
```
% Taha Akhlaq MATLAB Assignment 6: Foray and Zee

clc; % clear command window

% 1: dB to magnitude conversion
dB_to_Amplitude = @(dB) 10^(dB/20);

% 2: signal generation and DFT plot
Fs = 44100; % sampling frequency in Hz
N = 88200; % total number of samples
timeVec = (0:N-1) / Fs; % time vector

% frequencies and dB levels
freqList = [14.96e3, 5.52e3, 3.15e3, -3.76e3, 11.01e3];
dBvals = [20, 19, 7, -7, -4];

% construct composite signal
compositeSig = zeros(1, N);
for n = 1:length(freqList)
    compositeSig = compositeSig + ...
        dB_to_Amplitude(dBvals(n)) * exp(1j*2*pi*freqList(n)*timeVec);
end

% white noise at -10 dB
compositeSig = compositeSig + dB_to_Amplitude(-10) * randn(1, N);

% plot the real part of the signal
figure(1);
subplot(2,1,1)
plot(timeVec, real(compositeSig), 'LineWidth', 1)
grid on
xlabel('Time [s]')
ylabel('Amplitude')
title('Real Part of Composite Signal')

% compute and shift FFT
signalFFT = fft(compositeSig);
signalFFTshifted = fftshift(signalFFT);
freqRange = linspace(-Fs/2, Fs/2, N);

subplot(2,1,2)
plot(freqRange, abs(signalFFTshifted), 'LineWidth', 1)
grid on
xlabel('Frequency [Hz]')
ylabel('Magnitude')
title('Magnitude of DFT [Shifted]')

% 3: pole-zero plot and frequency response
```

```

myZeros = [0.55+1i*0.08;
           0.55-1i*0.08;
           0.39+1i*0.96;
           0.39-1i*0.96;
           0.07+1i*0.68;
           0.07-1i*0.68;
           0.08];

myPoles = [-0.06+1i*0.83;
           -0.06-1i*0.83;
           0.30+1i*0.69;
           0.30-1i*0.69;
           0.77+1i*0.61;
           0.77-1i*0.61;
           -0.68;
           -0.34];

overallGain = -0.43;
numCoeffs = overallGain * poly(myZeros);
denCoeffs = poly(myPoles);

% pole-zero diagram
figure(2);
plot(real(myZeros), imag(myZeros), 'o', 'MarkerSize', 8, 'LineWidth', 1)
hold on
plot(real(myPoles), imag(myPoles), 'x', 'MarkerSize', 8, 'LineWidth', 1)
angleCircle = linspace(0, 2*pi, 800);
plot(cos(angleCircle), sin(angleCircle), '--', 'LineWidth', 1)
grid on
axis equal
xlabel('Real Axis')
ylabel('Imag Axis')
title('Pole-Zero Plot')
legend('Zeros', 'Poles', 'Unit Circle', 'Location', 'Best')

% frequency response calculation
fftSize = 1024;
omegaVec = linspace(0, 2*pi, fftSize);
H = zeros(1, fftSize);
for idx = 1:fftSize
    z = exp(1j * omegaVec(idx));
    H(idx) = polyval(numCoeffs, z) / polyval(denCoeffs, z);
end

% convert frequencies to kHz
freqInHz = (omegaVec / (2*pi)) * Fs;
freqInkHz = freqInHz / 1000;

% plot magnitude and phase responses
figure(3);
subplot(2,1,1)
plot(freqInkHz, 20*log10(abs(H)), 'LineWidth', 1)
grid on

```

```

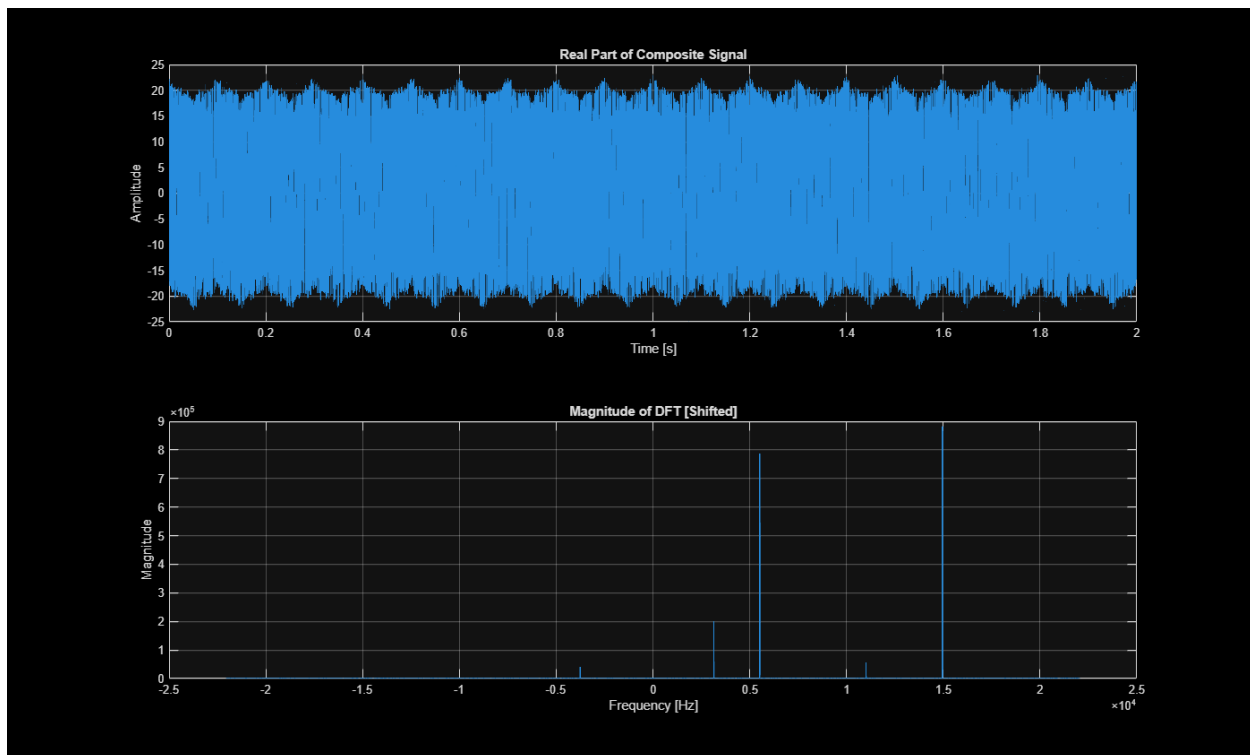
xlabel('Frequency [kHz]')
ylabel('Magnitude [dB]')
title('Magnitude Response')

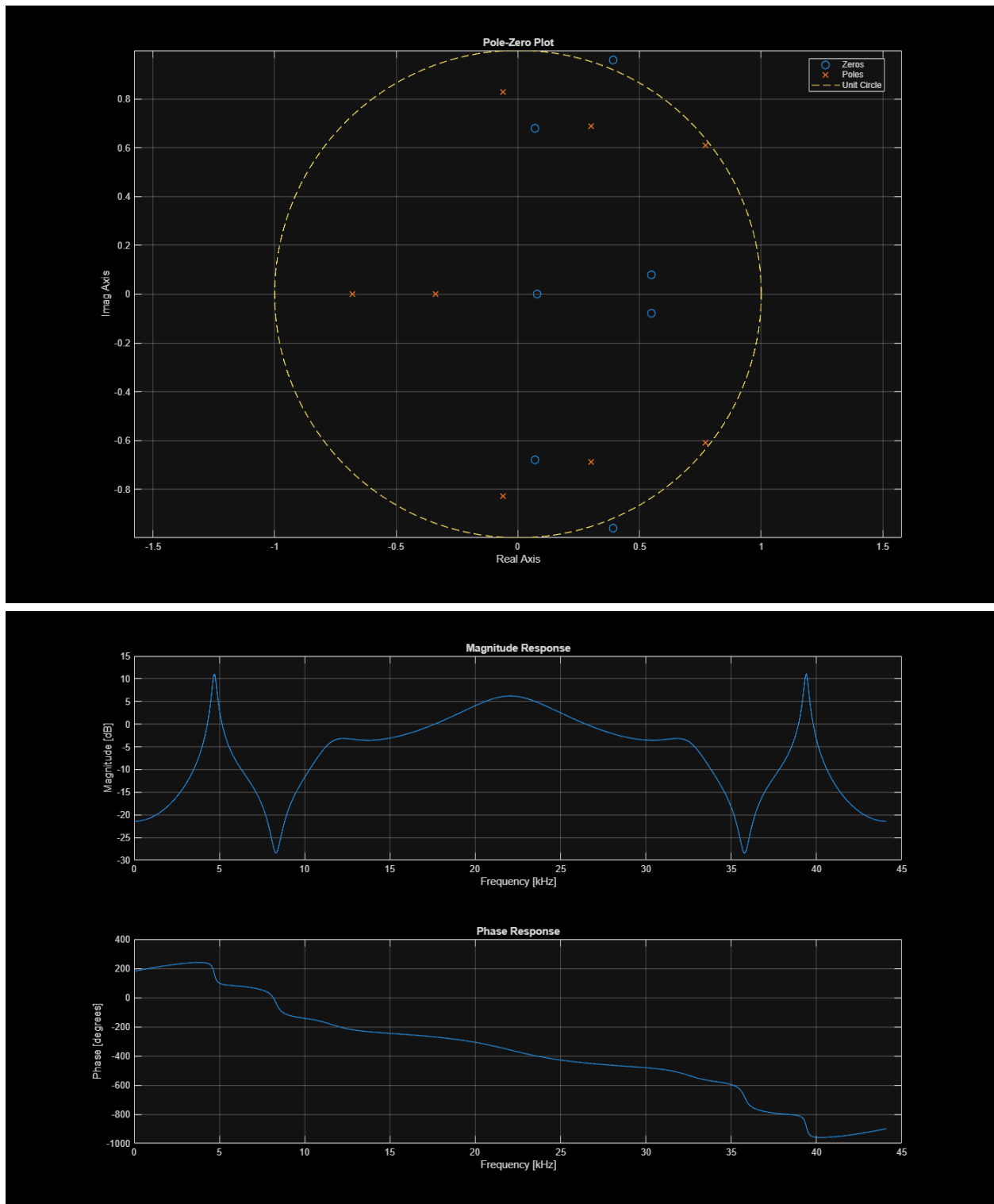
phaseUW = unwrap(angle(H));
phaseDeg = phaseUW * (180/pi);

subplot(2,1,2)
plot(freqInkHz, phaseDeg, 'LineWidth', 1)
grid on
xlabel('Frequency [kHz]')
ylabel('Phase [degrees]')
title('Phase Response')

```

% Output:





Published with MATLAB® R2024b