**ENEL102, fall term 2017**

**Assignment 2**

**Writing Matlab Functions Chapter 7**

**Due October 2**

Assignment questions are based on material in the Gilat textbook from chapter 7. Suggest you review this chapter before answering these questions. Fill in the following template with your answers using Matlab plots and screen shots as necessary. Then submit your Word document on D2L.

………………………………………………………………………………………………………………………………..

**Q1.** Write an **anonymous function** , that can calculate the following sum:



Test your function with . (hint – your anonymous function needs to be one command line starting with f = @(x,y)… )   
………………………………………………………………………………………………………………………

**(Matlab input)**

>> syms n

>> f = @(x,y) symsum(sin(x^n)+3\*y^n,n,1,5);

>> z = f(1,2)

**(Matlab Response)**z =

5\*sin(1) + 186

………………………………………………………………………………………………………………………………..

**Q2.** Use the anonymous function of **Q1** to generate a plot of z=f(x,y) over the range of



Use 400 points for the x variable. Label the axis of your graph.

**(Matlab input)**

syms n

f = @(x,y) symsum(sin(x^n)+3\*y^n,n,1,5);

x = linspace(-1,1,400);

y = 0.5;

z = zeros(1,400);

for i=1:1:400

z(i) = f(x(i),y);

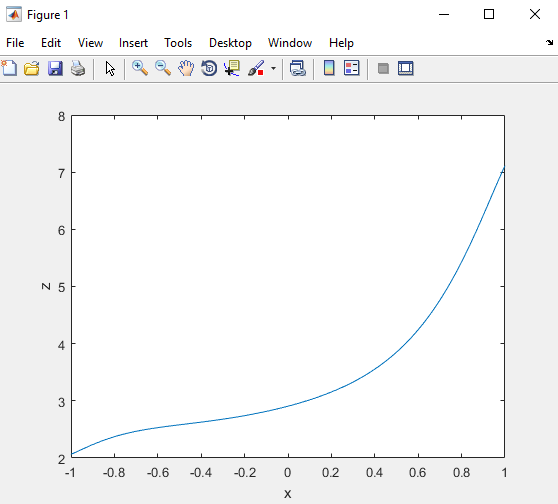
end

plot(x,z)

xlabel('x');

ylabel('z');

**(Matlab Response)**



**Q3.** Consider the analog second order band pass filter consisting of a capacitor C, inductor L and a resistor R in series as shown in the figure.



The transfer function is given as



where



Write a function that determines the magnitude of the frequency response of the filter. That is we want a function that determines



The function must be written such that it accepts a vector of frequencies for  and input parameters of R, L and C.

**(Matlab input)**

function [f] = a2q3\_frequency\_responce(R, L, C, w)

f = abs(R./(R +j.\*w.\*L + 1./(j.\*w.\*C)));

end

………………………………………………………………………………………………………………………………..

**Q4.** Write a function that computes the frequency response of the bandpass filter with L=1, C=1 and R=10 for a range of using a linear frequency scale. Then plot for  using a log frequency plot.

**(Matlab input)**

% Within Function:

function [freq] = a2q4\_frequency\_responce(w)

freq=abs(10./(10+1i.\*w+(1./(1i.\*w))));

end

%In Command Line:

>> w1 = linspace(0,10,100);

>> w2 = linspace(0.01,100,100);

>> frequency1 = a2q4\_frequency\_responce(w1);

>> frequency2= a2q4\_frequency\_responce(w2);

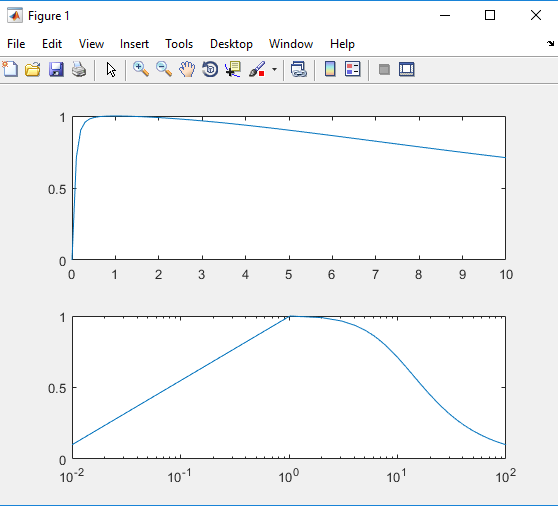
>> subplot(2,1,1)

>> plot(w1,frequency1);

>> subplot(2,1,2)

>> semilogx(w2,frequency2);

**(Matlab Response)**



………………………………………………………………………………………………………………………………..

**Q5.** Write a function for the bandpass filter of Q3 that determines the phase of the transfer function in degrees. Generate a plot of the phase shift of the bandpass filter for the frequency range of using a log frequency plot.

**(Matlab input)**

%in function

function [phase] = a2q5\_phase(R,L,C,w)

transfer = (R./(R+1i.\*w.\*L+(1./(1i.\*w.\*C))));

r = real(H);

im = imag(transfer);

phase=rad2deg(atan(im./r));

end

%in command line

>> w = linspace(.01, 100, 1000);

>> semilogx(w, a2q5\_phase(1, 1, 1, w))

>> grid on

**(Matlab Response)**

………………………………………………………………………………………………………………………………..

**Q6.** Modify the program in Q5 such that the parameters of R C and L are global variables and hence do not have to be passed to the function filtfreq(). List your modified Matlab code (both the function and the main program). Plot the phase response for the case of L=1, C=1, R=0.1 and .

**(Matlab input)**

% in function

function [phase] = a2q6\_global\_phase(w)

global R;

global L;

global C;

transfer = (R./(R+1i.\*w.\*L+(1./(1i.\*w.\*C))));

r=real(transfer);

im=imag(transfer);

phase=rad2deg(atan(im./r));

end

%in main file

global R;

global L;

global C;

R=0.1;

L=1;

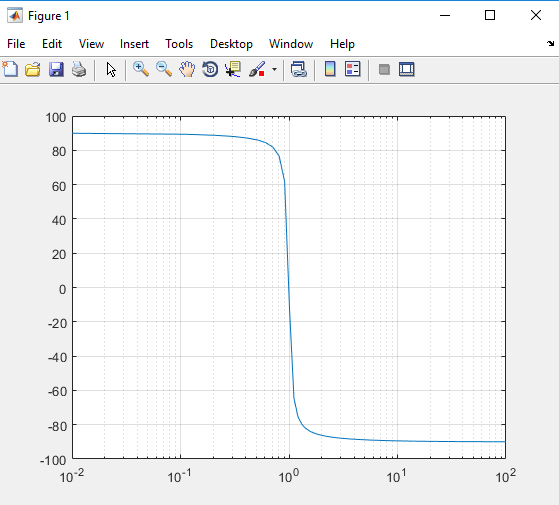
C=1;

w=linspace(0.01,100,1000);

semilogx(w,a2q6\_global\_phase(w))

grid on

**(Matlab Response)**



………………………………………………………………………………………………………………………………..

**Q7.** Give a reason why global variables are useful. Then explain why the excessive use of global variables is generally considered to be bad programming.

**(ans)**

One useful of global variables is that they are stored in memory, so they will not fill up a functions stack memory when the function is called. They also make calling functions simpler to understand and code, because the designer must write less and the reader can understand the use of global variables.

It can be considered bad programing to use many of them because if all variables are global the functions may run slower. Also, it can be confusing to a programmer and it requires extra code to define variables as global in matlab.

………………………………………………………………………………………………………………………………..

**Q8**. The signal time delay through the filter is given by the rate of change of the phase shift of the filter with respect to excitation frequency. That is, the filter delay, denoted by D, is given as



Write a function that will determine D for a set of parameters L, C, R and. Use a numerical means of approximating the derivative. Hint – evaluate  for two closely spaced frequencies and then determine the change in angle from this. That is compute the slope from the ‘rise over run’.

Evaluate the time delay for R=1, C=1, L=1 and .

**(Matlab input)**

% in function

function [delay] = a2q8\_time\_delay(L,R,C,w)

w1 = w + 0.0001;

frequency1 = (R./(R+1i.\*w.\*L+(1./(1i.\*w.\*C))));

frequency2 = (R./(R+1i.\*w1.\*L+(1./(1i.\*w1.\*C))));

r1 = real(frequency1);

im1 = imag(frequency1);

phase1 = (atan(im1./r1));

r2 = real(frequency2);

im2 = imag(frequency2);

phase2 = (atan(im2./r2));

delay = (phase1 - phase2)/(w1-w);

end

% in command line

>> a2q8\_time\_delay(1, 1, 1, 1)

**(Matlab Response)**

ans =

1.9999

………………………………………………………………………………………………………………………………..

**Q9**. The resonance frequency of the bandpass filter considered in the previous questions is given by



Write a program the calculates the delay of the filter as determined in Q8, at the resonance frequency, as a function of the resistance R over the range of 0.1>R>10 and plots this delay. Use L=1 and C=1 such that D is evaluated at .

**(Matlab input)**

**(Matlab Response)**