DSP Project

Audio equalizer

Students :

Rawan Saleh galal 9186

Mohammed Amir Elhofy 8790

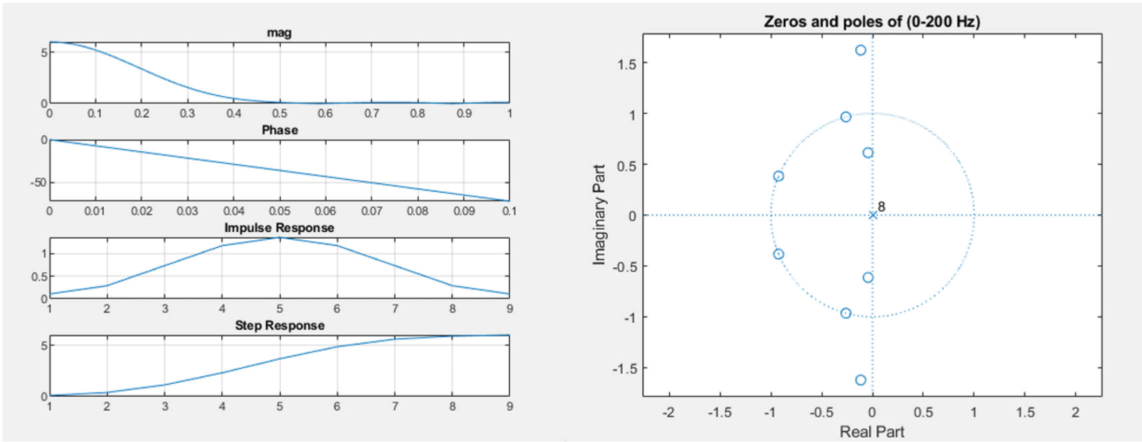
CODE:

```
% Prompt for file name and read audio

audioFileName = input('Enter the name of the audio file: ', 's');

[inputSignal, inputFs] = audioread(audioFileName);

originalSignal = inputSignal;

processedSignal = 0;

% Frequency bands and user-defined gains

freqBands = [0, 200, 400, 800, 1200, 3000, 6000, 12000, 15000, 20000];

numBands = length(freqBands) - 1;

bandGains = zeros(1, numBands);

disp('Enter gain (in dB) for each frequency band:');

for idx = 1:numBands

   prompt = sprintf('Gain for %d - %d Hz: ', freqBands(idx), freqBands(idx+1));

   bandGains(idx) = input(prompt);

end

% Prompt for new sample rate

sampleRatePrompt = sprintf('Enter output sample rate (original Fs = %d): ', inputFs);

outputFs = input(sampleRatePrompt);

% Resample if needed

if outputFs ~= inputFs

   resampleRatio = outputFs / inputFs;

   inputSignal = resample(inputSignal, round(outputFs), inputFs);

end

halfFs = outputFs / 2;

Filter selection loop

While  true

   disp('Filter Types:');

   disp('1) IIR Filter');

   disp('2) FIR Filter');
```

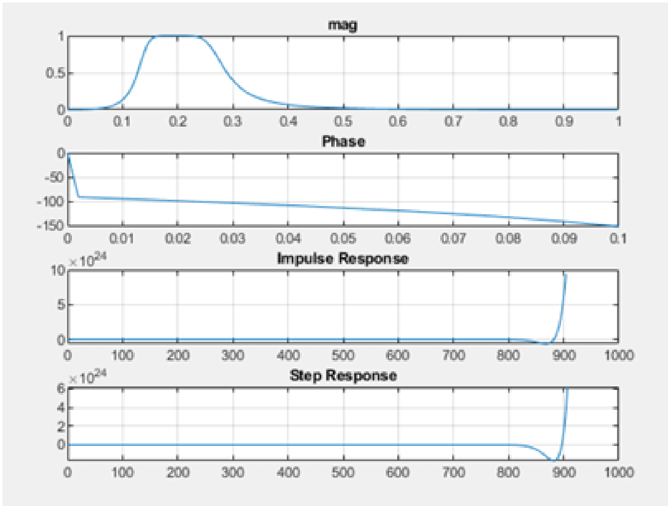
```
filterChoice = input('Choose a filter type (1 or 2): ');


switch filterChoice
    case 1  % IIR Filter
        filterOrder = input('Enter IIR filter order: ');
        for idx = 1:numBands
            if idx == 1
                Wn = freqBands(idx+1) / halfFs;
                [b, a] = butter(filterOrder, Wn);
            else
                Wn = [freqBands(idx), freqBands(idx+1)] / halfFs;
                [b, a] = butter(filterOrder, Wn);
            end

            sysTF = tf(b, a) * bandGains(idx);
            bandFiltered = filter(b, a, inputSignal) * bandGains(idx);
            processedSignal = processedSignal + bandFiltered;

            [H, w] = freqz(b, a);
            figure;
            subplot(4, 1, 1); plot(w/pi, abs(H)); title('Magnitude Response'); grid on;
            subplot(4, 1, 2); plot(w/pi, angle(H) * 180 / pi); title('Phase Response'); grid on;
            subplot(4, 1, 3); impulse(sysTF); title('Impulse Response'); grid on;
            subplot(4, 1, 4); step(sysTF); title('Step Response'); grid on;

            figure;
            pzmap(sysTF);
            title(sprintf('Poles and Zeros (%d - %d Hz)', freqBands(idx), freqBands(idx+1)));
        end
        break;
```

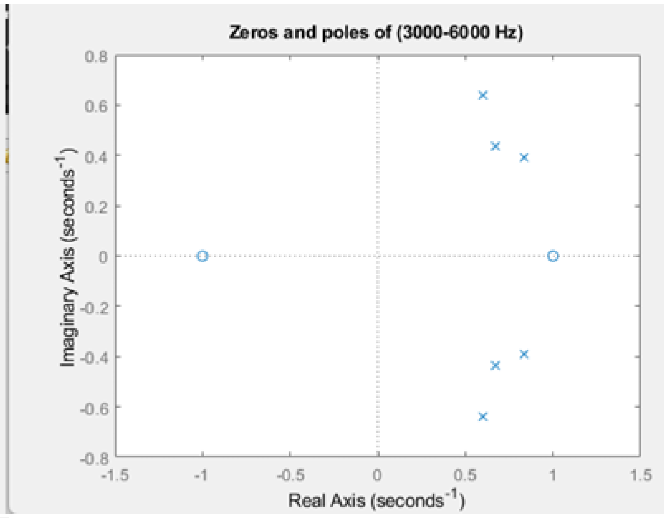
```
case 2  % FIR Filter
    filterOrder = input('Enter FIR filter order: ');
    for idx = 1:numBands
        if idx == 1
            Wn = freqBands(idx+1) / halfFs;
            firCoeffs = fir1(filterOrder, Wn);
        else
            Wn = [freqBands(idx), freqBands(idx+1)] / halfFs;
            firCoeffs = fir1(filterOrder, Wn);
        end

        firCoeffs = firCoeffs * bandGains(idx);
        bandFiltered = filter(firCoeffs, 1, inputSignal);
        processedSignal = processedSignal + bandFiltered;

        [H, w] = freqz(firCoeffs, 1);
        figure;
        subplot(4, 1, 1); plot(w/pi, abs(H)); title('Magnitude Response'); grid on;
        subplot(4, 1, 2); plot(w/pi, angle(H) * 180 / pi); title('Phase Response'); grid on;
        subplot(4, 1, 3); impz(firCoeffs); title('Impulse Response'); grid on;
        subplot(4, 1, 4); stepz(firCoeffs); title('Step Response'); grid on;

        figure;
        zplane(firCoeffs, 1);
        title(sprintf('Poles and Zeros (%d - %d Hz)', freqBands(idx), freqBands(idx+1)));
    end
    break;

otherwise
    disp('Invalid choice. Please enter 1 or 2.');
end
```

```
end

% Normalize output
processedSignal = processedSignal / max(abs(processedSignal));

% Save output audio
outputFileName = sprintf('filtered_%s.wav', audioFileName);
audiowrite(outputFileName, processedSignal, outputFs);
disp(['Filtered audio saved as ', outputFileName]);

% Time-domain plots
figure;
subplot(2, 1, 1); plot(originalSignal); title('Original Signal');
subplot(2, 1, 2); plot(processedSignal); title('Filtered Signal');

% Frequency-domain plots
fftOriginal = abs(fftshift(fft(originalSignal)));
fftFiltered = abs(fftshift(fft(processedSignal)));

fOriginal = linspace(-inputFs/2, inputFs/2, length(fftOriginal));
fFiltered = linspace(-outputFs/2, outputFs/2, length(fftFiltered));

figure;
subplot(2, 1, 1); plot(fOriginal, fftOriginal); title('Original Signal Spectrum');
subplot(2, 1, 2); plot(fFiltered, fftFiltered); title('Filtered Signal Spectrum');
```




```
Please enter the name of the audio file: test.wav
Please enter the gain for each freq range:
Enter gain in db for 0-200 Hz: 6
Enter gain in db for 200-400 Hz: 9
Enter gain in db for 400-800 Hz: 3
Enter gain in db for 800-1200 Hz: 3
Enter gain in db for 1200-3000 Hz: 3
Enter gain in db for 3000-6000 Hz: 2
Enter gain in db for 6000-12000 Hz: 5
Enter gain in db for 12000-15000 Hz: 8
Enter gain in db for 15000-20000 Hz: 1
Enter output sample rate (orignal Fs = 44100): 80000
1) IIR Filter
2) FIR Filter
Please choose the type of filter:2
enter your order: 8
Filtered audio saved successfully as filteretestd "fliename".wav
```


mag
Phase
Impulse Response
Step Response


Zeros and poles of (3000-6000 Hz)
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

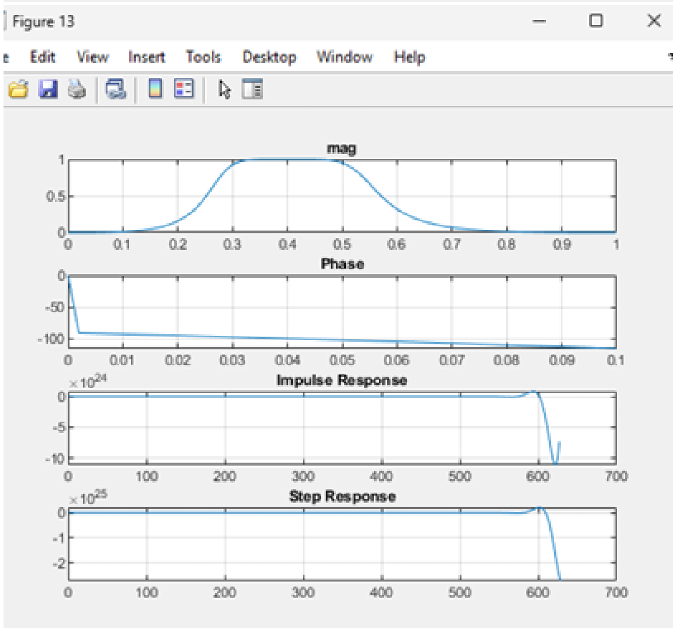


Figure 13
Edit View Insert Tools Desktop Window Help
mag
Phase
Impulse Response
Step Response

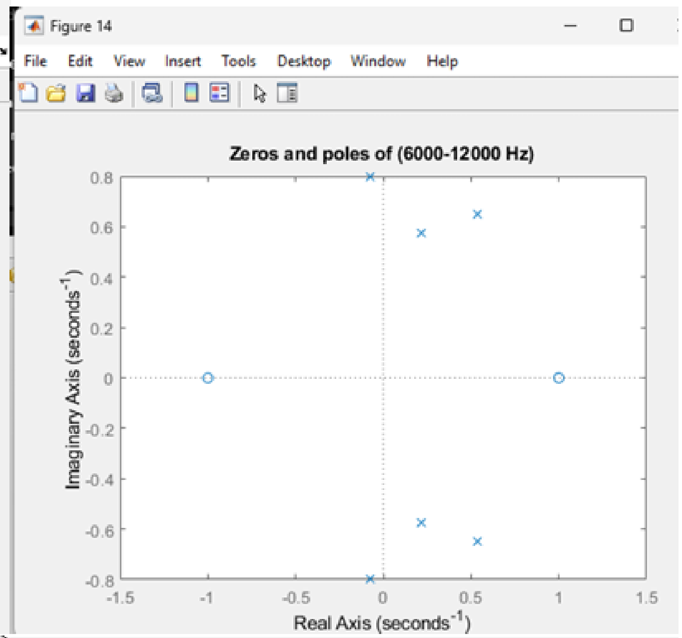


Figure 14
File Edit View Insert Tools Desktop Window Help
Zeros and poles of (6000-12000 Hz)
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

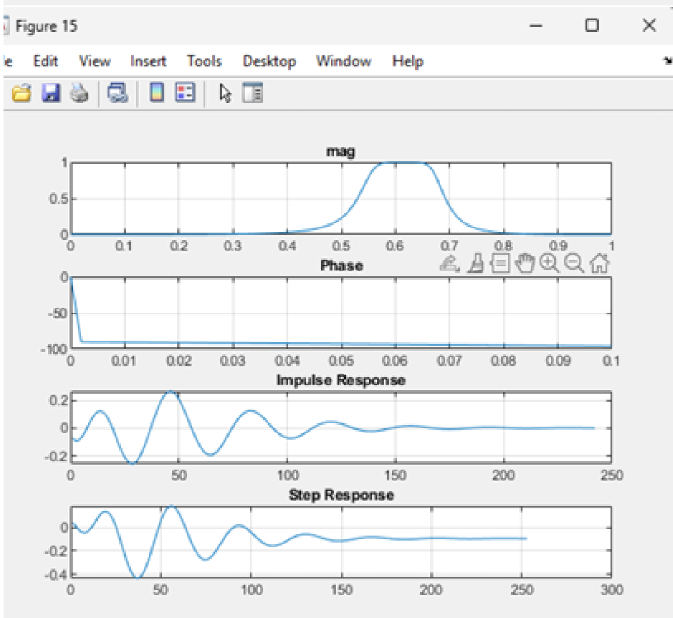


Figure 15
Edit View Insert Tools Desktop Window Help
mag
Phase
Impulse Response
Step Response

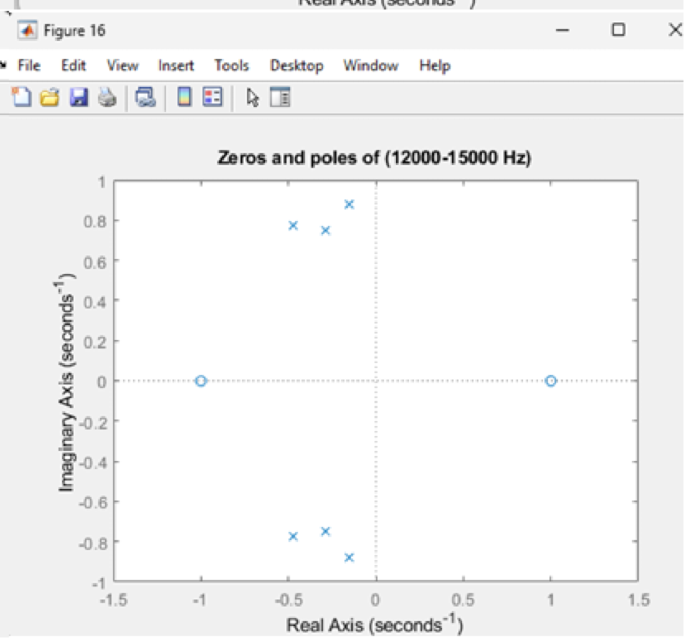
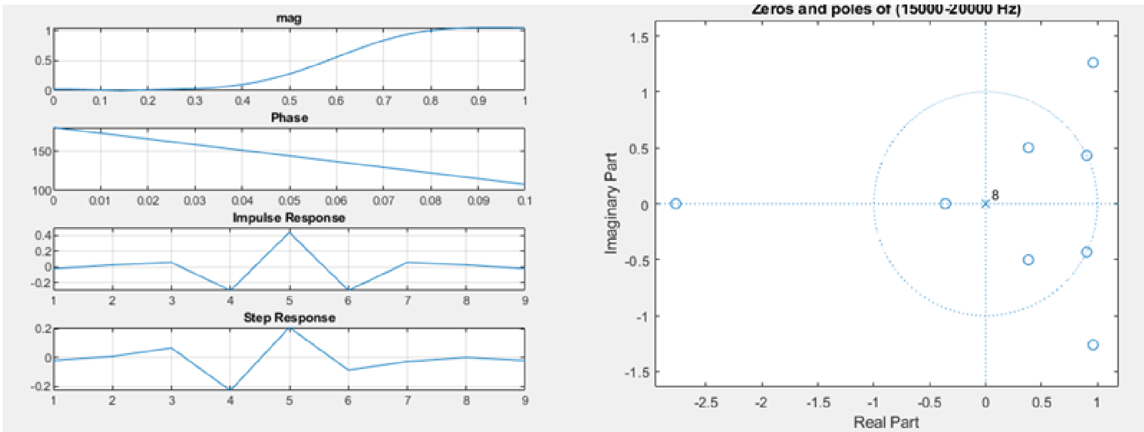


Figure 16
File Edit View Insert Tools Desktop Window Help
Zeros and poles of (12000-15000 Hz)
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})


mag
Phase
Impulse Response
Step Response
Zeros and poles of (15000-20000 Hz)
Imaginary Part
Real Part
8

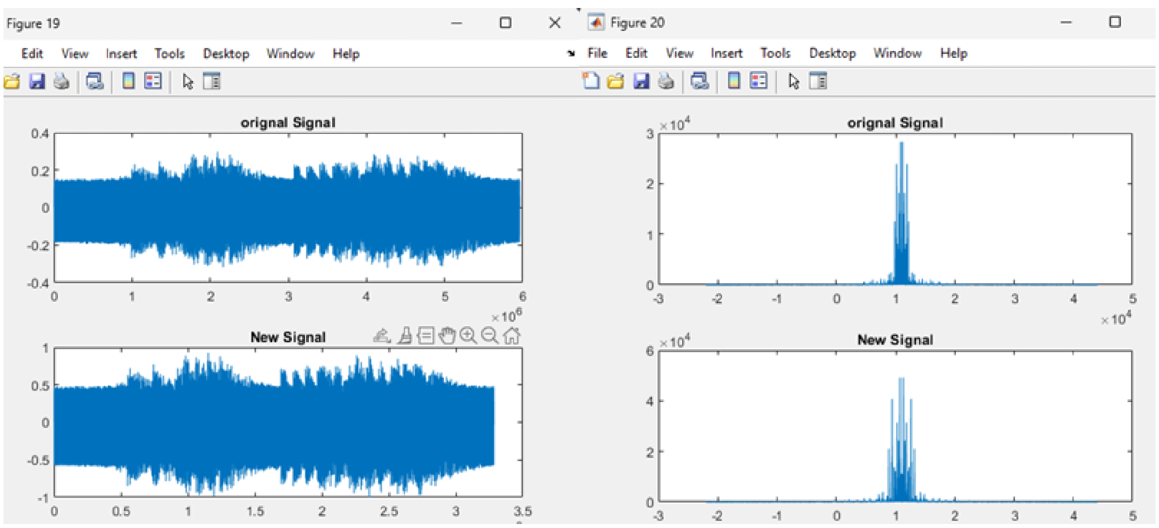
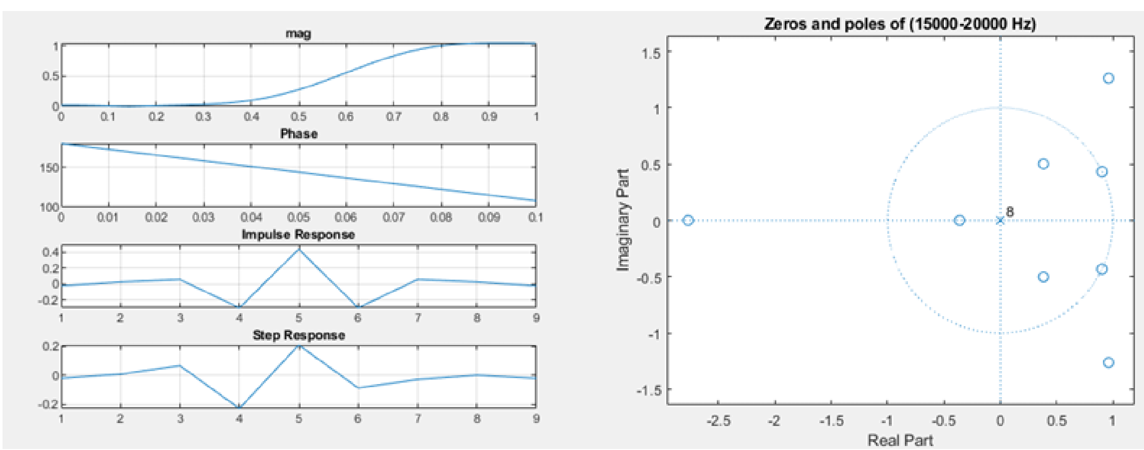
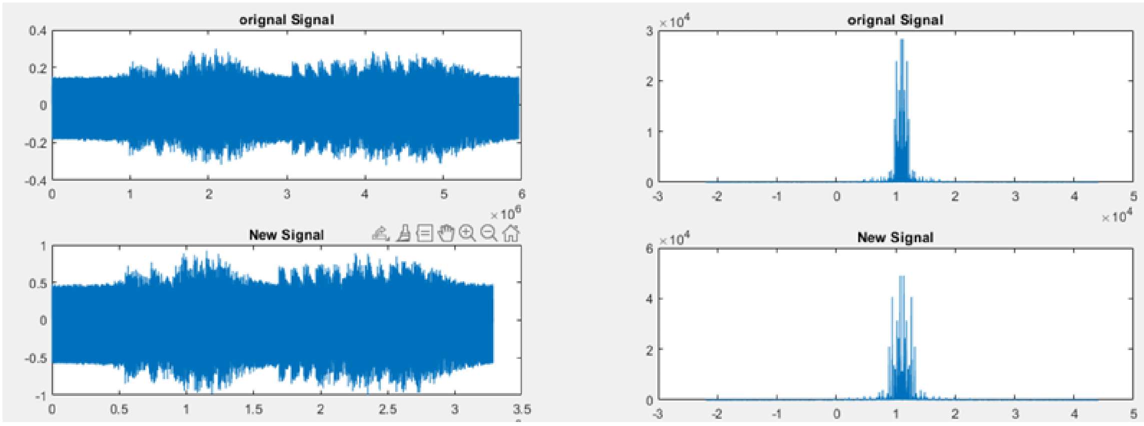


Figure 19
Edit View Insert Tools Desktop Window Help
orignal Signal
New Signal
Figure 20
File Edit View Insert Tools Desktop Window Help
orignal Signal
New Signal


mag
Phase
Impulse Response
Step Response
Zeros and poles of (15000-20000 Hz)
Imaginary Part
Real Part
8



orignal Signal
0.4
0.2
0
-0.2
-0.4
0 1 2 3 4 5 6
×10^6
New Signal
1
0.5
0
-0.5
-1
0 0.5 1 1.5 2 2.5 3 3.5
orignal Signal
3 ×10^4
2
1
0
-3 -2 -1 0 1 2 3 4 5
×10^4
New Signal
6 ×10^4
4
2
0
-3 -2 -1 0 1 2 3 4 5