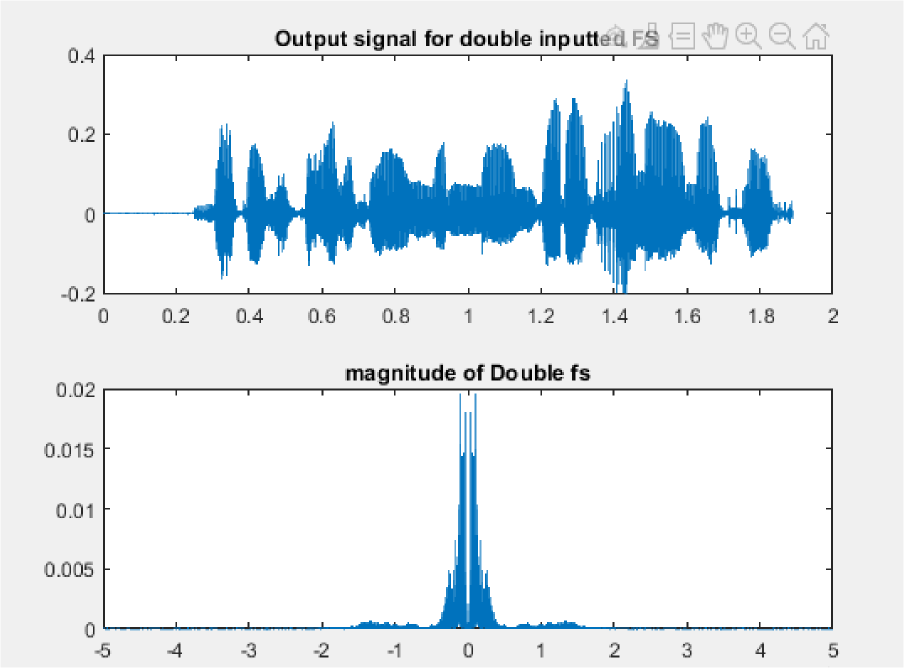
Filter to Plot 300-610 HZ ▼

Filter_type
● FIR
○ IIR

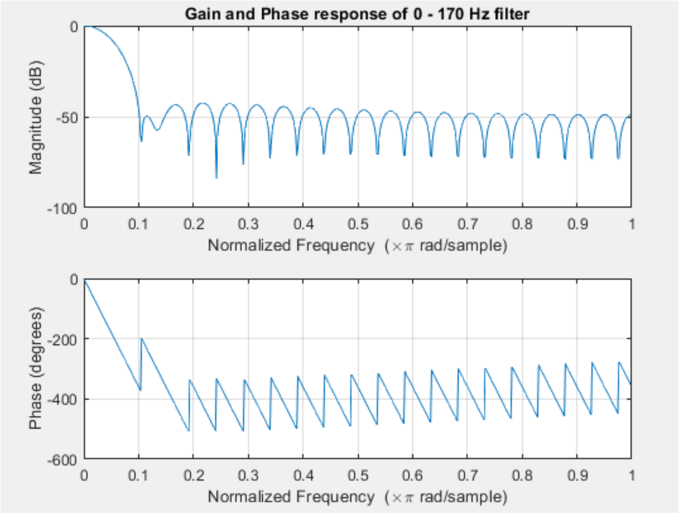
FIR Half

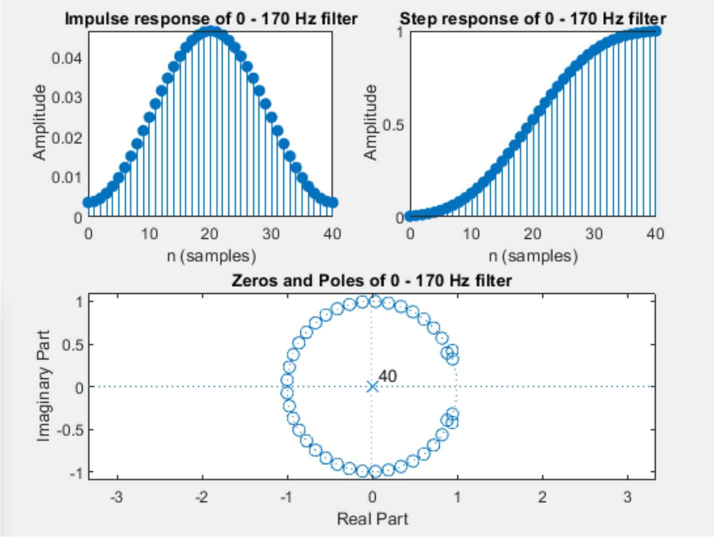


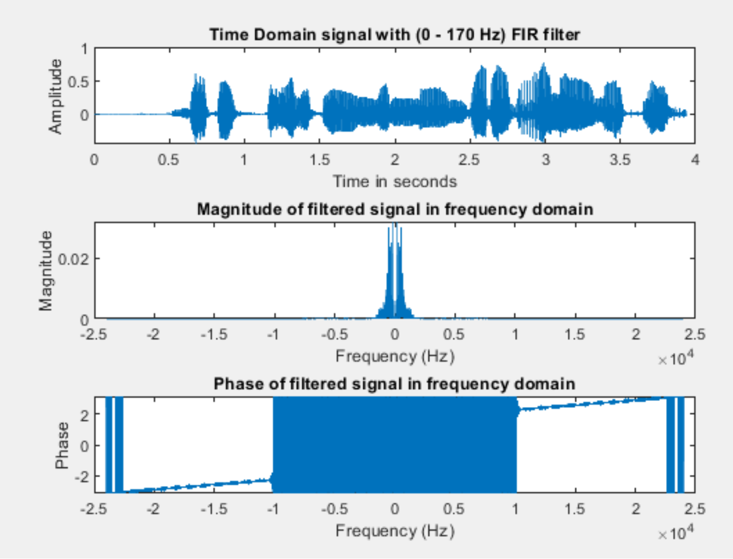
FIR Double



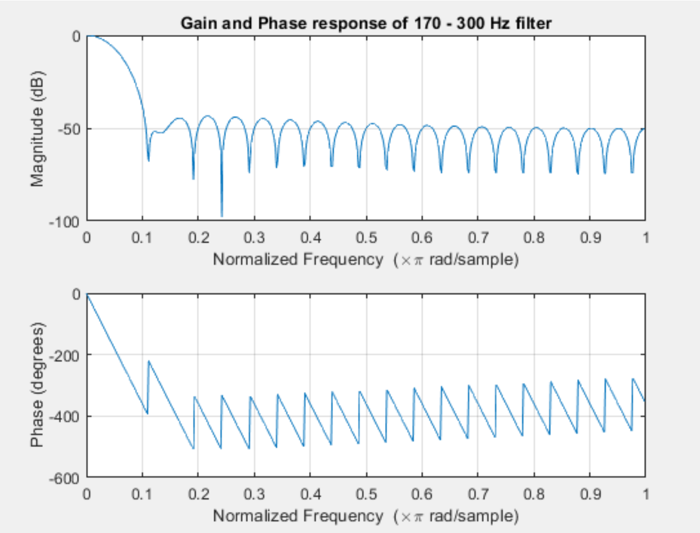
Filter 0- 170 Hz:

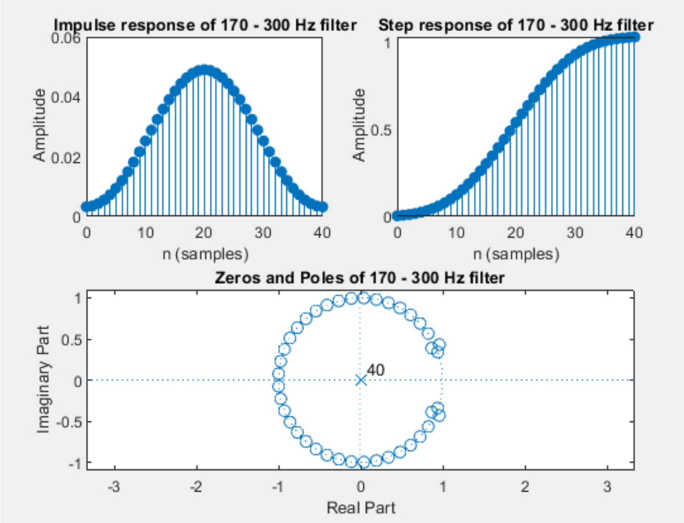


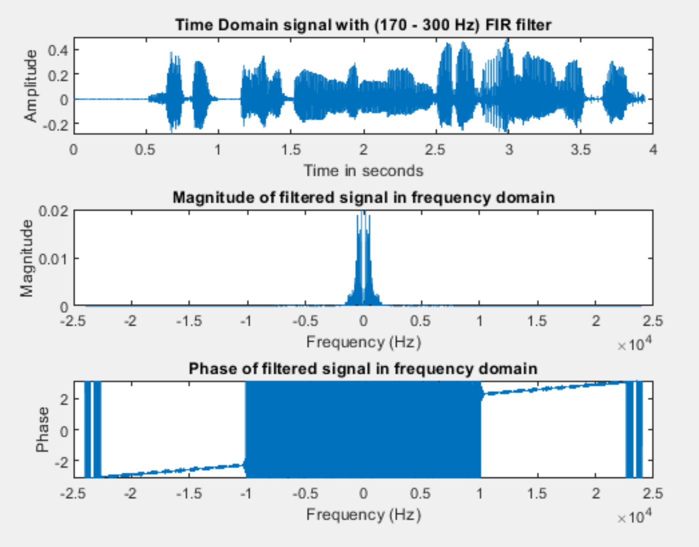




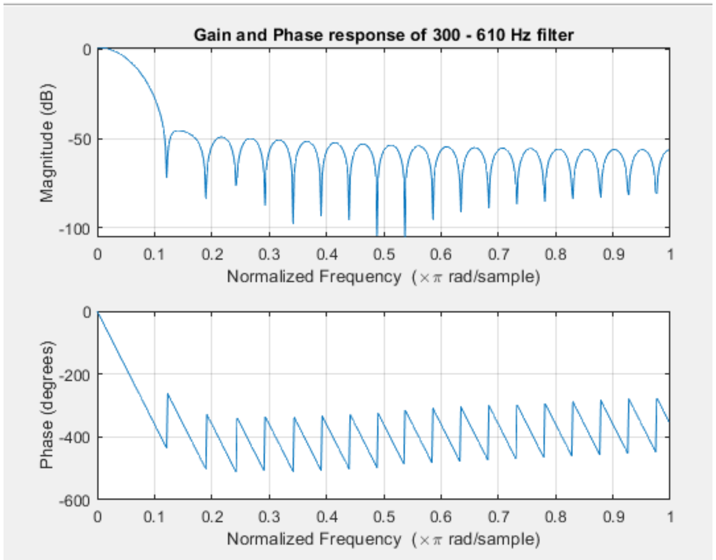
Filter 170- 300 Hz:

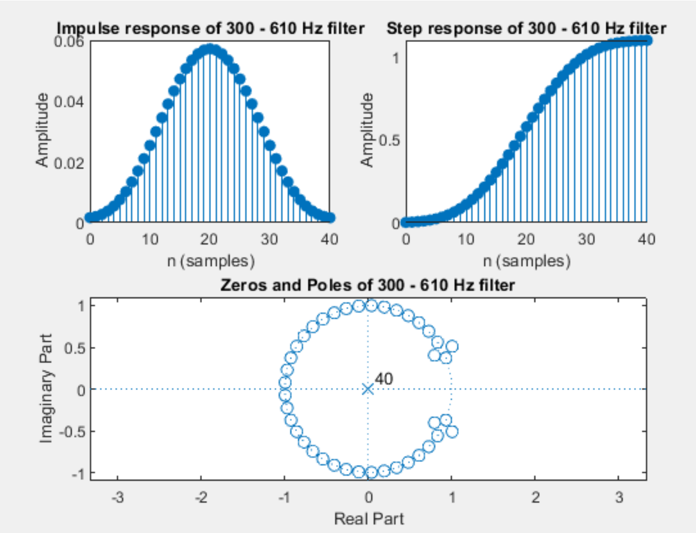


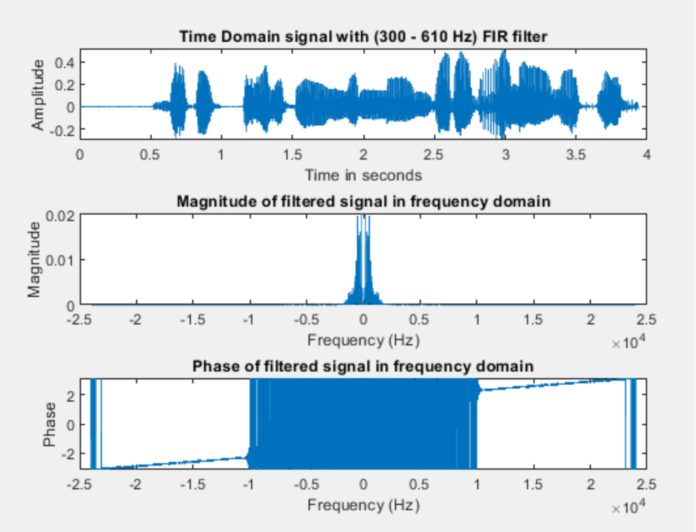




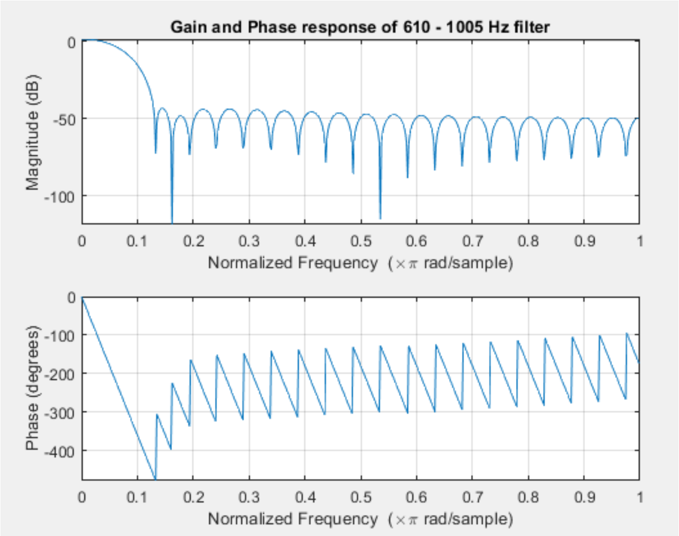
Filter 300-610 Hz:

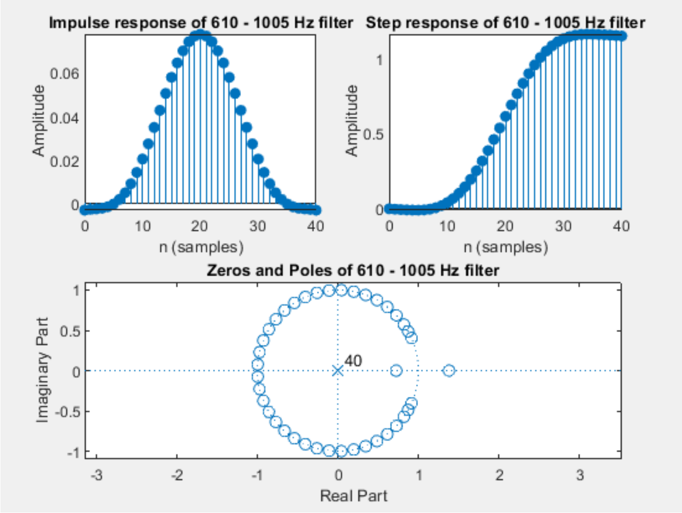


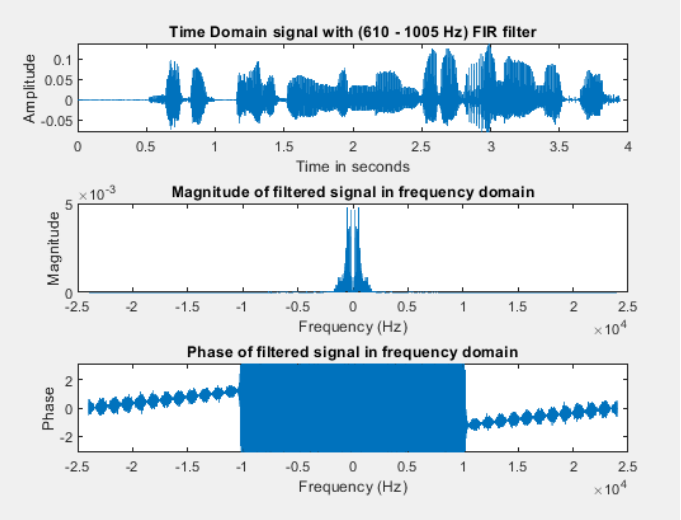




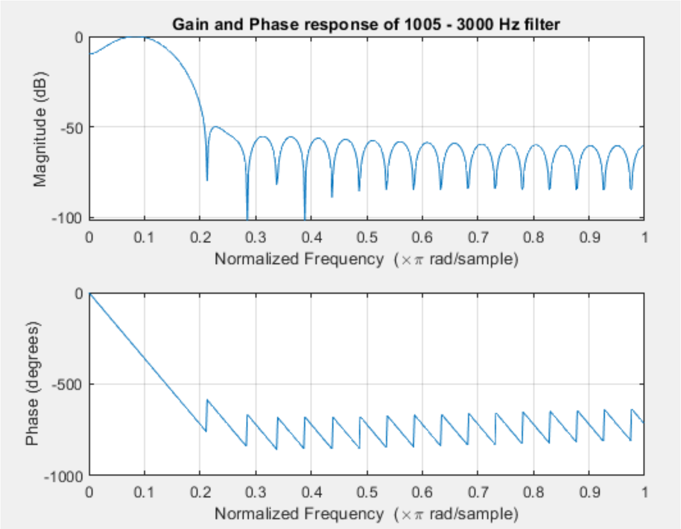
Filter 610-1005 Hz:

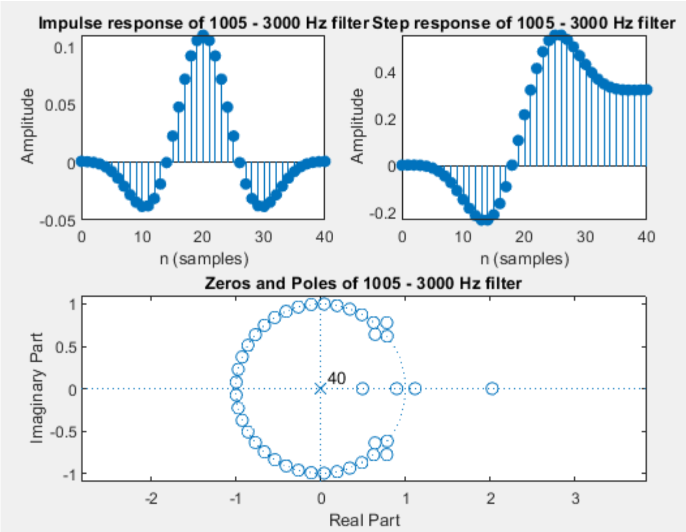


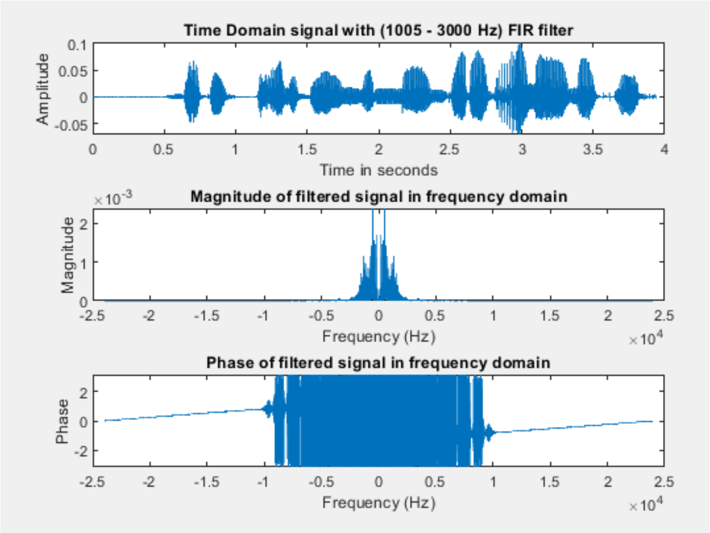




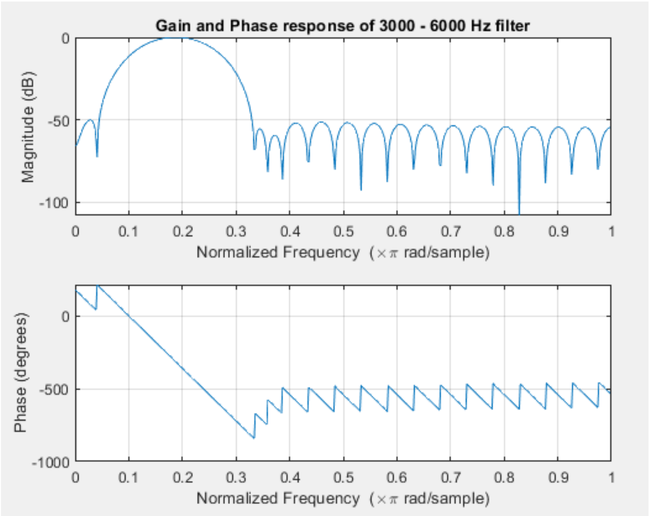
Filter 1005 – 3000 Hz:

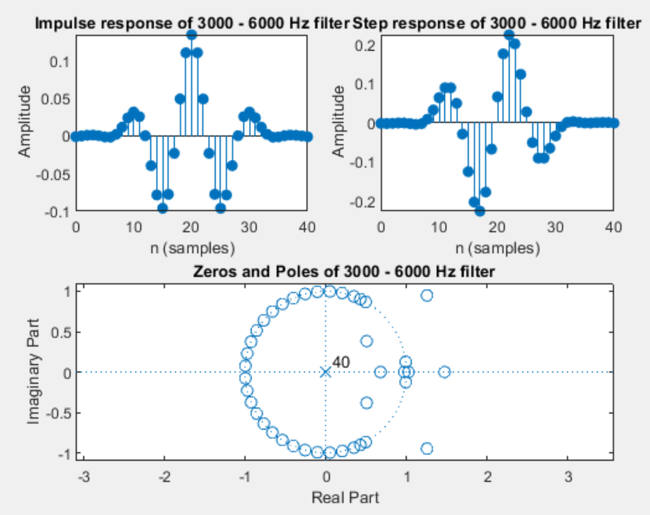


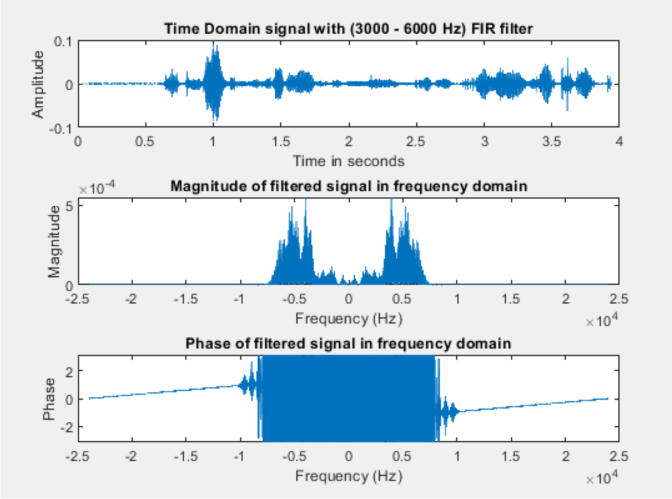




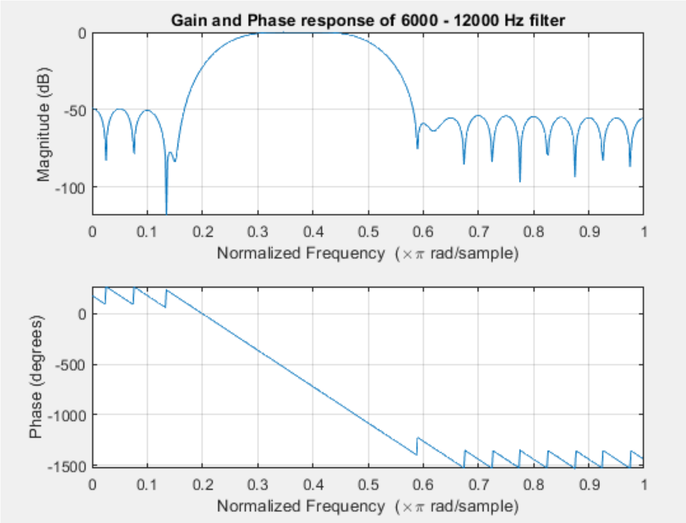
Filter 3000 – 6000 Hz:

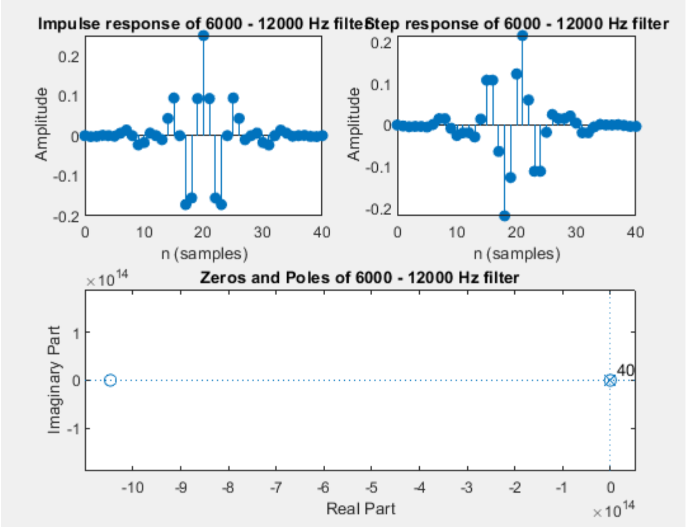


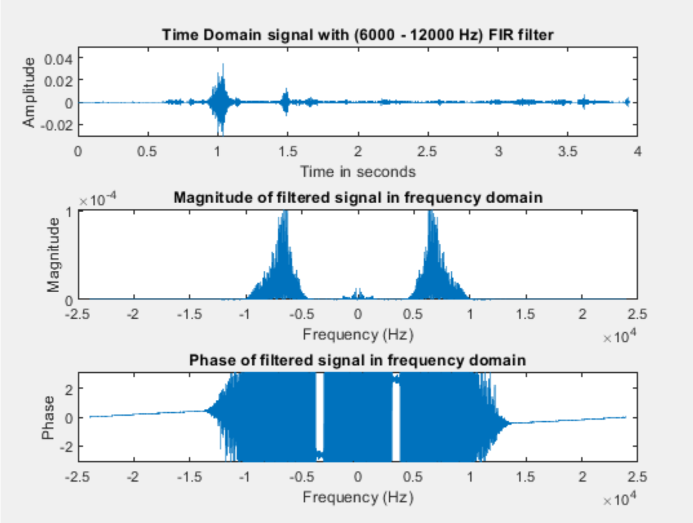




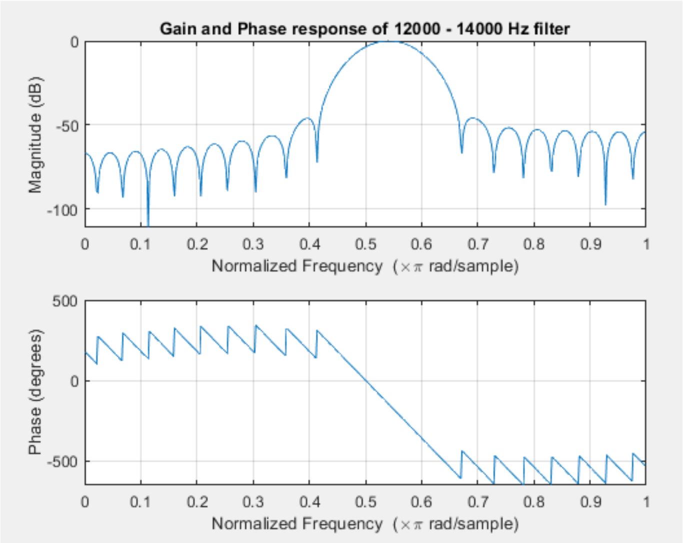
Filter 6000 – 12000 Hz:

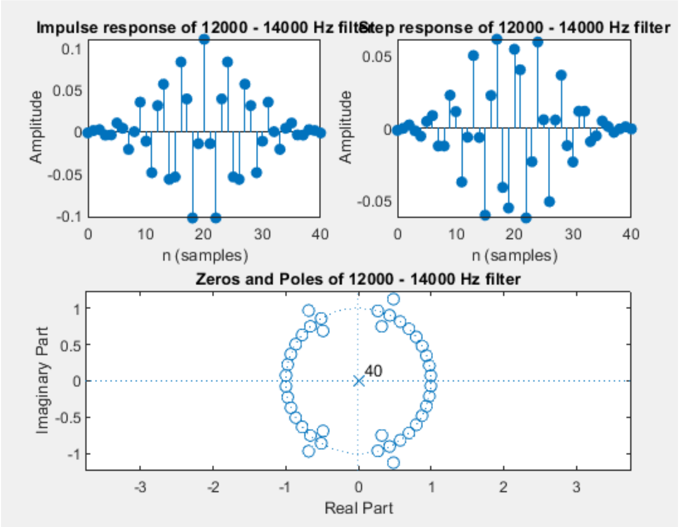


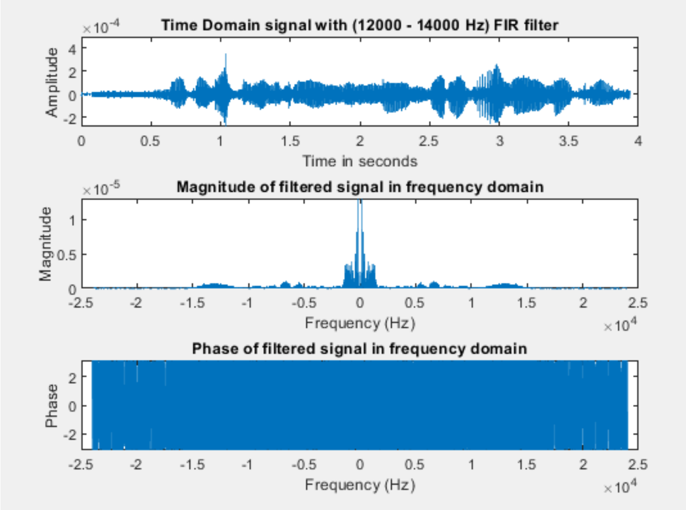




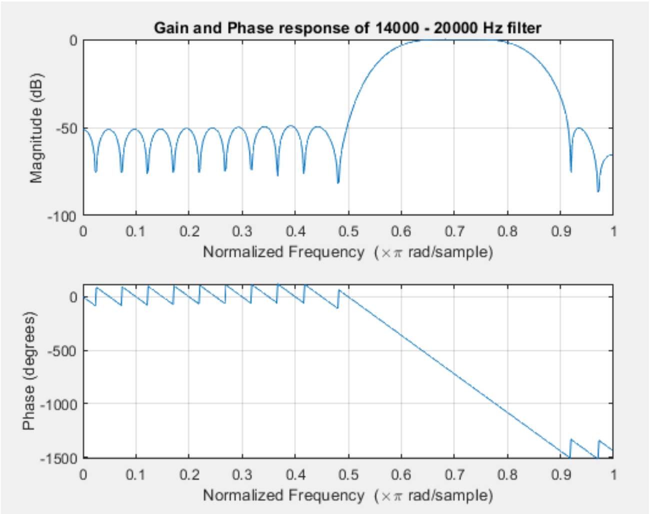
Filter 12000 – 14000 Hz:

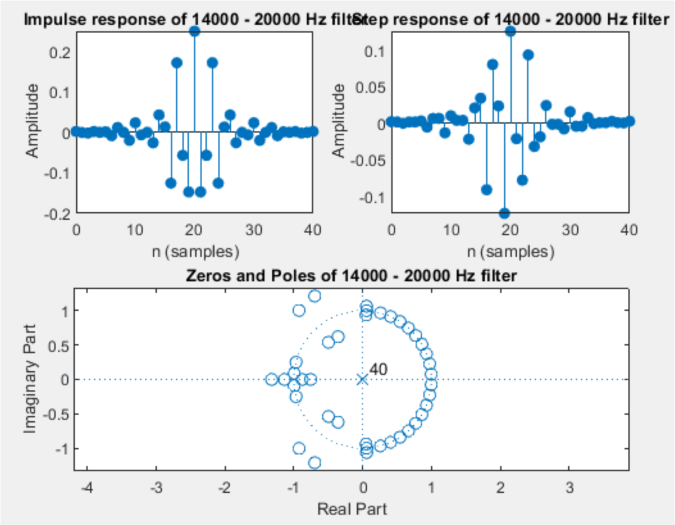


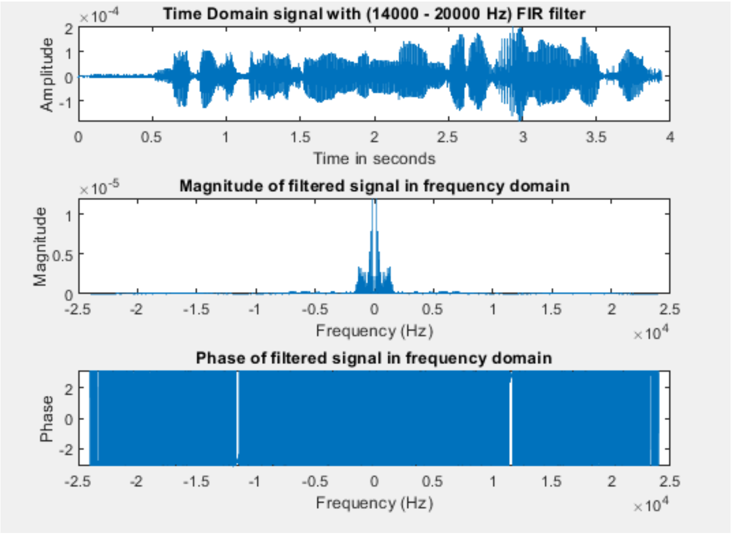




Filter 14000 – 20000 Hz:

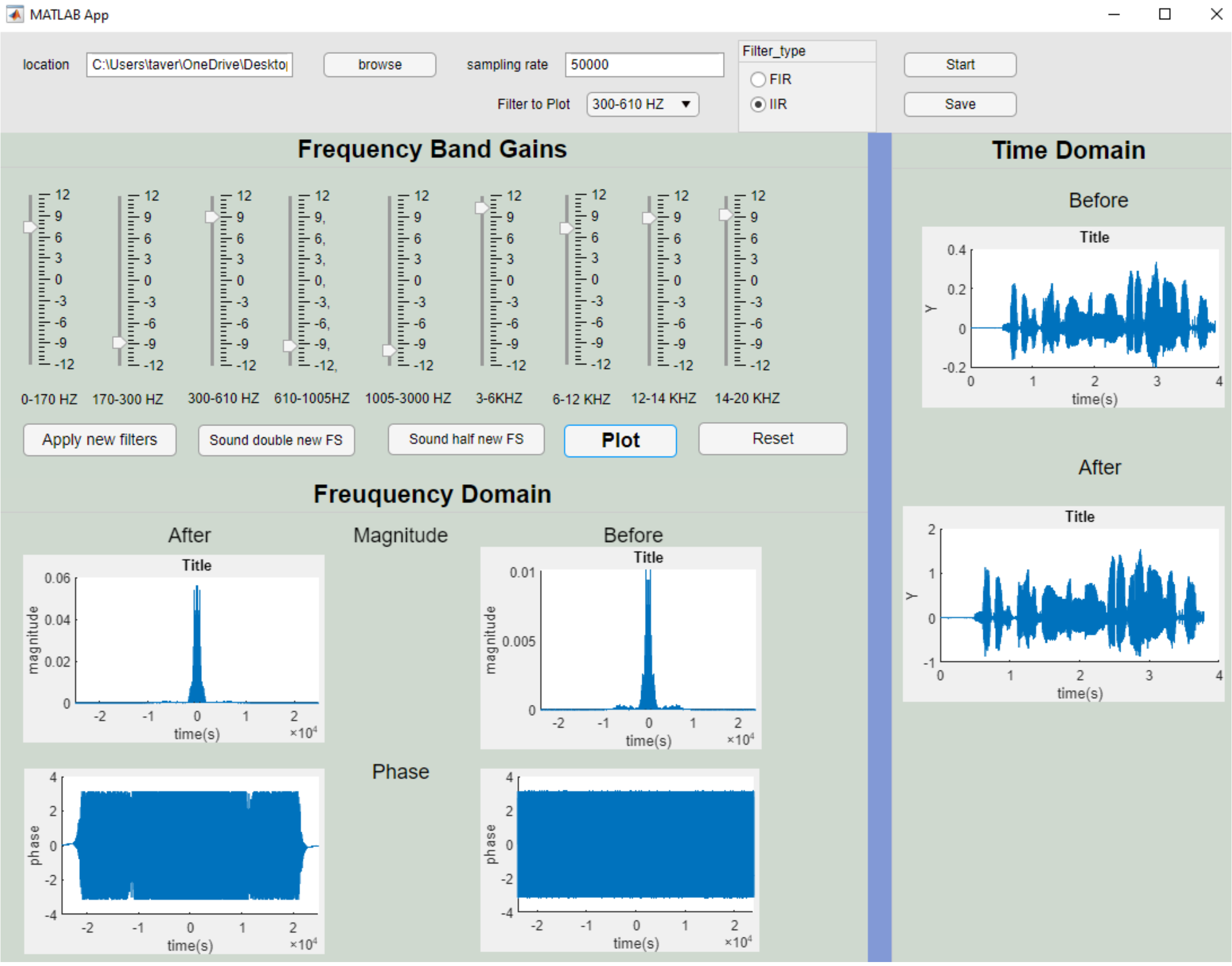






IIR

Fs = 48 KHZ (original):



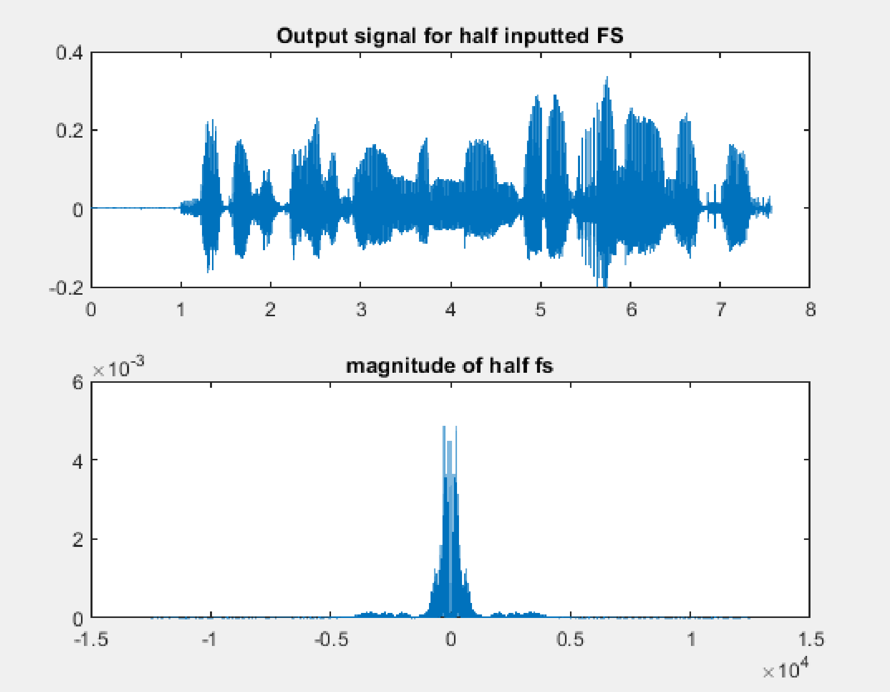
location: C:\Users\taver\OneDrive\Deskto|

sampling rate: 50000

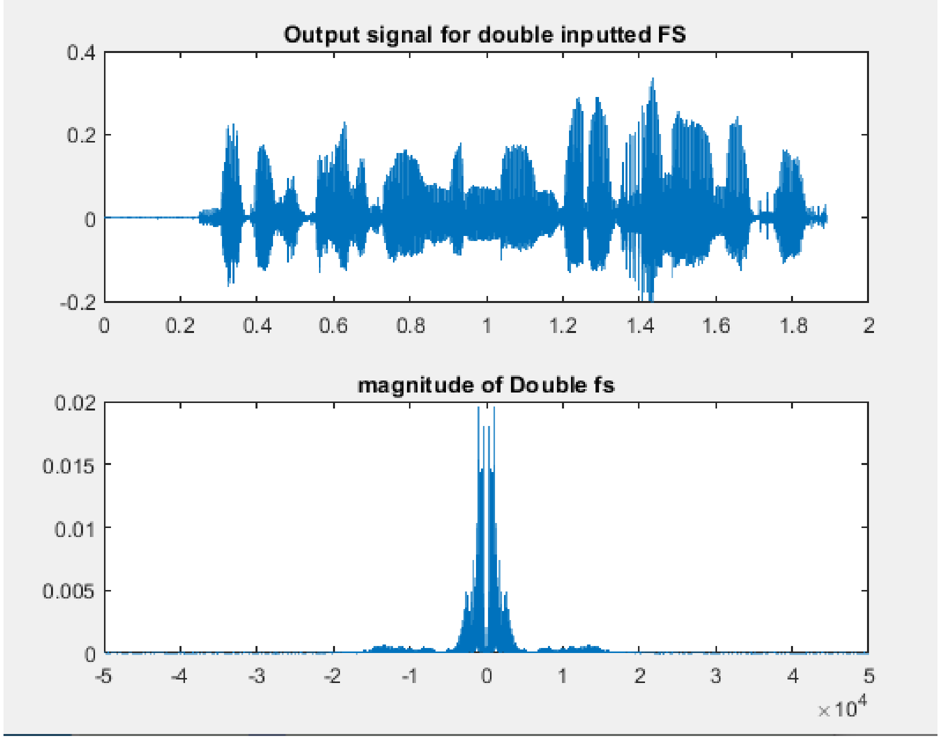
Filter to Plot: 300-610 HZ

Filter_type: ○ FIR ● IIR

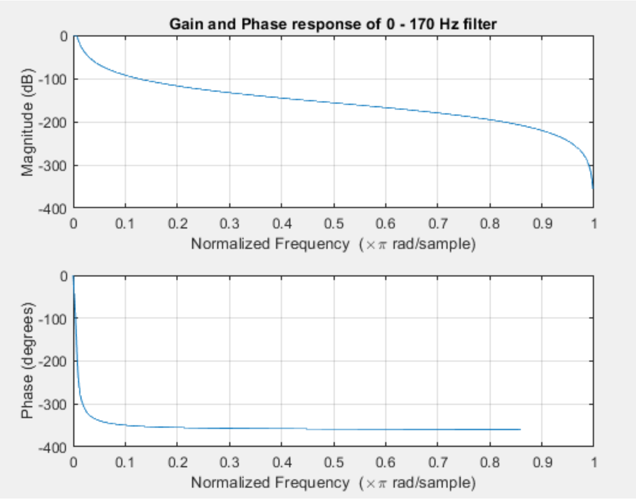
IIR Half

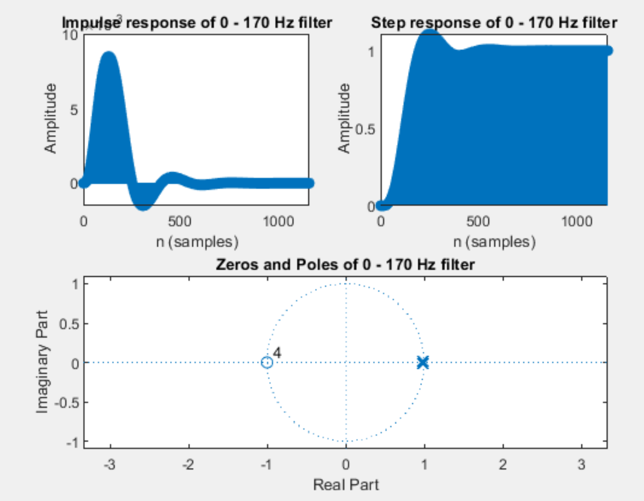


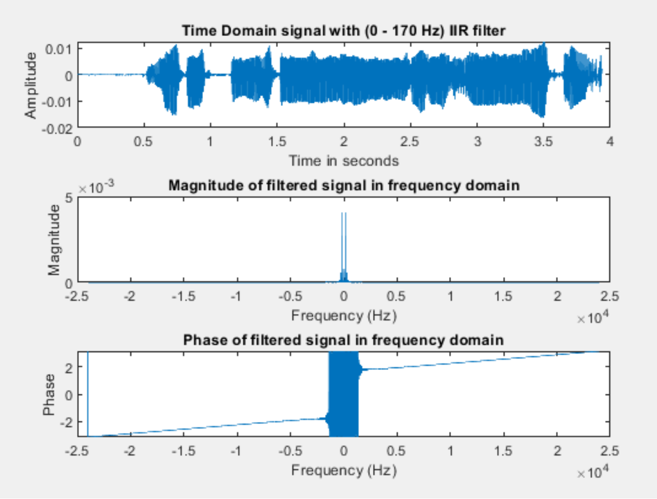
IIR Double



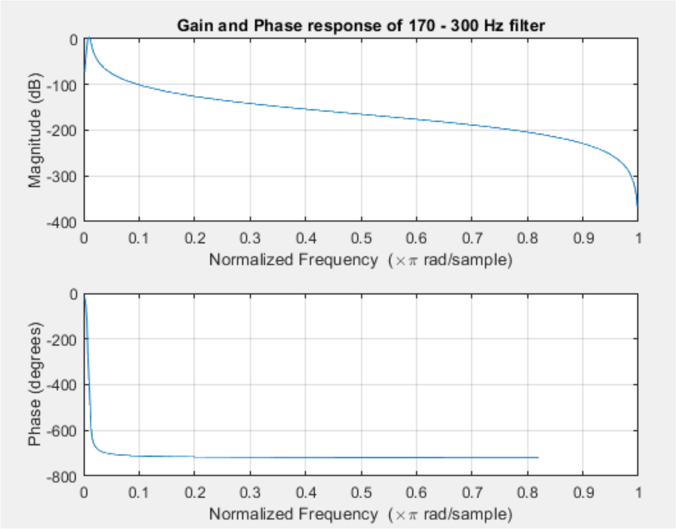
Filter 0 – 170 Hz:

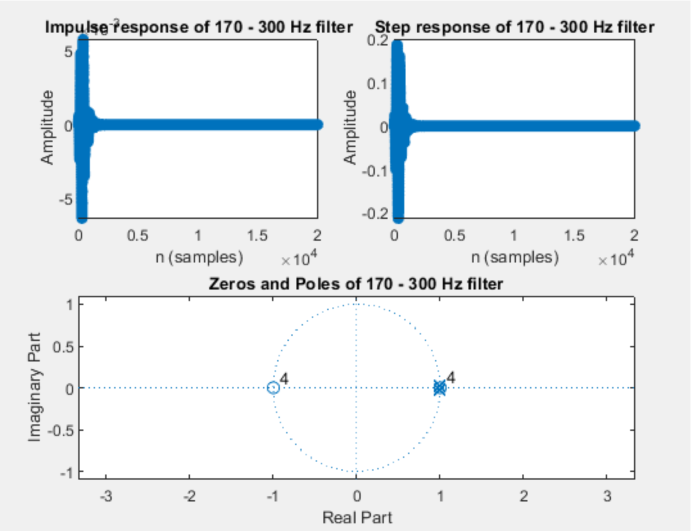


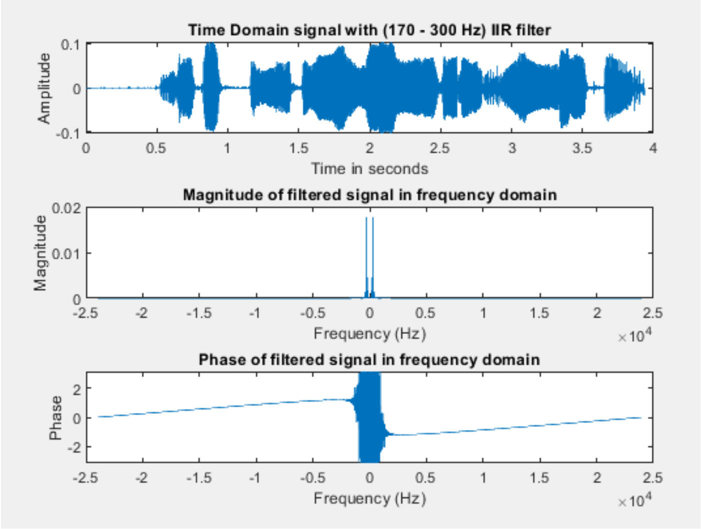




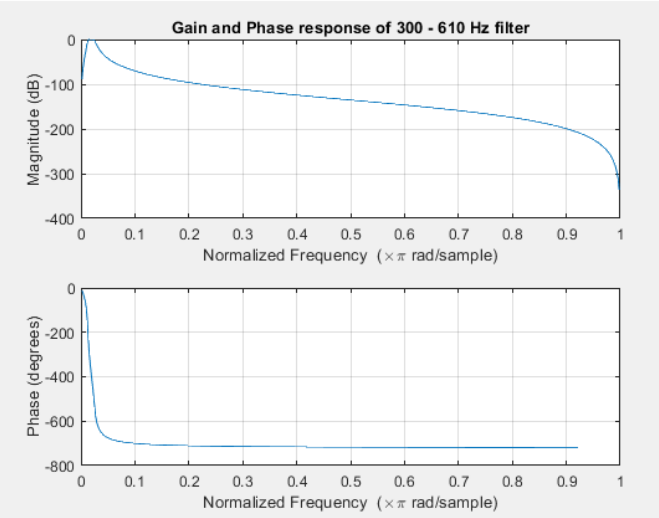
Filter 170 – 300 Hz:

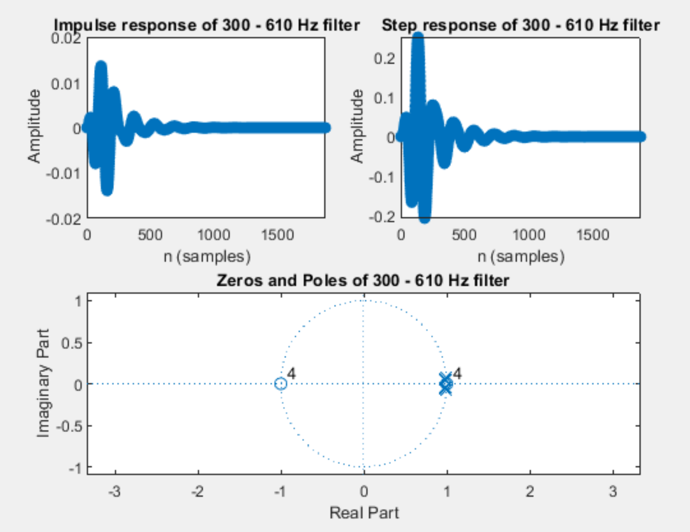


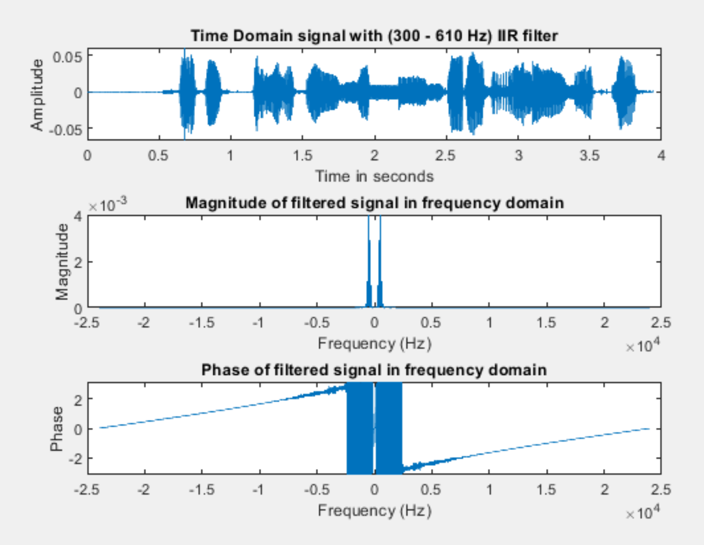




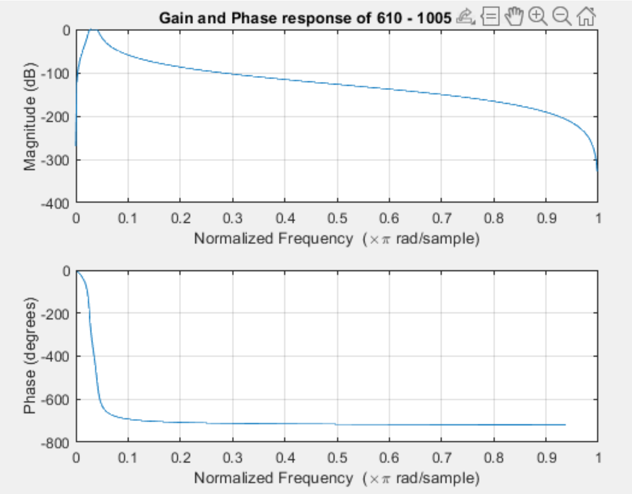
Filter 300 – 610 Hz:

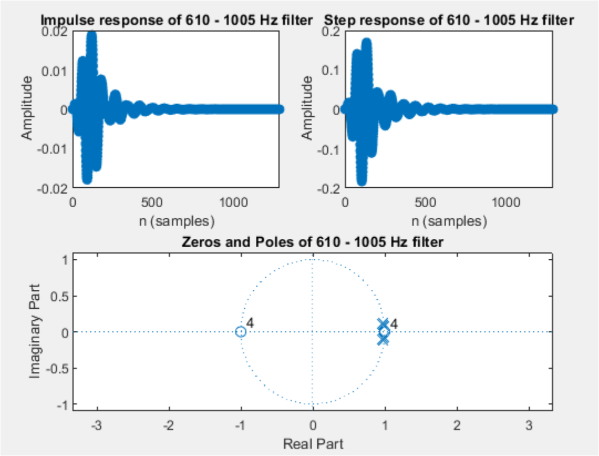


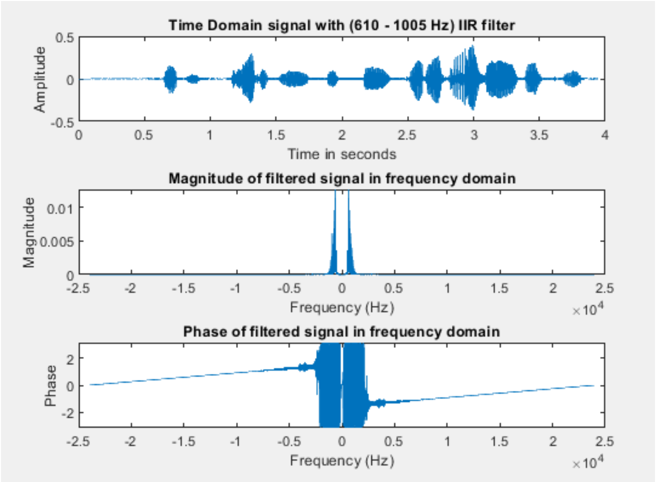




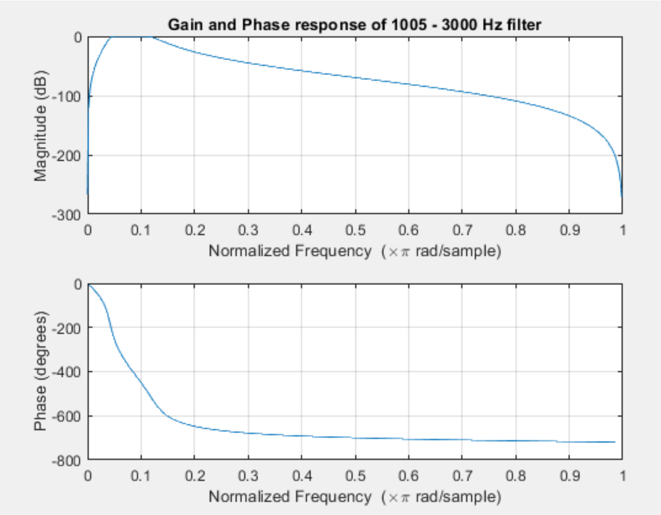
Filter 610 – 1005 Hz:

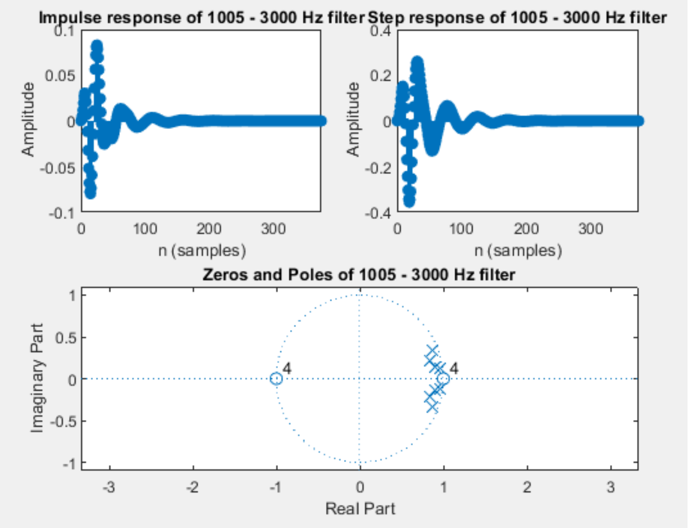


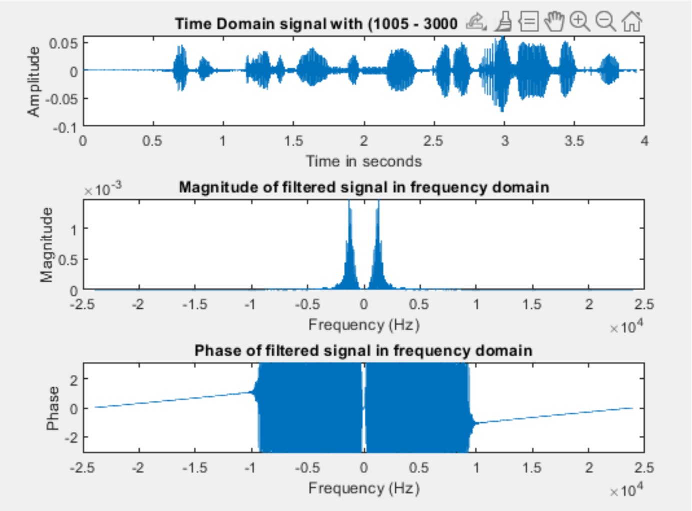




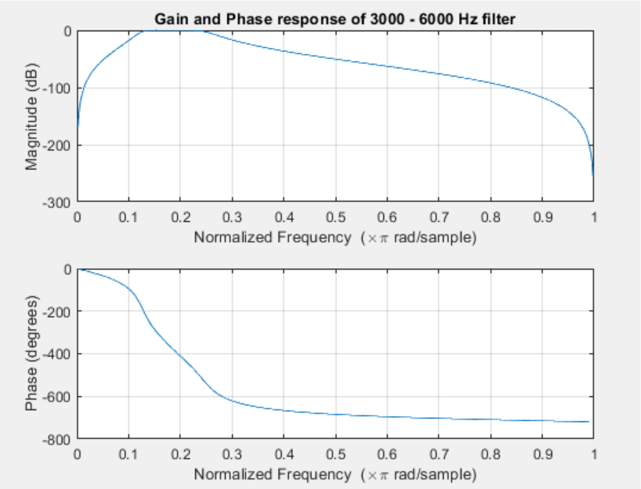
Filter 1005 – 3000 Hz:

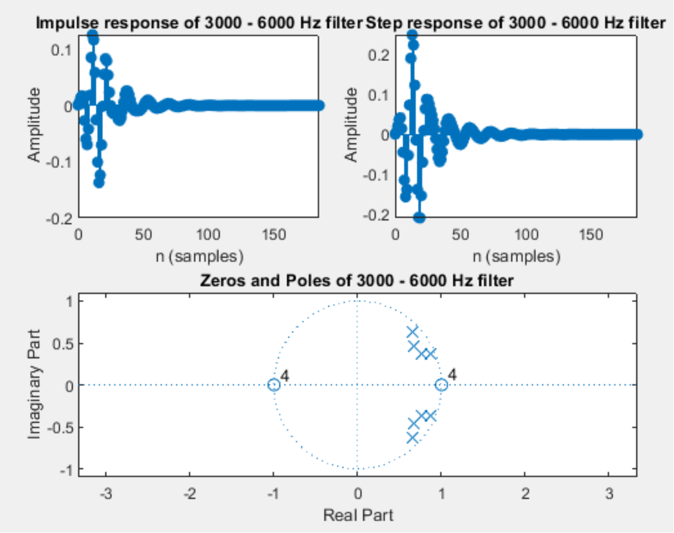


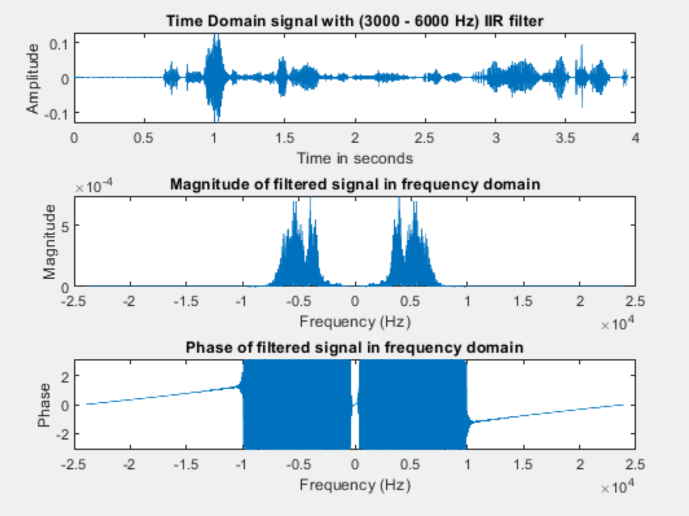




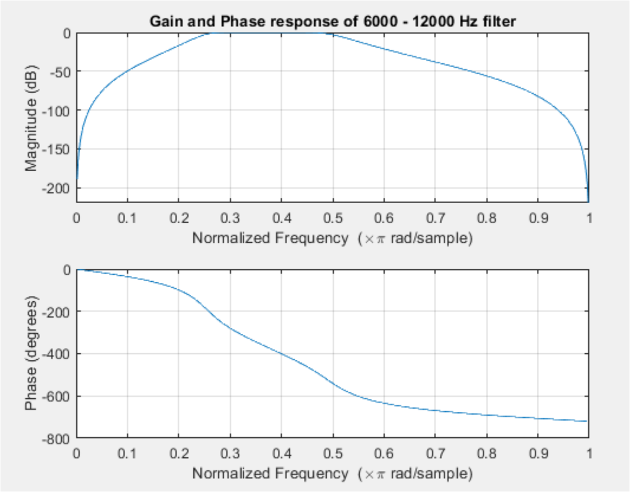
Filter 3000 – 6000 Hz:

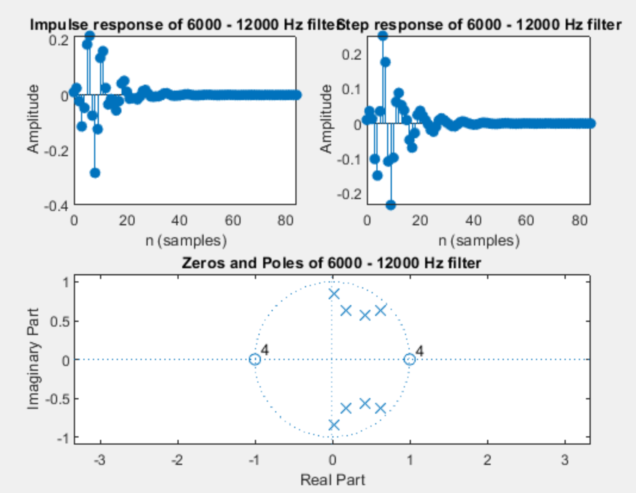


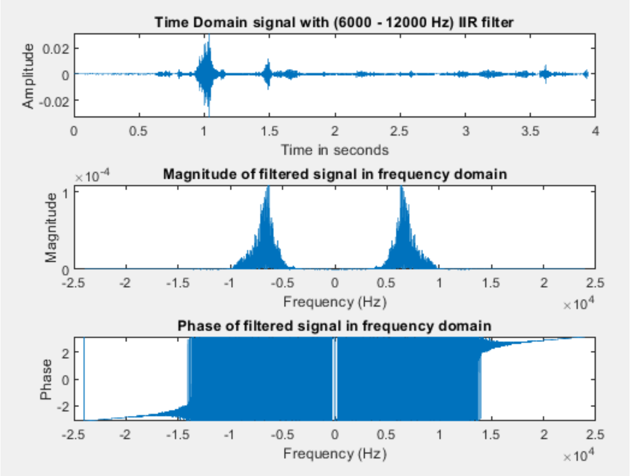




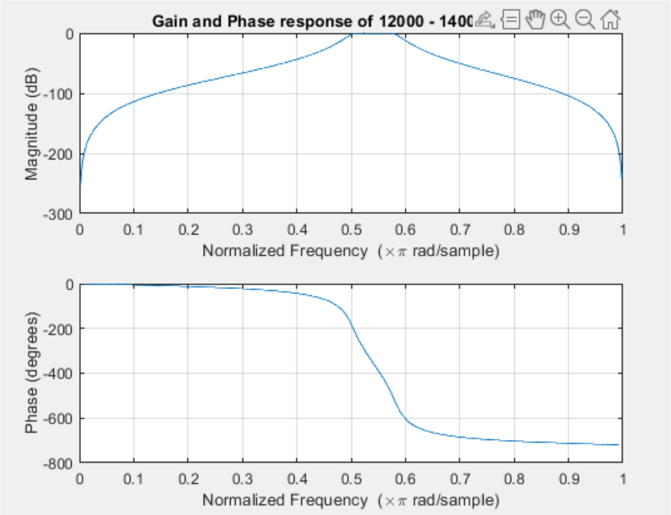
Filter 6000 – 12000 Hz:

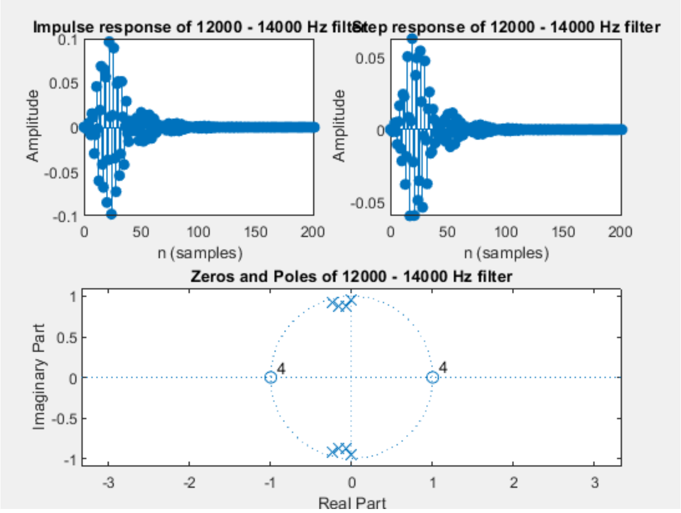


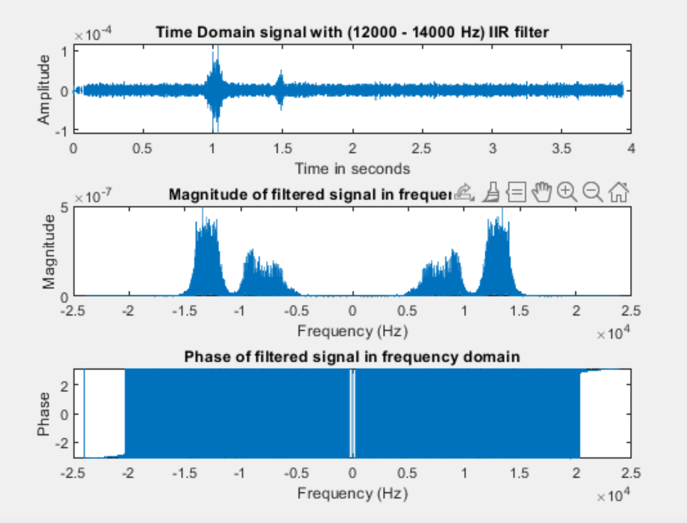




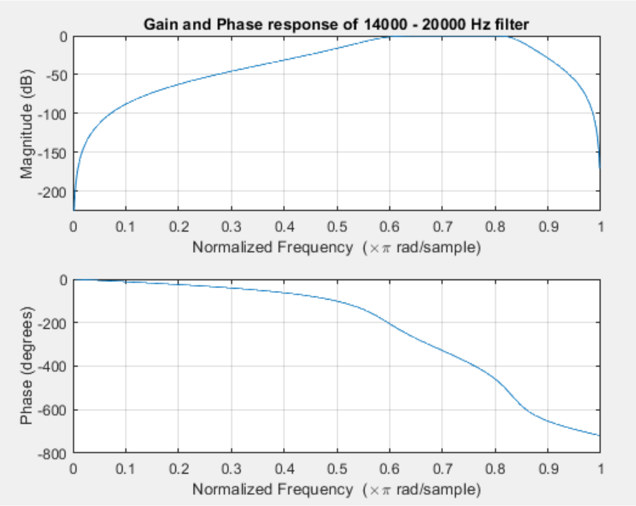
Filter 12000– 14000 Hz:

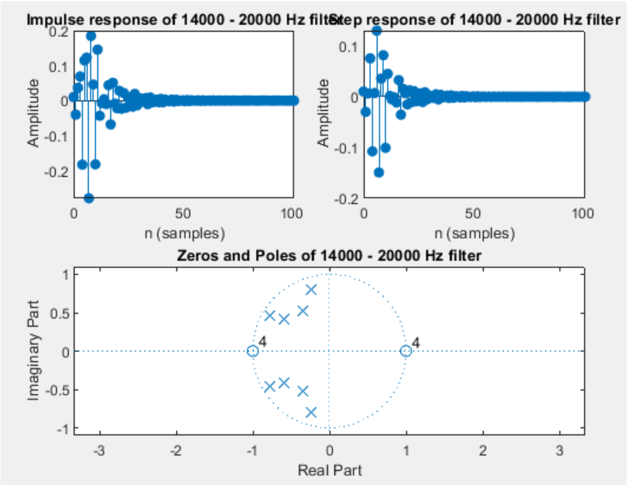


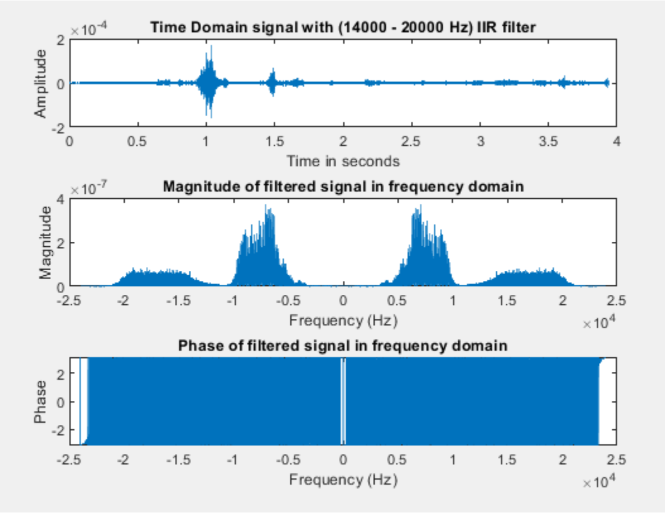




Filter 14000– 20000 Hz:


Gain and Phase response of 14000 - 20000 Hz filter
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)


Impulse response of 14000 - 20000 Hz filter
Amplitude
n (samples)
Step response of 14000 - 20000 Hz filter
Amplitude
n (samples)
Zeros and Poles of 14000 - 20000 Hz filter
Imaginary Part
Real Part


Time Domain signal with (14000 - 20000 Hz) IIR filter
×10^-4
Amplitude
Time in seconds
Magnitude of filtered signal in frequency domain
×10^-7
Magnitude
Frequency (Hz)
×10^4
Phase of filtered signal in frequency domain
Phase
Frequency (Hz)
×10^4