Wasam Al-bayti

101041465

Assignment 4

Question 1 and 2

A) The code for this part is shown below:

```
% The parameters
R1 = 1;
R2 = 2;
c = 0.25;
R3 = 10; %% R3 is 10 ohms
L = 0.2;
R4 = 0.1;
RO = 1000;

% The matrices for C and G, it is easier to do each row and fill with zeros instead of writing out
the zeros
G=zeros(6);
C=zeros(6);
G(1,:)=[1 0 0 0 0 0];
C(1,:)=[0 0 0 0 0 0];
G(2,:)=[(-1/R1) (1/R2+1/R1) 0 0 0 1];
C(2,:)=[-c +c 0 0 0 0];
G(3,:)=[0 0 1/R3 0 0 -1];
C(3,:)=[0 0 0 0 0 0];
G(4,:)=[0 0 -1*100/R3 1 0 0];
C(4,:)=[0 0 0 0 0 0];
G(5,:)=[0 0 0 -1/R4 (1/R4+1/RO) 0];
C(5,:)=[0 0 0 0 0 0];
G(6,:)=[0 -1 1 0 0 0];
C(6,:)=[0 0 0 0 0 L];


matrix=zeros(3,20);
matrix(1,:)=linspace(-10,10,20);
for i=1:20
   F=[matrix(1,i);
      0;
      0;
      0;
      0;
```

```
      0];
   V=G\F;
   matrix(2,i)=V(5);
   matrix(3,i)=V(3);
end

figure(1)
subplot(2,1,1);
plot(matrix(1,:),matrix(2,:));
title('Voltage as a Function of Input Voltage VO');
ylabel('Voltage (V)');
xlabel('V input (V)');
subplot(2,1,2);
plot(matrix(1,:),matrix(3,:));
title('Voltage as a Function of Input Voltage V3');
ylabel('Voltage (V)');
xlabel('V input (V)');



matrix2=zeros(2,2000);
matrix2(1,:)=linspace(0,500,2000);
Vin=1;
for i=1:2000
   F=[Vin;
      0;
      0;
      0;
      0;
      0];
   V=(G+1j*matrix2(1,i)*C)\F;
   matrix2(2,i)=V(5);
end

figure(2)
plot(matrix2(1,:),real(matrix2(2,:)));
title('Output Voltage as a Function of Angular Frequency');
ylabel('Output voltage');
xlabel('omega');

figure(3)
semilogx(matrix2(1,:),20*log10(real(matrix2(2,:))./Vin));
title('Gain as a Function of Angular Frequency');
ylabel('Gain (dB)')
```

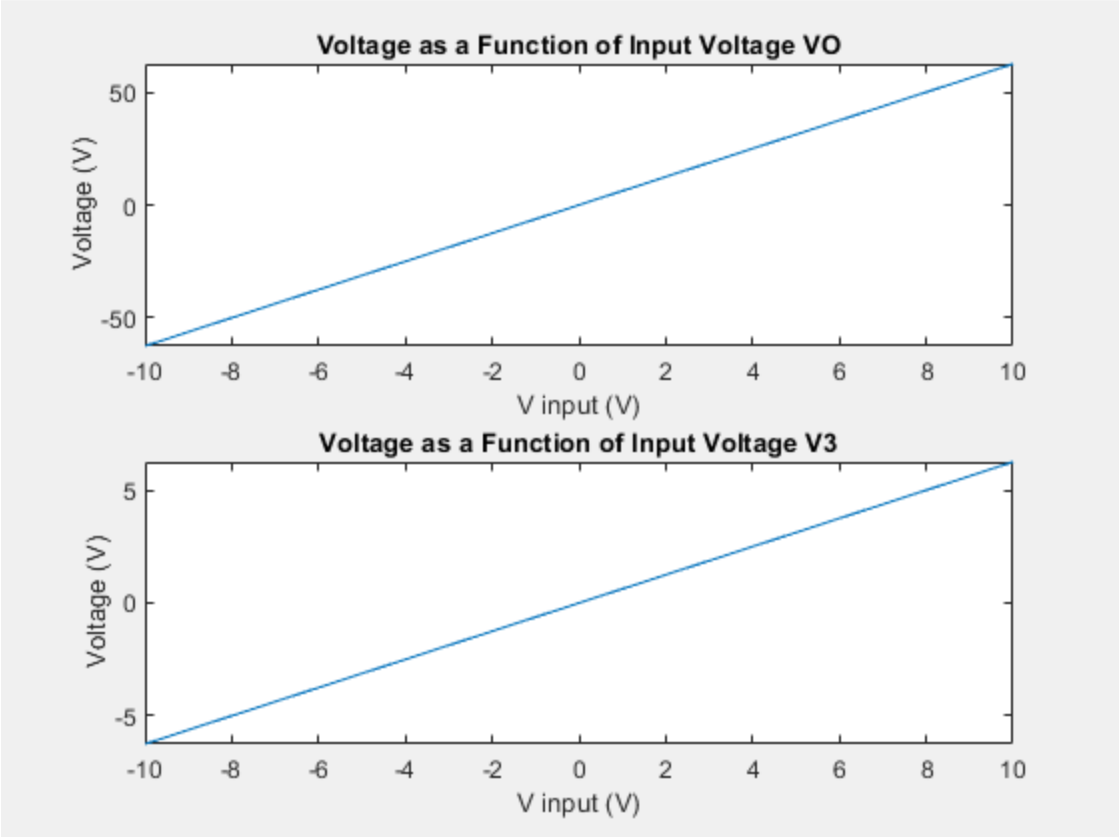
```
xlabel('omega');


matrix3=zeros(1,10000);
for i=1:10000
   C(2,:)=[-(c+randn()*0.05) c+randn()*0.05 0 0 0 0];
   F=[Vin;
      0;
      0;
      0;
      0;
      0];
   V=(G+1j*pi*C)\F;
   matrix3(1,i)=V(5);
end

figure(4);
hist(real(matrix3),50);
title('Gain for perturbations in C');
ylabel('Counter')
xlabel('Gain dB');

C(2,:)=[-c +c 0 0 0 0];
stepsize=1/1000;
freq=(-(1000)/2:(1000)/2)*(1/1/1000/1000+1);
matrix3=zeros(3,3,1000+1);
oldVoltage=[0; 0; 0; 0; 0; 0];
input_sim=zeros(3,1000+1);
for type=1:3
   for i=1:1000
      if(type==1)
         if(i*stepsize<0.03)
            input_sim(type,i)=0;
         else
            input_sim(type,i)=1;
         end
      elseif(type==2)
         input_sim(type,i)=sin(2*pi*1/0.03*i*stepsize);
      else
         input_sim(type,i)=exp(-(i*stepsize-0.1)^2/(2*0.03^2));
      end
   end
end
```

```
for type=1:3
   oldVoltage=[0; 0; 0; 0; 0; 0];
   for i=1:1000
      F=[input_sim(type,i); 0; 0; 0; 0; 0];
      matrix3(type,1,i+1)=i*stepsize;
      matrix3(type,2,i+1)=input_sim(type,i);
      V=(C/stepsize+G)\(C*oldVoltage/stepsize+F);
      matrix3(type,3,i+1)=V(5);
      oldVoltage=V;
   end

end
```

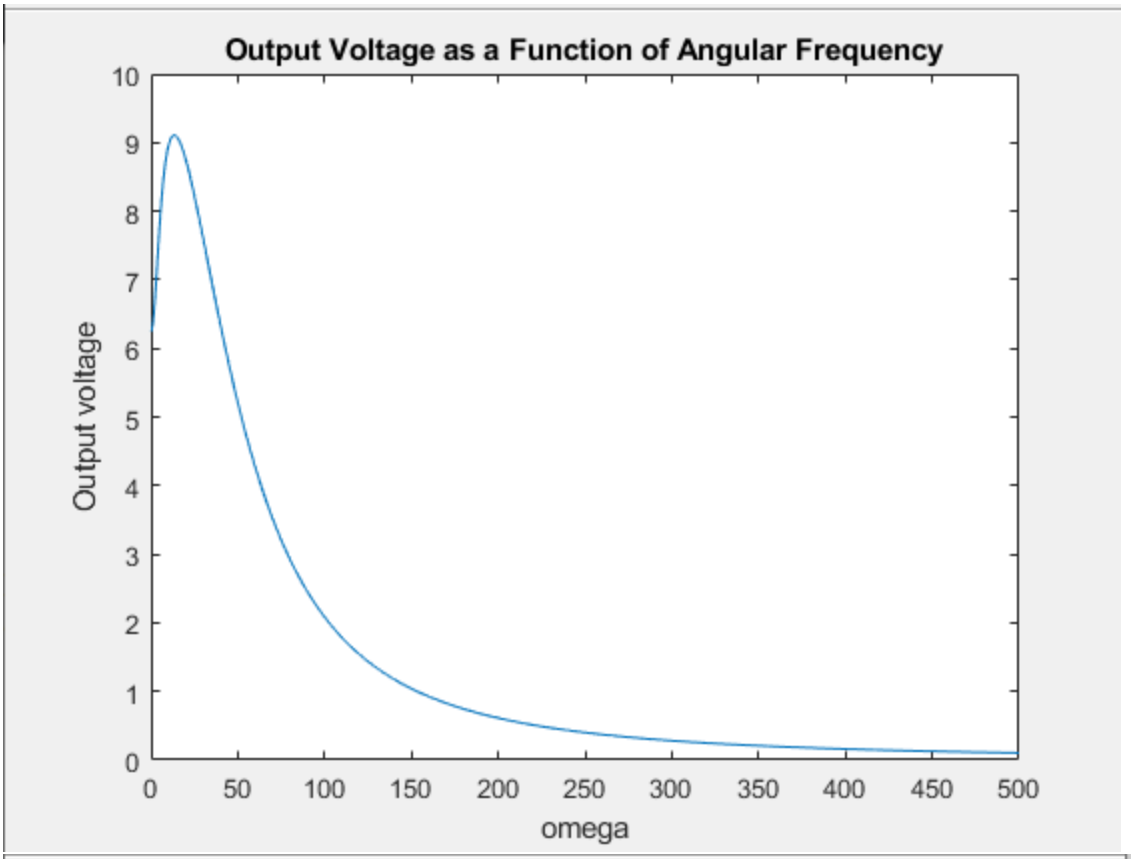
The matrices with their real values

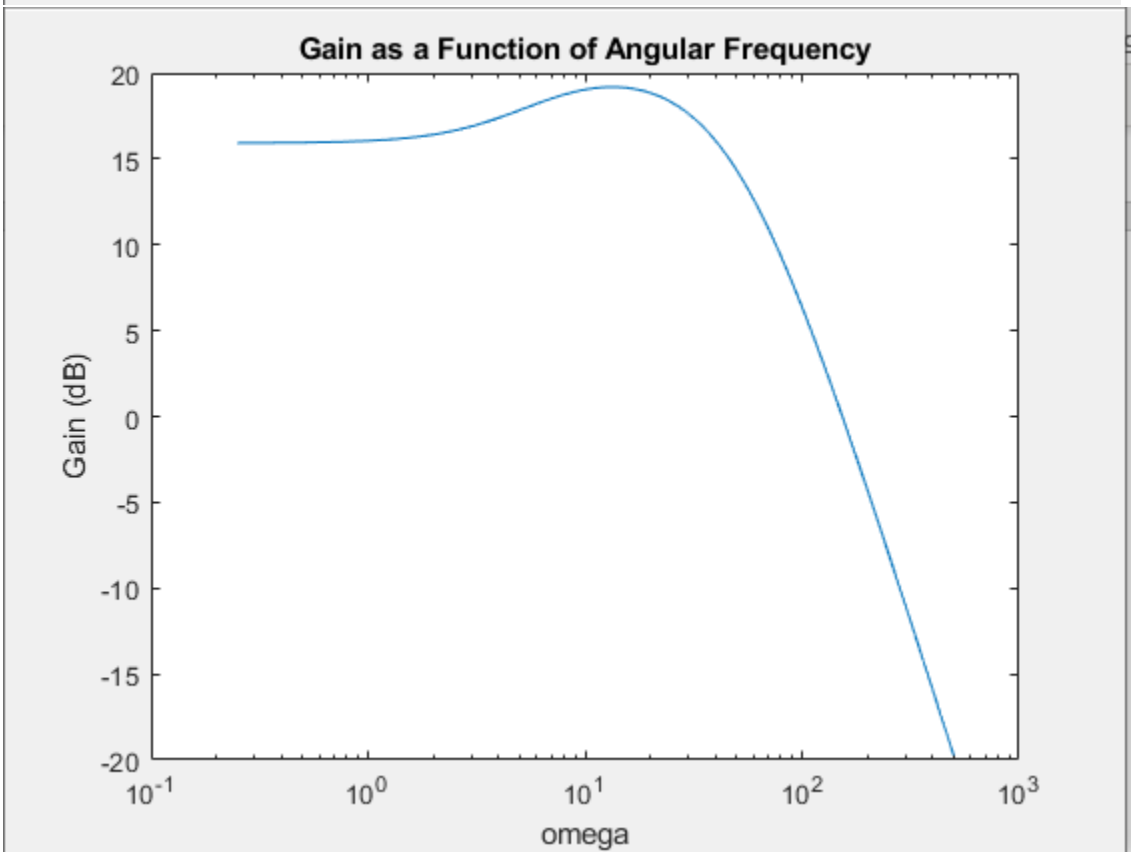
```
C =

         0         0         0         0         0         0
   -0.2500    0.2500         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0    0.2000


G =

    1.0000         0         0         0         0         0
   -1.0000    1.5000         0         0         0    1.0000
         0         0    0.1000         0         0   -1.0000
         0         0  -10.0000    1.0000         0         0
         0         0         0  -10.0000   10.0010         0
         0    1.0000    1.0000         0         0         0
```

b)

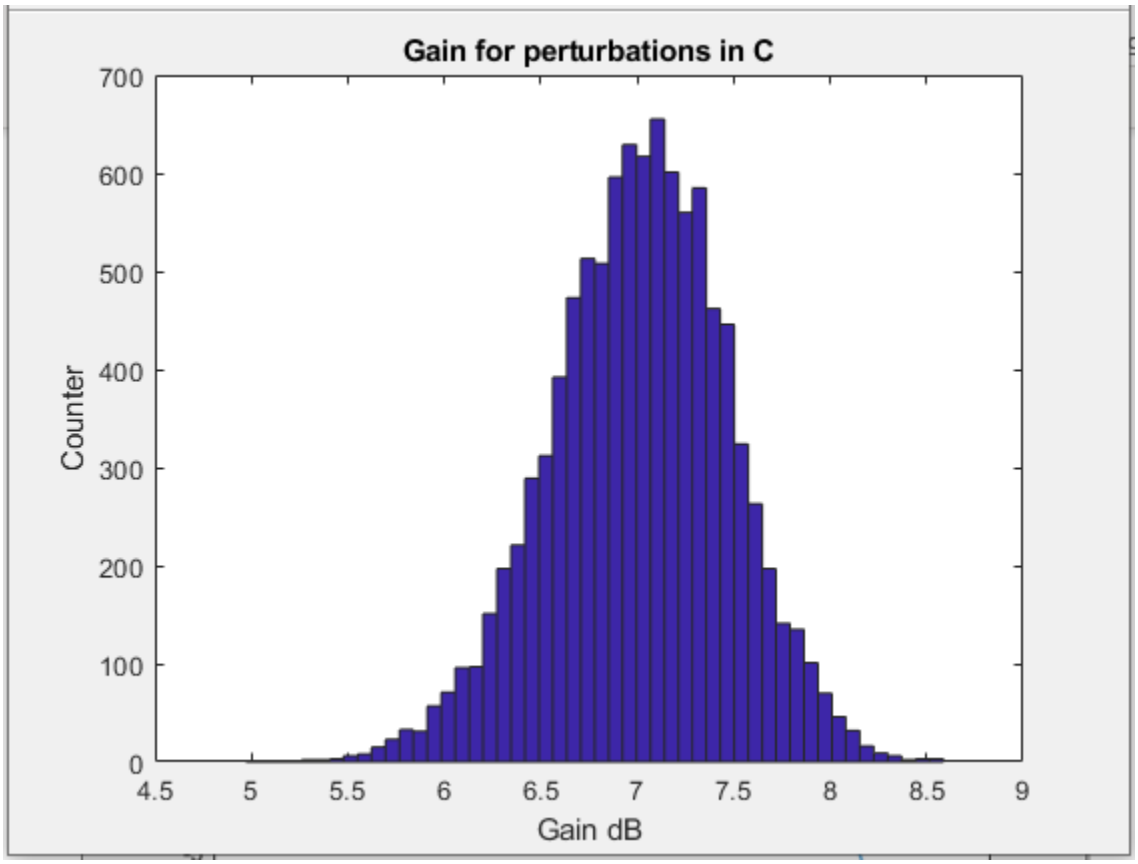

Voltage as a Function of Input Voltage VO
Voltage (V)
50
0
-50
-10
-8
-6
-4
-2
0
2
4
6
8
10
V input (V)
Voltage as a Function of Input Voltage V3
Voltage (V)
5
0
-5
-10
-8
-6
-4
-2
0
2
4
6
8
10
V input (V)

c)


Output Voltage as a Function of Angular Frequency
Output voltage
omega
0
1
2
3
4
5
6
7
8
9
10
0
50
100
150
200
250
300
350
400
450
500

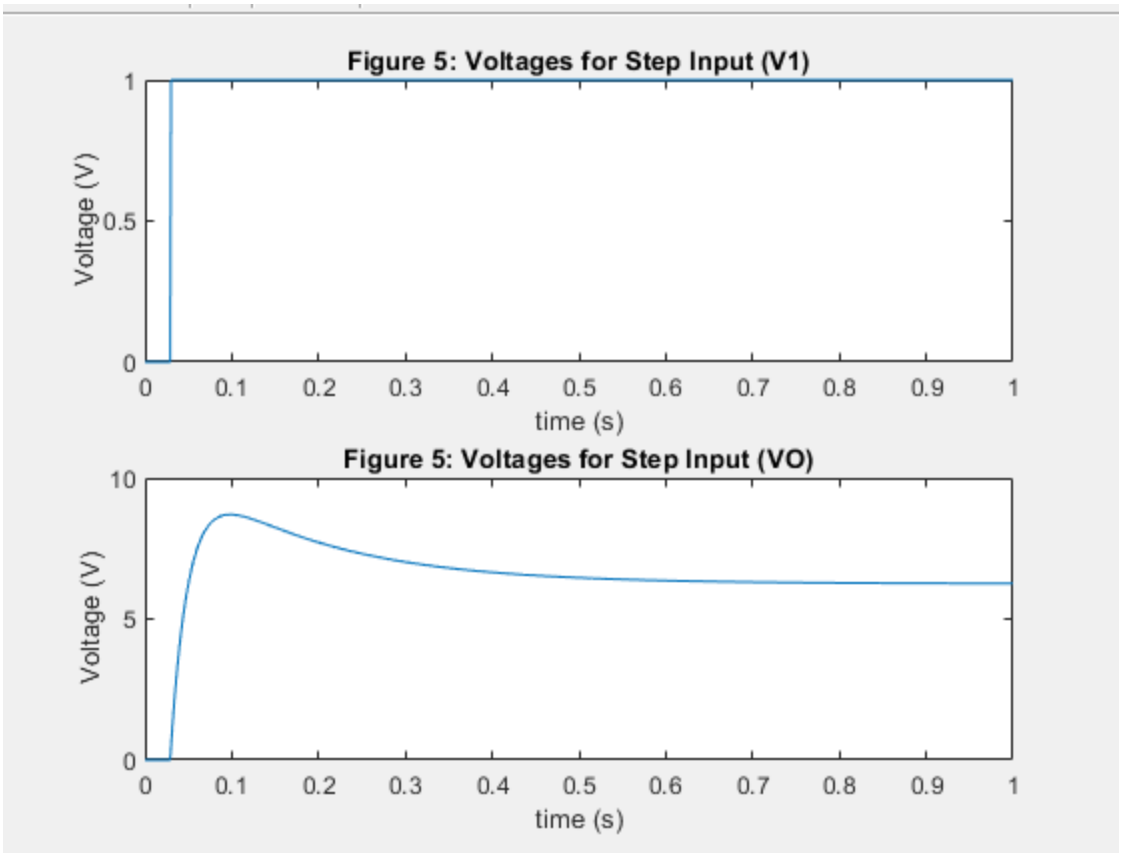

Gain as a Function of Angular Frequency
Gain (dB)
omega
20
15
10
5
0
-5
-10
-15
-20
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}

This circuit can be seen to behave as a low pass filter from the above plots.

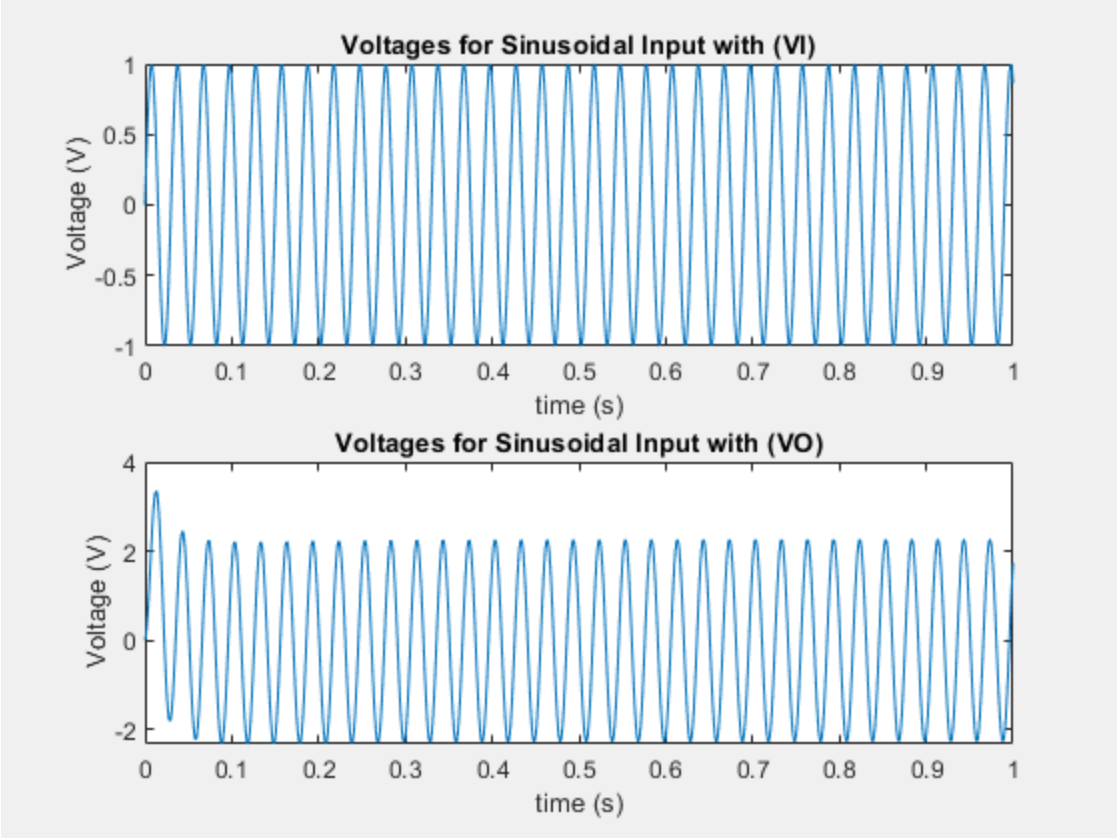


The histogram above shows the gain for perturbations in C and it is seen that it follows a normal distribution.

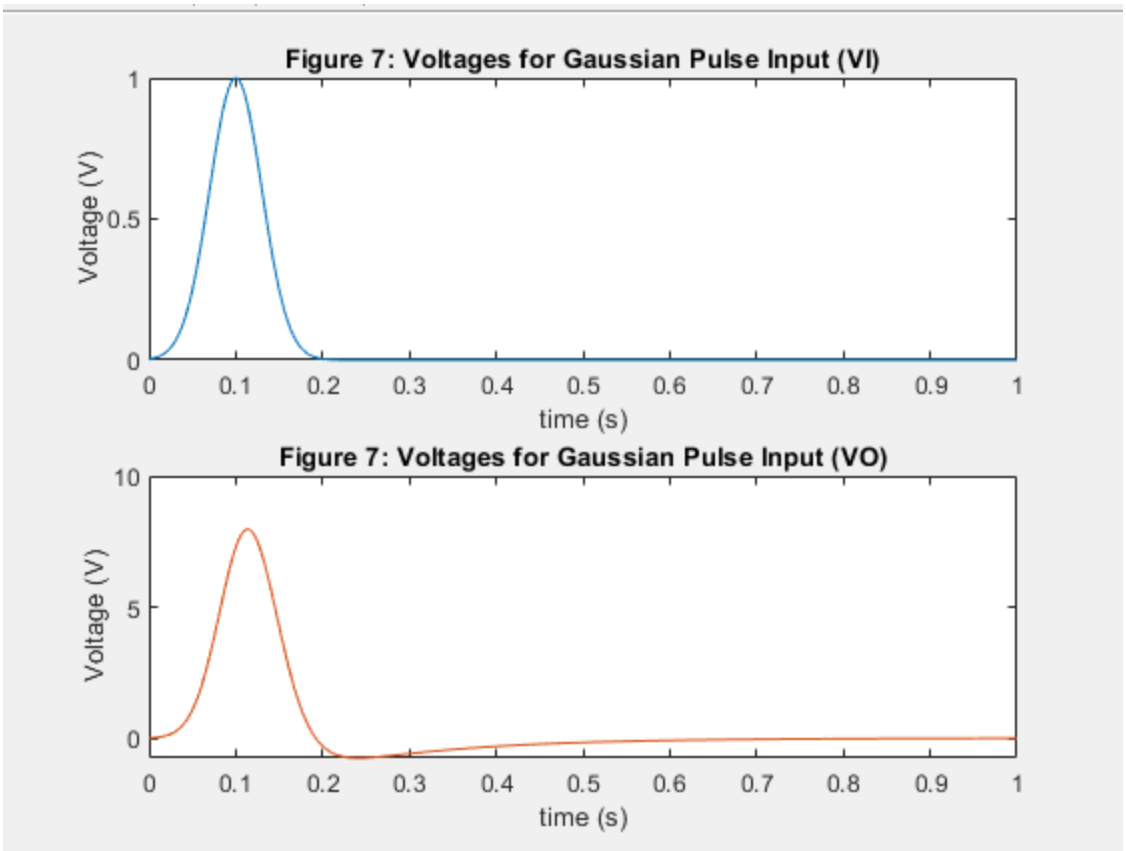
d)



The plot shows the input and output voltage for the step unit, the output voltage overshoots than stabilizes at around 0.3 seconds.

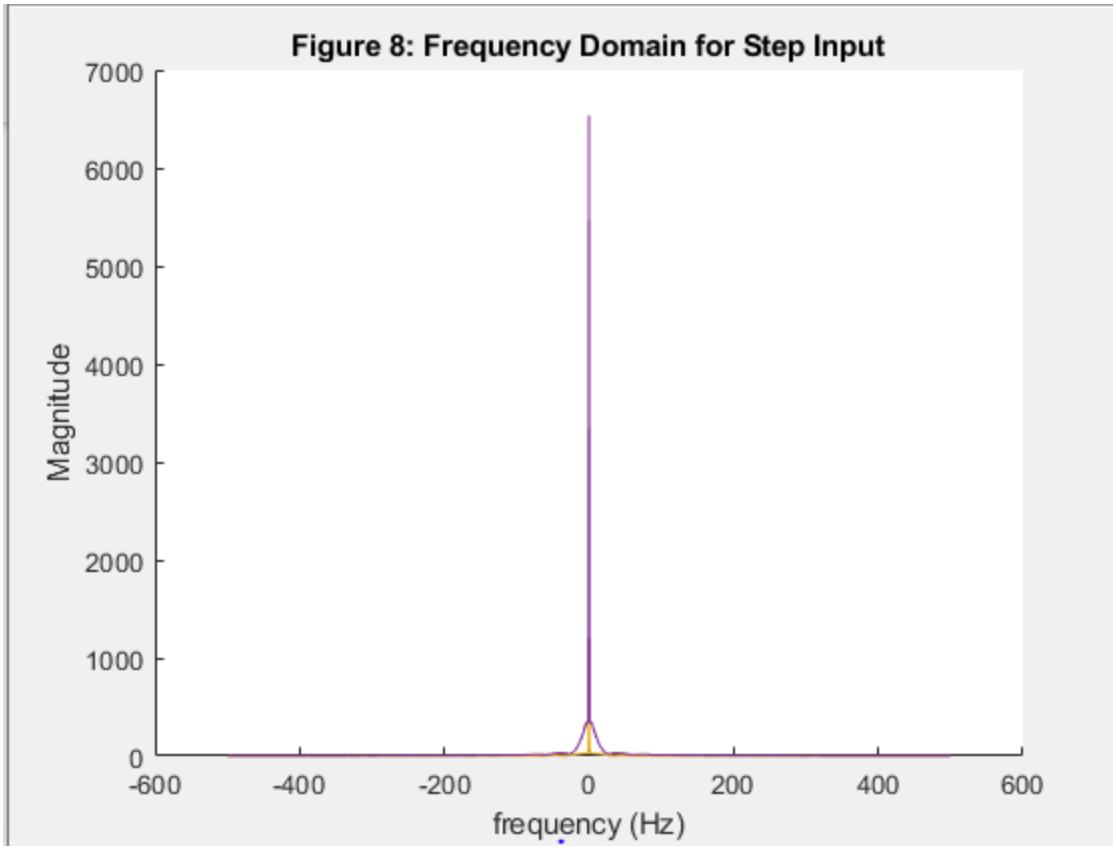


The figure above shows the input and output for a sinusoidal input. The output signal is seen to have a higher amplitude and higher phase shift when compared to the input.

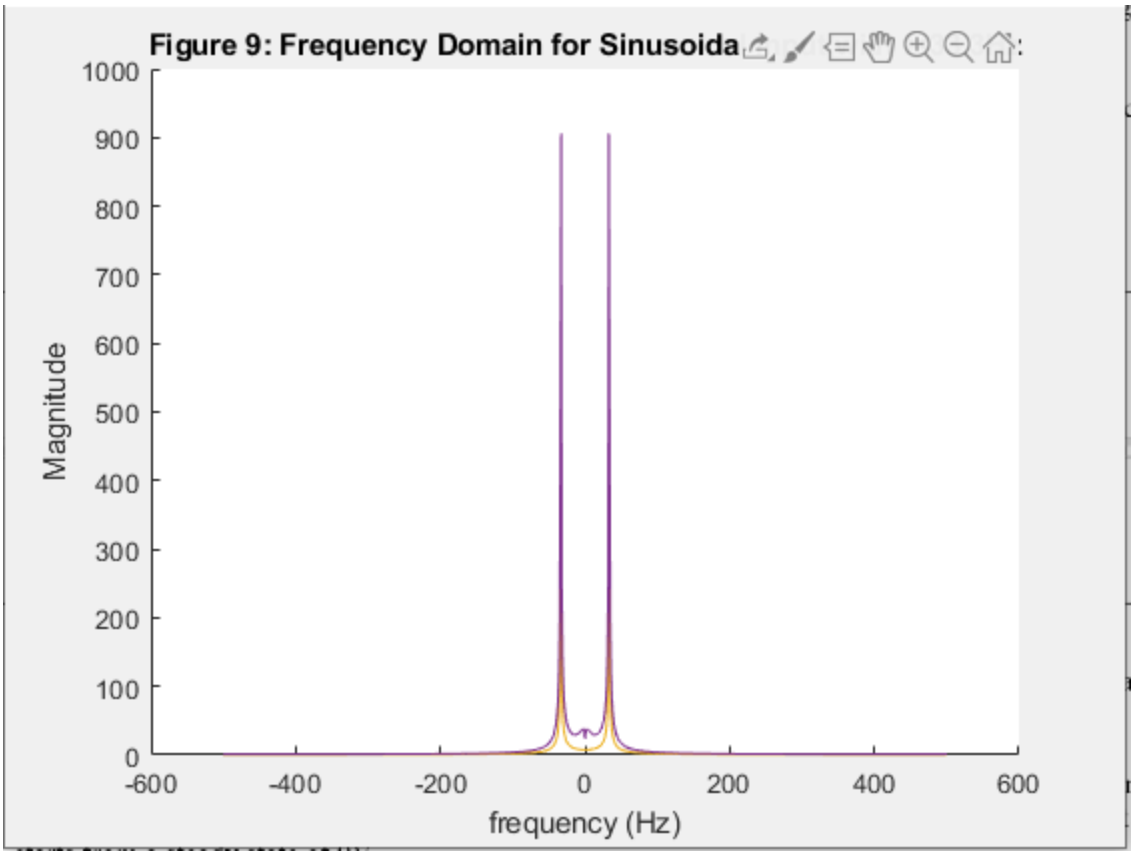


The plot above shows the input and output voltages for a gaussian pulse input. The output voltage is seen to be much higher amplitude than it stabilizes near 0, it follows a gaussian distribution.

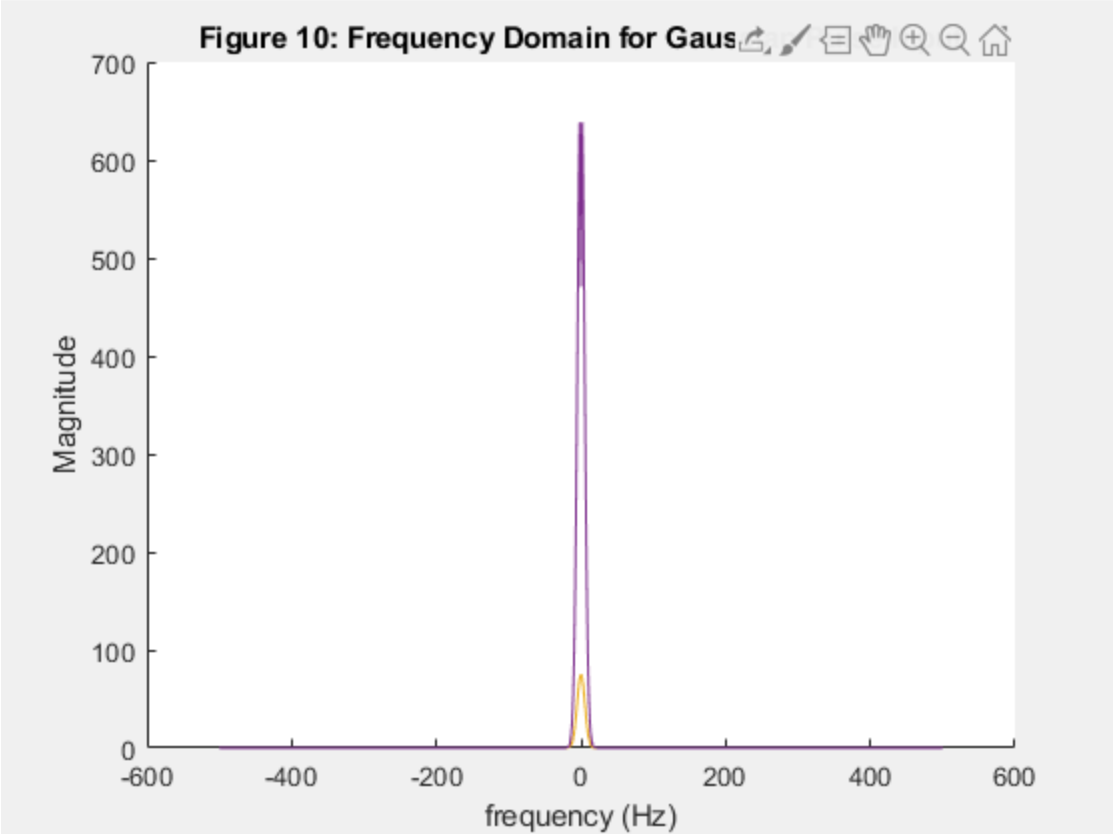
e)



The plot above shows the frequency domain for step unit. The output is seen to similar behave as a Dirac delta which is centered at 0 Hz.



The plot above shows that the input and output to behave like Dirac delta function that are centered at around -30 Hz to 30 Hz.



The above plot shows the frequency domain for the gaussian pulse input, it is seen that both input and output behave like the gaussian distribution. When comparing the two, the output is seen to have much larger magnitude which is what we saw in the gaussian pulse input plot above.

Question 3

The code for this question is shown below:

```
% paramaters
R1 = 1;
R2 = 2;
R3 = 10;
R4 = 0.1;
RO = 1000;
c = 0.25;
L = 0.2;
Cn = 0.00001;

%matrices
Vin=1;
G=zeros(6);
C=zeros(6);
G(1,:)=[1 0 0 0 0 0];
C(1,:)=[0 0 0 0 0 0];
G(2,:)=[(-1/R1) (1/R2+1/R1) 0 0 0 1];
```

```
C(2,:)=[-c +c 0 0 0 0];
G(3,:)=[0 0 1/R3 0 0 -1];
C(3,:)=[0 0 Cn 0 0 0];
G(4,:)=[0 0 -1*100/R3 1 0 0];
C(4,:)=[0 0 0 0 0 0];
G(5,:)=[0 0 0 -1/R4 (1/R4+1/RO) 0];
C(5,:)=[0 0 0 0 0 0];
G(6,:)=[0 -1 1 0 0 0];
C(6,:)=[0 0 0 0 0 L];



C(2,:)=[-0.25 +0.25 0 0 0 0];
timesteps=1000;
Step=1/1000;
fD_Q2=(-(timesteps)/2:(timesteps)/2)*(1/Step/timesteps+1);
simInput=zeros(3,timesteps+1);
for x=1:3
    for i=1:timesteps
        simInput(x,i)=exp(-(i*Step-0.1)^2/(2*0.03^2));
    end
end
simdata=zeros(3,3,timesteps+1);

for x=1:3

    if(x==1)
        cn1=Cn;
    elseif(x==2)
        cn1=Cn*100;
    else
        cn1=Cn*1000;
    end
    C(3,:)=[0 0 cn1 0 0 0];
    oldVoltage=[0;
        0;
        0;
        0;
        0;
        0];
    for i=1:timesteps
        Vin=simInput(x,i);
        F=[Vin;
            0;
            -0.001*randn();
            0;
            0;
            0];
        simdata(x,1,i+1)=i*Step;
        simdata(x,2,i+1)=Vin;
        V=(C/Step+G)\(C*oldVoltage/Step+F);
        simdata(x,3,i+1)=V(5);
        oldVoltage=V;
    end
end
C(3,:)=[0 0 Cn 0 0 0];
```

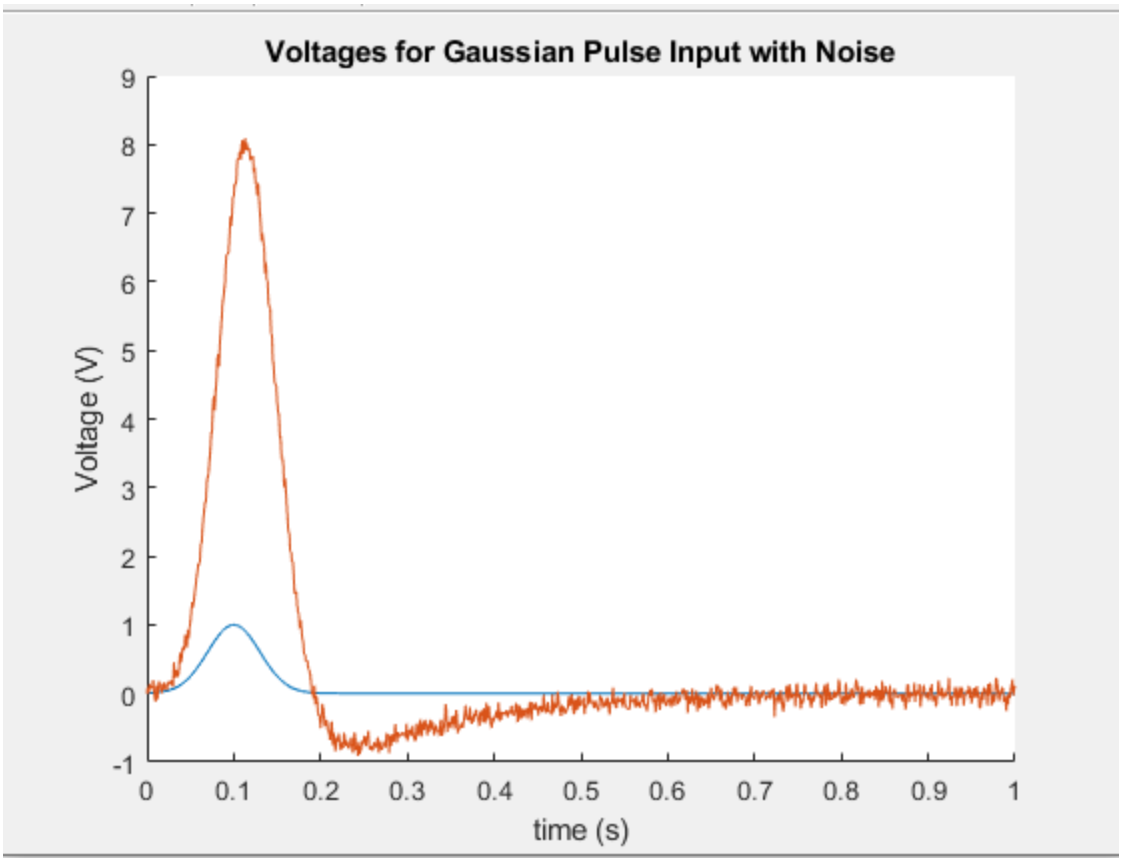
```
simdata_1=zeros(3,timesteps+1);
simdata_1(:,:)=simdata(1,:,:);
figure(1)
hold on;
plot(simdata_1(1,:),simdata_1(2,:));
plot(simdata_1(1,:),simdata_1(3,:));
hold off;
title('Voltages for Gaussian Pulse Input with Noise');
ylabel('Voltage (V)');
xlabel('time (s)');

figure(2)
X=fft(simdata_1(2,:));
Y=fft(simdata_1(3,:));
hold on;
plot(fD_Q2,fftshift(abs(X)));
plot(fD_Q2,fftshift(abs(Y)));
hold off;
title('Frequency Domain for Gaussian Pulse Input with Noise');
ylabel('Magnitude');
xlabel('frequency (Hz)');
simdata_1=zeros(3,timesteps+1);
simdata_1(:,:)=simdata(1,:,:);
figure(3)
subplot(3,1,1)
hold on;
plot(simdata_1(1,:),simdata_1(2,:));
plot(simdata_1(1,:),simdata_1(3,:));
hold off;
title({'Voltages with Noise for Different Cn'});
ylabel('Voltage (V)');
xlabel('time (s)');

simdata_2=zeros(3,timesteps+1);
simdata_2(:,:)=simdata(2,:,:);
subplot(3,1,2)
hold on;
plot(simdata_2(1,:),simdata_2(2,:));
plot(simdata_2(1,:),simdata_2(3,:));
hold off;
title('Cn = 1mF');
ylabel('Voltage (V)');
xlabel('time (s)');

simdata_3=zeros(3,timesteps+1);
simdata_3(:,:)=simdata(3,:,:);
subplot(3,1,3)
hold on;
plot(simdata_3(1,:),simdata_3(2,:));
plot(simdata_3(1,:),simdata_3(3,:));
hold off;
title('Cn=10mF');
ylabel('Voltage (V)');
xlabel('time (s)');
```

```
C(2,:)=[-c +c 0 0 0 0];
timesteps_ALT=10000;
nStep=1/timesteps_ALT;
numT4=timesteps_ALT+1;
fD_Q3=(-(numT4-1)/2:(numT4-1)/2)*(1/nStep/numT4);
datasimulation1=zeros(3,3,timesteps_ALT+1);
oldVoltage=[1; 0; 0; 0; 0; 0];
simInput_ALT=zeros(3,timesteps_ALT+1);
freq=1/0.03;
for x=1:1
    for i=1:timesteps_ALT
        simInput_ALT(x,i)=exp(-(i*nStep-0.1)^2/(2*0.03^2));
    end
end

for x=1:1
    C(3,:)=[0 0 Cn 0 0 0];

    oldVoltage=[0; 0; 0; 0; 0; 0];

    for i=1:timesteps_ALT
        Vin=simInput_ALT(x,i);
        In=0.001*randn();
        F=[Vin; 0; -In; 0; 0; 0];
        datasimulation1(x,1,i+1)=i*nStep;
        datasimulation1(x,2,i+1)=Vin;
        A=C/nStep+G;
        V=(A)\(C*oldVoltage/nStep+F);
        datasimulation1(x,3,i+1)=V(5);
        oldVoltage=V;
    end

end
```

a)

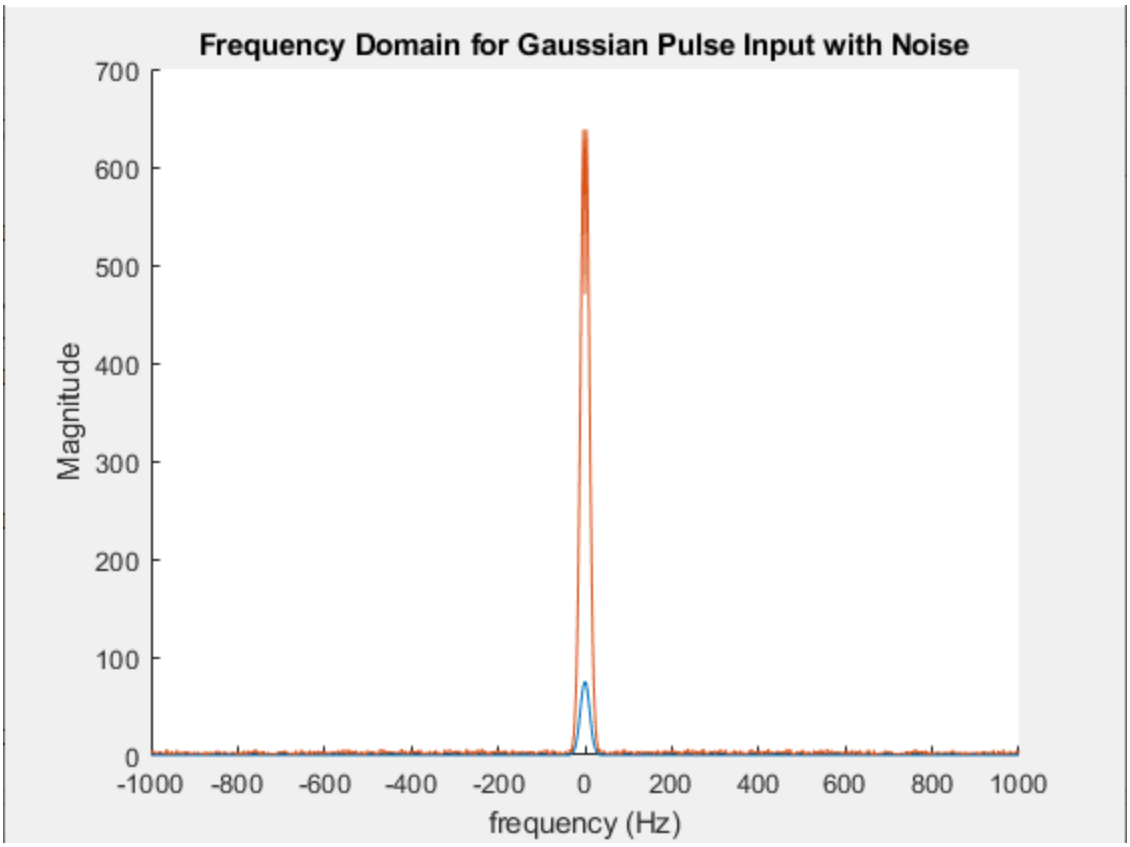
$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -C_1 & C_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_1 \end{matrix}$$

b)

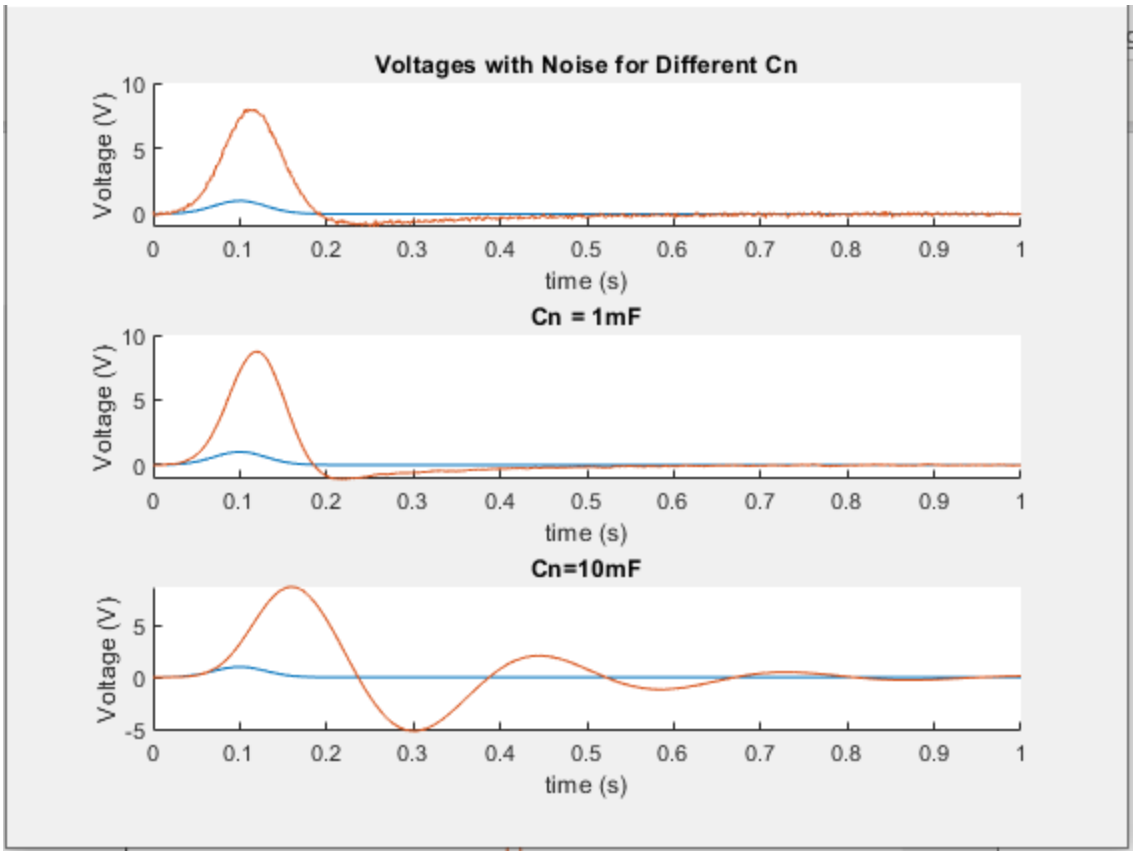


The output in read is seen to have much larger voltage amplitude and it takes longer to stabilize to 0 voltage.

c/d)

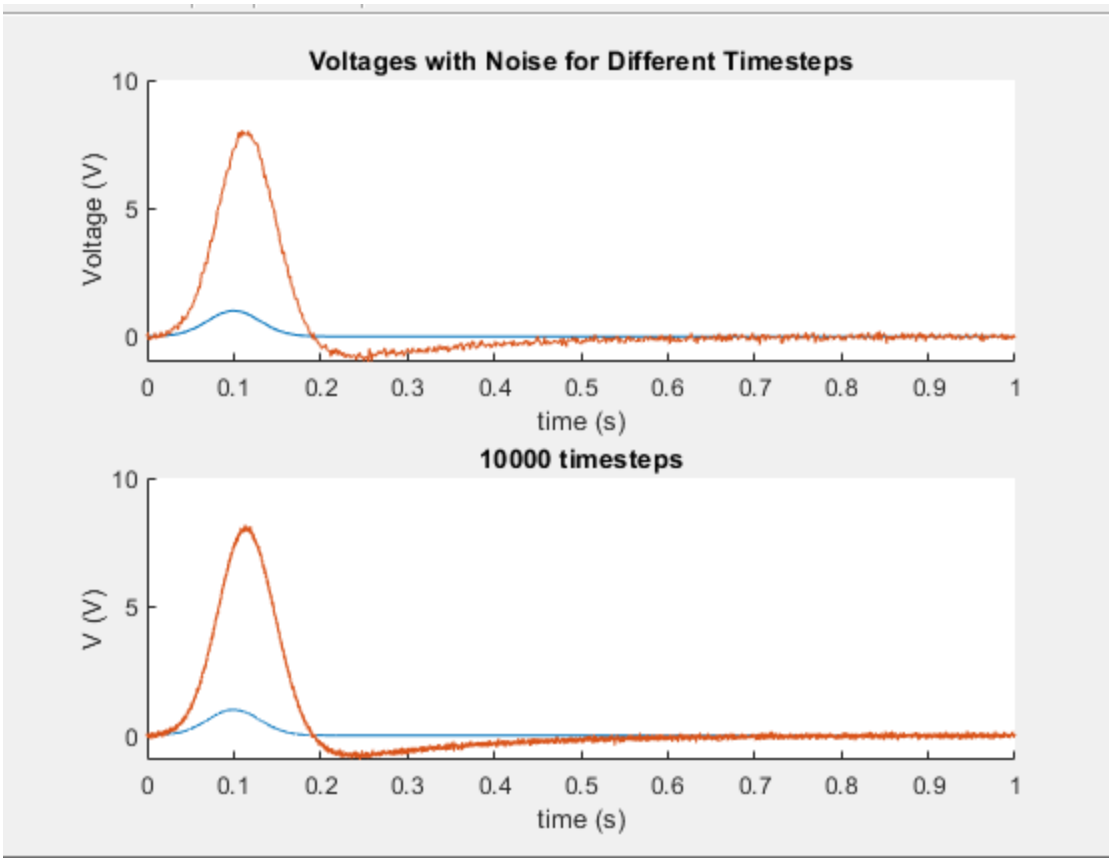

Frequency Domain for Gaussian Pulse Input with Noise
Magnitude
frequency (Hz)
700
600
500
400
300
200
100
0
-1000
-800
-600
-400
-200
0
200
400
600
800
1000

e)



The plots above show different cn values where when increasing the cn value it causes the system to take longer time to stabilize. The noise is canceled when increasing the cn value also.

e)



The plot above shows the voltage with noise for a 10000 timesteps and it is seen that the noise is much less than 1000 timesteps (upper plot).

Question 4

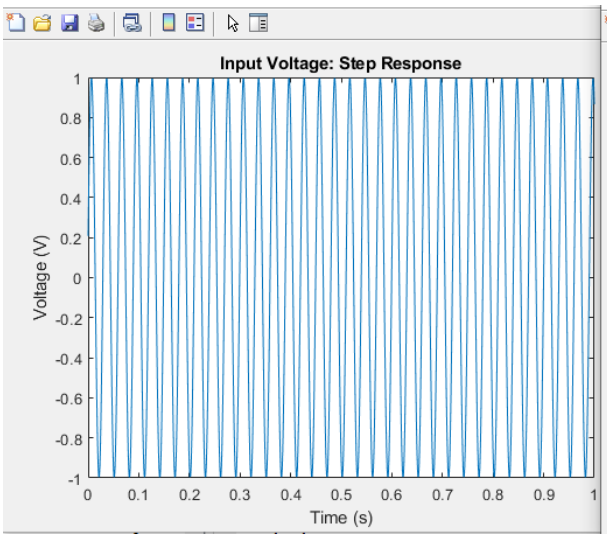
a) In order to implement this a new matrix must be added to the B, so the equation can be rewritten to a form shown below:

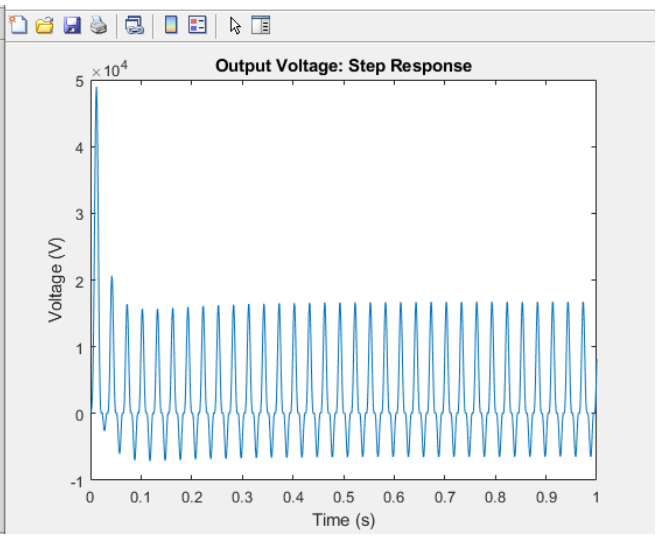
$$C\frac{dV}{dt} + G\hat{V} + \hat{B}(\hat{V}) = \hat{F}(t)$$

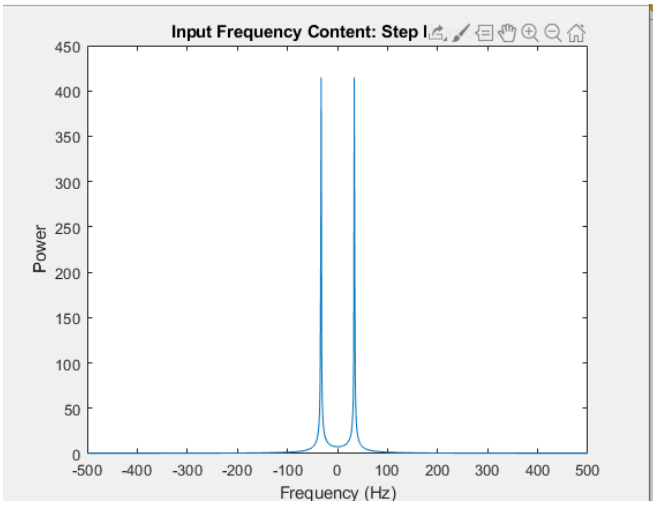
A jacobian matrix must be now implemented which is the partial derivative of the B matrix's contents with respect to the V matrix. The matrix can be rewritten as shown below:

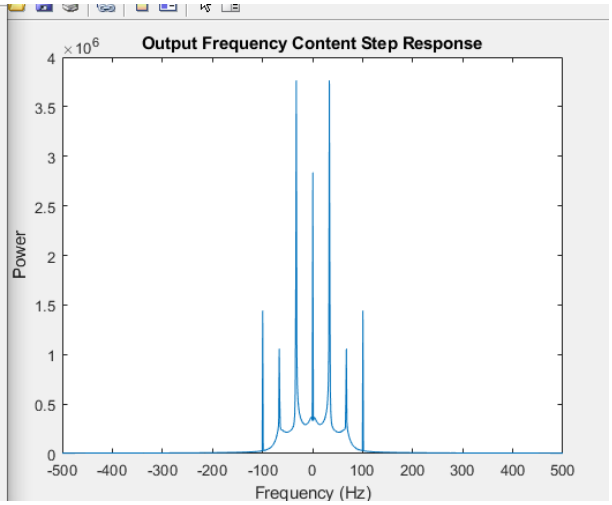
$$\left(\frac{\bar{C}}{\Delta t} + \hat{G}\right)\widehat{V_n} - \frac{C}{\Delta t}\widehat{V_{n-1}} - \hat{B}(\hat{V}) - \hat{F}(t) = 0$$

The above equation can be solved iteratively at each time step.









The code is shown below:

```
R1 = 1;
R2 = 2;
R3 = 10;
R4 = 0.1;
RO = 1000;
C1 = 0.25;
L1 = 0.2;
alpha = 100;
% new values for beta and gamma
beta=10e6;
gamma=1e9;

inputVoltage=1;
G=zeros(6);
C=zeros(6);
G(1,:)=[1 0 0 0 0 0]; % V1
C(1,:)=[0 0 0 0 0 0]; % V1
G(2,:)=[(-1/R1) (1/R2+1/R1) 0 0 0 1];
C(2,:)=[-C1 +C1 0 0 0 0];
```

```
G(3,:)=[0 0 1/R3 0 0 -1];
C(3,:)=[0 0 0 0 0 0];
G(4,:)=[0 0 -1*alpha/R3 1 0 0];
C(4,:)=[0 0 0 0 0 0];
G(5,:)=[0 0 0 -1/R4 (1/R4+1/RO) 0];
C(5,:)=[0 0 0 0 0 0];
G(6,:)=[0 -1 1 0 0 0];
C(6,:)=[0 0 0 0 0 L1];

stepN=1000;
timestep=1/stepN;

A=(C./timestep+G);

f=1/0.03;

V(1:6,1)=[0;0;0;0;0;0];
oldVolt=V;

for step=1:stepN
    t=step*timestep;
    inputVoltage=sin(2*pi*f*t);
    F=[inputVoltage; 0; 0; 0; 0; 0];
    counter=0;
    while(counter<150)
        I3=V(6);
        voltageD=-2*beta*I3-3*gamma*I3.^2;

        J=[0,0,0,0,0,0;
           0,0,0,0,0,0;
           0,0,0,0,0,0;
           0,0,voltageD,0,0,0;
           0,0,0,0,0,0;
           0,0,0,0,0,0];

        B=[0;
            0;
            0;
            beta*I3.^2+gamma*I3.^3;
            0;
            0;];
        voltageFreq=(C./timestep+G)*V-(C./timestep)*oldVolt-F-B;
        H=C./timestep+G-J;
        dV=-inv(H)*voltageFreq;
        V=V+dV;
        if(max(abs(dV))<5e-4)
            break;
        end
        counter=counter+1;
    end
    inputvoltage_1(step)=V(1,1);
    outputVoltage_1(step)=V(5,1);
    oldVolt=V;

end
```