Systems Theory Laboratory Assignment 5: Frequency response. Bode diagrams

Exercise 1. Consider four systems with the following transfer functions:

$$G_1(s) = \frac{0.1(s+10)}{s+1}, \quad G_2(s) = \frac{10(s+1)}{s+10}, \quad G_3(s) = \frac{10}{s^2+s+1}, \quad G_4(s) = \frac{s^2+s+1}{s^2+s+10}$$

1. Plot the system response for a sinusoidal input, $r(t) = sin(t)$, using the Matlab function *lsim* for a time interval $t \in [0,\ 30]$ sec.
2. For each system, analyze the magnitude and phase angle of the output signal and compare it with the input signal. Determine if the systems have phase lead or phase lag.
3. Draw the Bode diagrams, using the Matlab function *bode* and read from the plots the magnitude and the phase angle for each output signal, when the input is $r(t) = sin(t)$.

Hint *See Lab5_selected_solutions.pdf for the solution with the transfer function* G_4.

Exercise 2. 1. Sketch (on paper) the Bode diagram for the systems with the following transfer functions:

$$G_1(s) = \frac{s^2}{(10s+1)^2}, \quad G_2(s) = \frac{10s+10^4}{s^2+s+1}, \quad G_3(s) = \frac{10^9 s}{(s+1000)(s+10^7)}, \quad G_4(s) = \frac{10s}{s^2+2s+4}$$

2. Determine the frequencies for which the system amplifies or attenuates the sinusoidal input signals.
3. For each system, use the Bode diagram to determine the magnitude of the output signal if the input is:

$$u_1(t) = sin(t), \qquad u_2(t) = 0.1sin(10^{-3}t), \qquad u_3(t) = 3sin(100t).$$

Hint *Use BodeRules.pdf to sketch the plots and see Lab5_selected_solutions.pdf for the solution with the transfer function* G_1. *Check the solutions with* bode *in Matlab.*

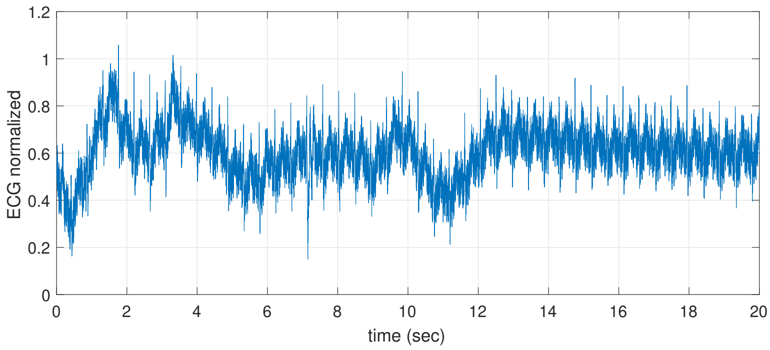
Exercise 3. ECG signal processing problem.

Consider the ECG measurements from [1] shown in Figure 1. The measurements include low frequency disturbances due to breathing and coughing of the patient, and high frequency noise due to the (low quality) sensors.

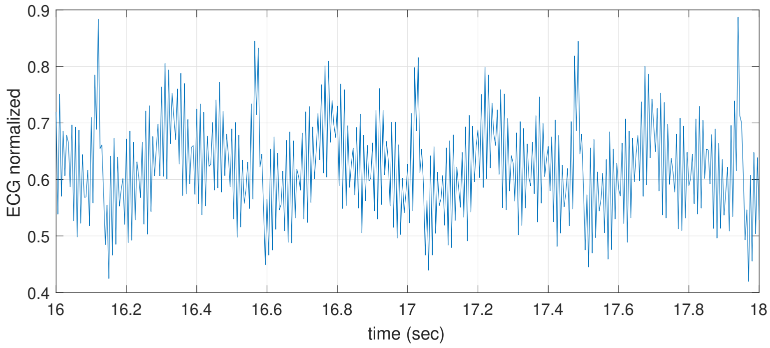
The ideal ECG measurement, taken on a short time span, should look as in Figure 2. The 3 peaks (R, T and P - from the largest to the smallest), along with their timing, are very important for medical diagnosis. For example, the period between the R peaks is used for determining the heart rate (reciprocal of the heart period)[1].

It is obvious that medical doctors cannot use the ECG measurements from Figure 1 for diagnosis purposes. In order to solve this problem, a band-pass filter (see Figure 3) must be designed to remove the low and high frequency artifacts. The frequency of interest for ECG measurements is usually between 0.5 Hz and 100 Hz [1].

[1] For details see also: http://www.medicine.mcgill.ca/physio/vlab/cardio/introecg.htm



a) Long time span



b) Short time span

Figure 1: ECG measurements

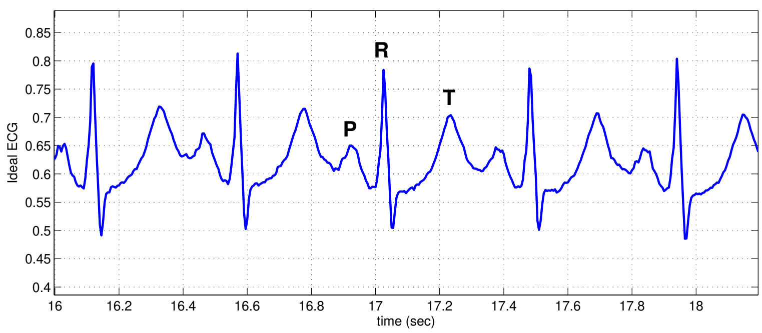


Figure 2: Ideal ECG measurements (short time span)

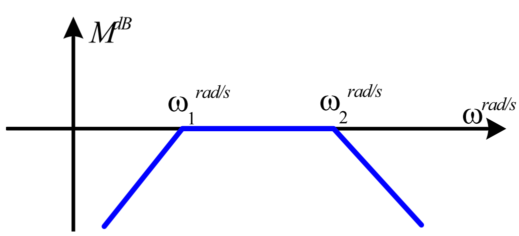


Figure 3: Magnitude plot of a band-pass filter

The goal of this application is to design a band-pass filter for removing the low and high frequency components and to compare it with a commonly used filter (Butterworth filter). The cut-off frequencies of the filters will be $f_1 = 0.5$Hz and $f_2 = 50$Hz.

1. Read data from *ECGdata.txt* file. The example code is given in Listing 2.

Listing 1: readECGdata.m

```
close all
clear all
clc
% ----- read data from ECGdata.txt ------
fileID = fopen('ECGdata.txt', 'r');       % open ECGdata.txt for read access
A = fscanf(fileID, '%f %f',[2,Inf]);      % read data from file in array A
fclose(fileID)                            % close ECGdata.txt
time = A(1,:);                            % save first row of data in variable "time"
necg = A(2,:);                            % save second row of data in variable "necg"

figure, plot(time, necg), grid on         % plot necg versus time
xlabel('t (sec)'), ylabel('ECG normalized')
```

   The data is organized now in two row vectors: *time* with 4000 values of time between 0 and 20 seconds, and *necg* with the normalized values of the ECG signal for all moments of time. Plot *necg* versus *time* and the figure will be similar to Figure 1 (a).

2. Design a Butterworth filter using the function *butter* with the following specifications:
   - Filter type: analog bandpass
   - Filter order: 8
   - Filter cutoff frequencies: $\omega_1 = 0.5 * 2 * \pi$ rad/s and $\omega_2 = 50 * 2 * \pi$ rad/s;

Listing 2: filterECGdata.m

```
% ------  Butterworth filter -------
[num, den] = butter(4, [0.5*2*pi 50*2*pi], 'bandpass', 's');
butter_filter = tf(num,den);                   % create the filter


% ------ Bode plot of the filter -------
figure, bode(butter_filter), grid on

% ---    filter the signal -------
        % system = butter_filter
        % input: necg = the noisy signal
        % time: time
 figure, lsim(butter_filter, necg, time), grid on
```

   The code example is given in Listing 2.

   Obs. The first input argument in function butter is $n = 4$, but a band-pass filter will have the order $2n = 8$. The function butter returns the numerator and denominator polynomials of the filter transfer function.

3. Plot the Bode diagram of the filter using the *bode* function (see Listing 2).

4. Determine the filtered ECG signal by using the *lsim* function, with the filter transfer function and the ECG signal as input (see Figure 4).

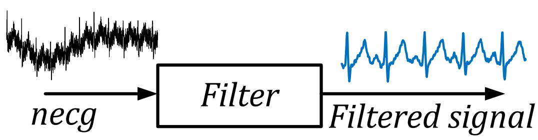


Figure 4: Filter

   The function *lsim* will plot the input ECG signal (*necg*) and the filtered ECG signal on the same figure. Can you read the three peaks on the plot? Notice the differences between the two signals:

- on the entire time interval,
- on a time interval between 16 and 18 sec,
- on a time interval between 2 and 4 sec.

5. Design an 8-th order bandpass filter using first-order elements, as close as possible to the previous Butterworth filter.

   (i) First try to find the right combination of elements in order to obtain a bandpass Bode diagram (Figure 5) with the same cutoff frequencies as for the Butterworth filter and the slopes of the lines are +20dB/dec and −20 dB/dec.

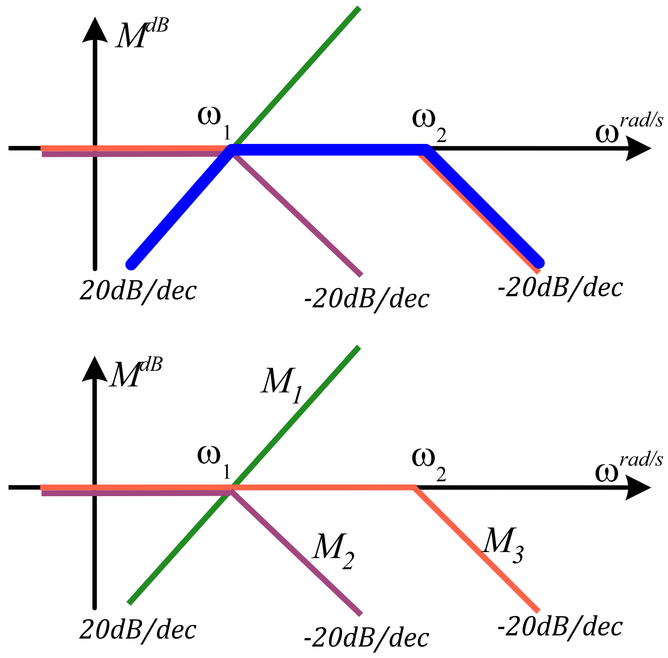


Figure 5: Bandpass filter decomposed

   (ii) Determine the parameters (gain and time constants of the filter transfer function) that give the desired cutoff frequencies. Obtain a second-order transfer function: $G_2(s)$.

```
G2 = tf([.....],[.......])
```

   (iii) Multiply the previously obtained filter transfer function by itself iteratively and notice how the Bode diagram changes (use Matlab) through each iteration. A 4 time multiplication should provide a Bode diagram similar to the Butterworth filter. Plot both Bode diagrams on the same figure and compare them.

```
G4 = G2*G2;                          % obtain a 4-th order filter
G8 = G2*G2*G2*G2;                    % obtain an 8-th order filter
figure, bode(G2, G4, G8), grid on    % plot on the same figure
```

6. Plot the ECG signal, the filtered ECG signal with the Butterworth filter, and the filtered ECG signal with the custom filter (from point (iii)) and compare the results (zoom in).

```
figure, lsim(butter_filter, G8, necg, time), grid on
```

Optional problem

Exercise 4. Consider a second-order system having the transfer function:

$$G(s) = \frac{K}{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1} \tag{1}$$

Several input signals have been applied to the system and the outputs have been recorded. The inputs are sinusoidal signals $r(t) = \sin \omega t$, where the frequency is $\omega \in \{1,\ 2,\ 5,\ 8,\ 9.5,\ 10,\ 20,\ 100\}$ rad/sec. The outputs, for each input frequency, are shown in Figure 6.

Using the frequency response, determine the system parameters: K, ω_n and ζ, by completing the following steps:

1. Sketch the Bode diagram for the transfer function given by (1).
2. Obtain an experimental Bode magnitude plot using the output signals from Figure 6, as follows:
   - Read the magnitude of the output signal at steady-state and divide it to the magnitude of the input. Save all numbers in an array $\mathbf{M} = [M_1,\ \ldots,\ M_8]$.
   - Convert the numbers from the array $\mathbf{M}$ into decibels and plot them versus all values of ω on a logarithmic scale (use the Matlab function *semilogx*).
3. Compare the Bode diagrams obtained at 1 and 2 and determine the system parameters from the plots.

References

[1] Rangaraj M. Rangayyan, *Biomedical Signal Analysis*, Wiley IEEE Press, 2001.

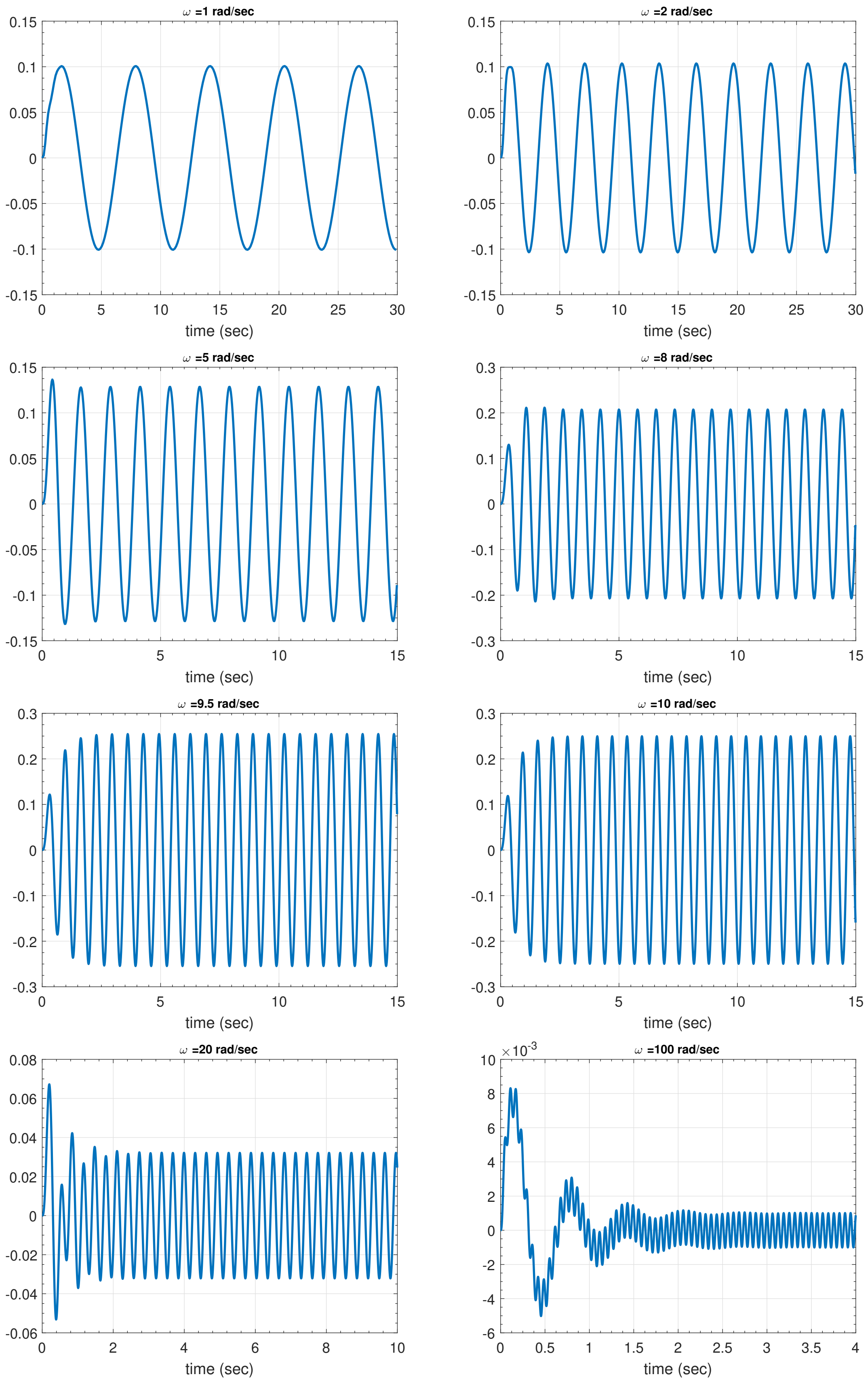


Figure 6: Output signals for various input frequencies