ELEC 4700 Assignment 4

Table of Contents

Question 1

In this question the circuit given was analysed in order to determine the G and C matricies. Once these were determined DC and AC simulations were preformed on the circuit. Using nodal analysis we see that the circuit is described by the following system of differential equations

$$V1 = Vin$$

$$G1(V2 - V1) + C1\frac{d(V2 - V1)}{dt} + Il = 0$$

$$G3V3 - Il = 0$$

$$G3V3 - I3 = 0$$

$$G4(Vo - V4) + GoVo = 0$$

$$V2 - V3 - L\frac{dIl}{dt} = 0$$

$$V4 - aI3 = 0$$

Now these equations must be put in the form CdV/dt + GV = F. V will be a 7 element column vector. Indicies 1 through 5 represent Vi through Vo. Index 6 is Il and index 7 is I3. The F vector has Vin at its first index and 0 for the rest of the Values. The rows of the G and C matrix correspond to the equations in the order that they were presented above. It is easy to see that these matricies are defined as follows

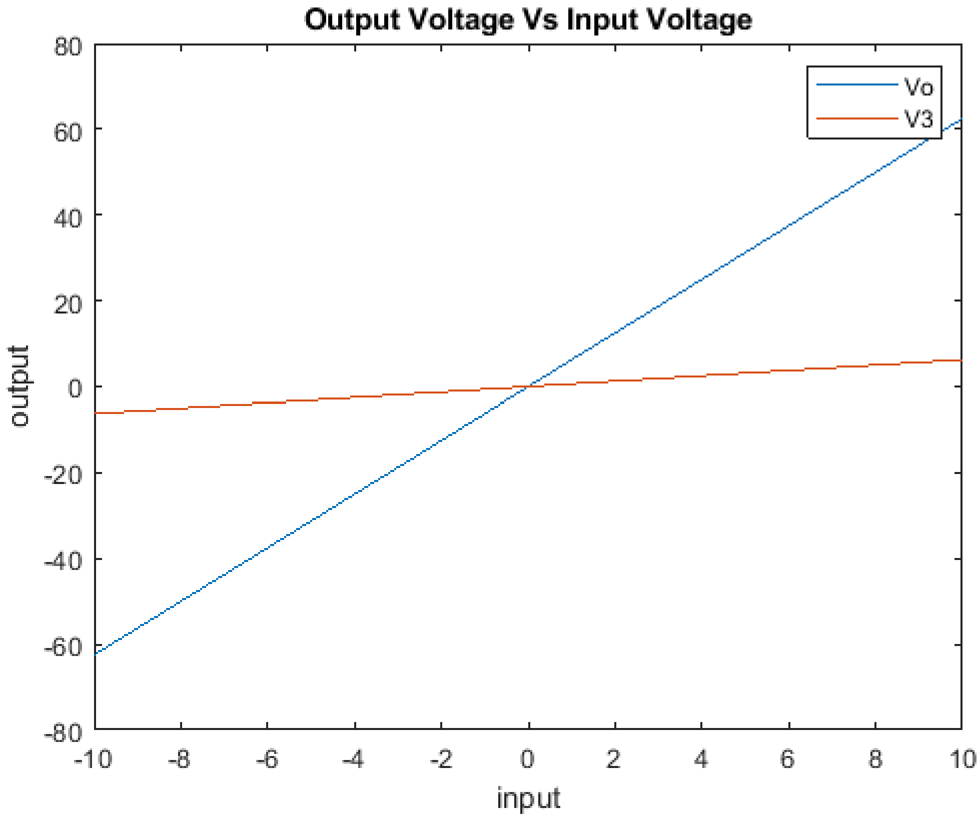
```
R1 = 1;
R2 = 2;
C = 0.25;
L = 0.2;
R3 = 10;
a = 100;
R4 = 0.1;
R0 = 1000;

G = zeros(7,7);
Cm = zeros(7,7);
```

```
G(1,1) = 1;
G(2,1) = -1/R1;
G(2,2) = 1/R1 + 1/R2;
G(2,6) = 1;
G(3,3) = 1/R3;
G(3,6) = -1;
G(4,3) = 1/R3;
G(4,7) = -1;
G(5,4) = -1/R4;
G(5,5) = 1/R4 + 1/R0;
G(6,2) = 1;
G(6,3) = -1;
G(7,4) = 1;
G(7,7) = -a;
Cm(2,1) = -C;
Cm(2,2) = C;
Cm(6,6) = -L;

G
Cm


G =

    1.0000         0         0         0         0         0         0
   -1.0000    1.5000         0         0         0    1.0000         0
         0         0    0.1000         0         0   -1.0000         0
         0         0    0.1000         0         0         0   -1.0000
         0         0         0  -10.0000   10.0010         0         0
         0    1.0000   -1.0000         0         0         0         0
         0         0         0    1.0000         0         0 -100.0000


Cm =

         0         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         0         0         0   -0.2000         0
         0         0         0         0         0         0         0
```

Now we are ready to perform DC simulations. With DC simulations that C matrix is ignore and we simply solve GV = F. Here we swept the input voltage from -10V to 10V.

```
F = zeros(7,1);
V = zeros(7,1);
count = 1;
for i = -10:10
    F(1) = i;
    V = G\F;
    Vodc(count) = V(5);
```

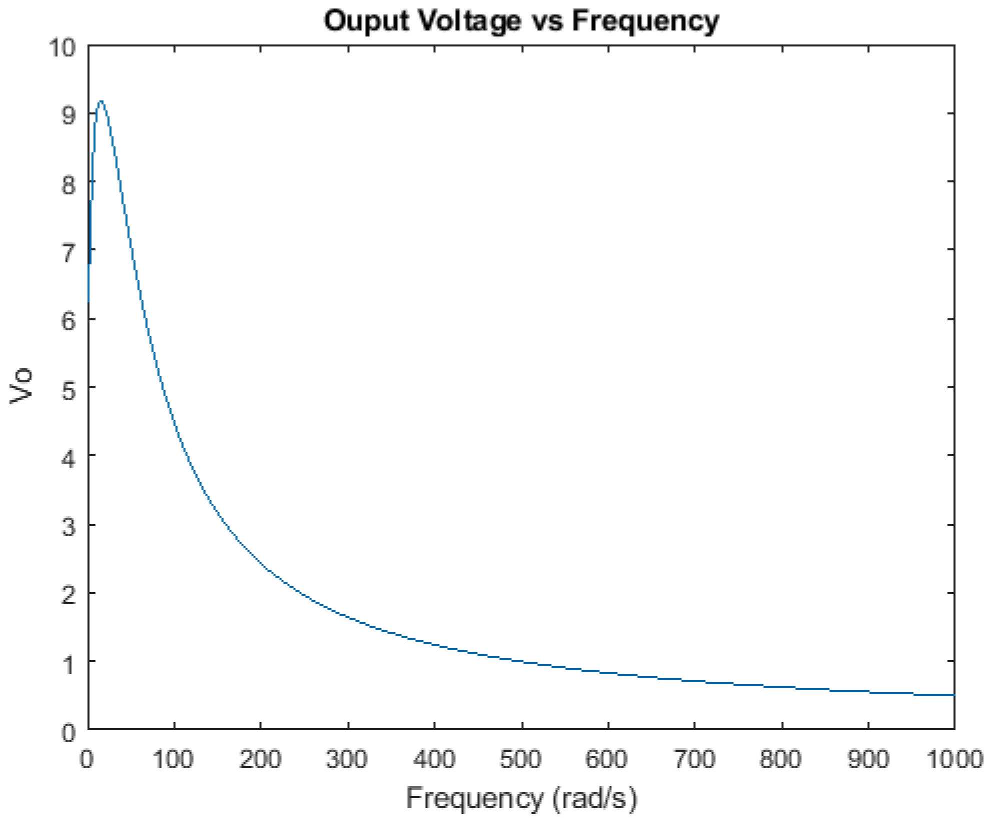
```
    V3(count) = V(3);
    count = count + 1;
end

figure(1)
plot(linspace(-10,10,21),Vodc)
hold on
plot(linspace(-10,10,21),V3)
title('Output Voltage Vs Input Voltage')
xlabel('input')
ylabel('output')
legend('Vo','V3')
```

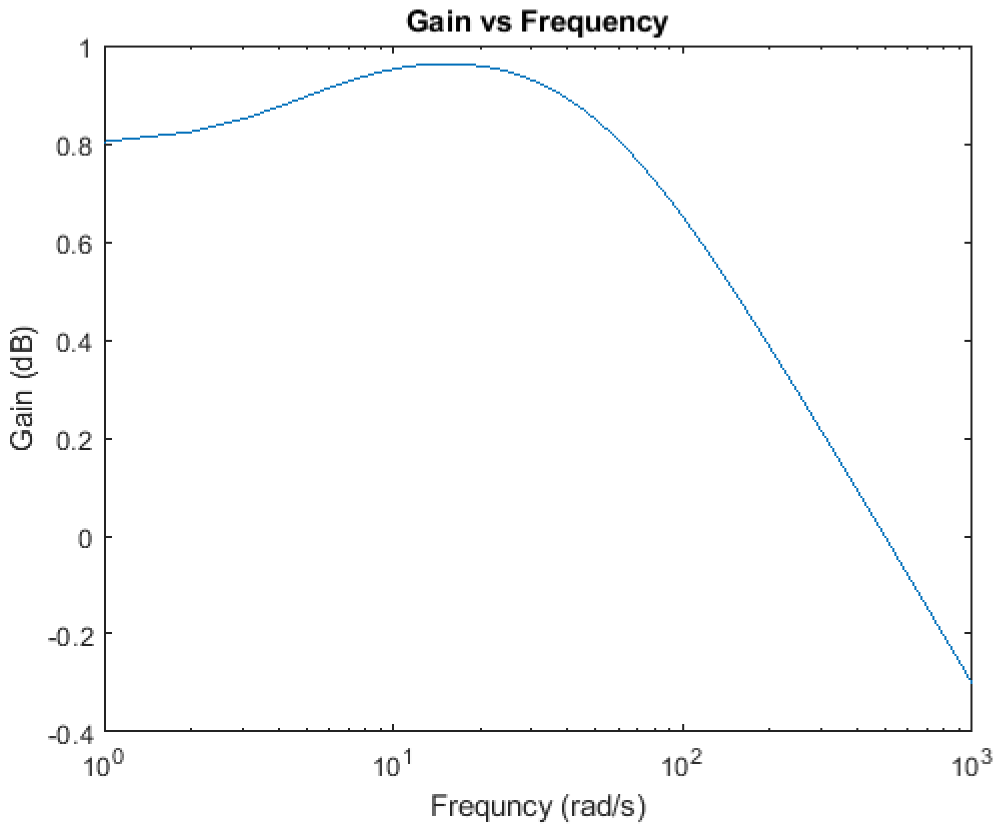


Next is AC simulations. For these we need to solve the following system of of equations (jwC + G)V = F(w). For this simulation I kept the input voltage at a constant 1V regardless of frequency.

```
j = sqrt(-1);

count = 1;
F(1) = 1;
for w = 0:1000
    Gac = G + j*w*Cm;
    V = Gac\F;
    Voac(count) =  V(5);
    count = count+1;
end
figure(2)
```

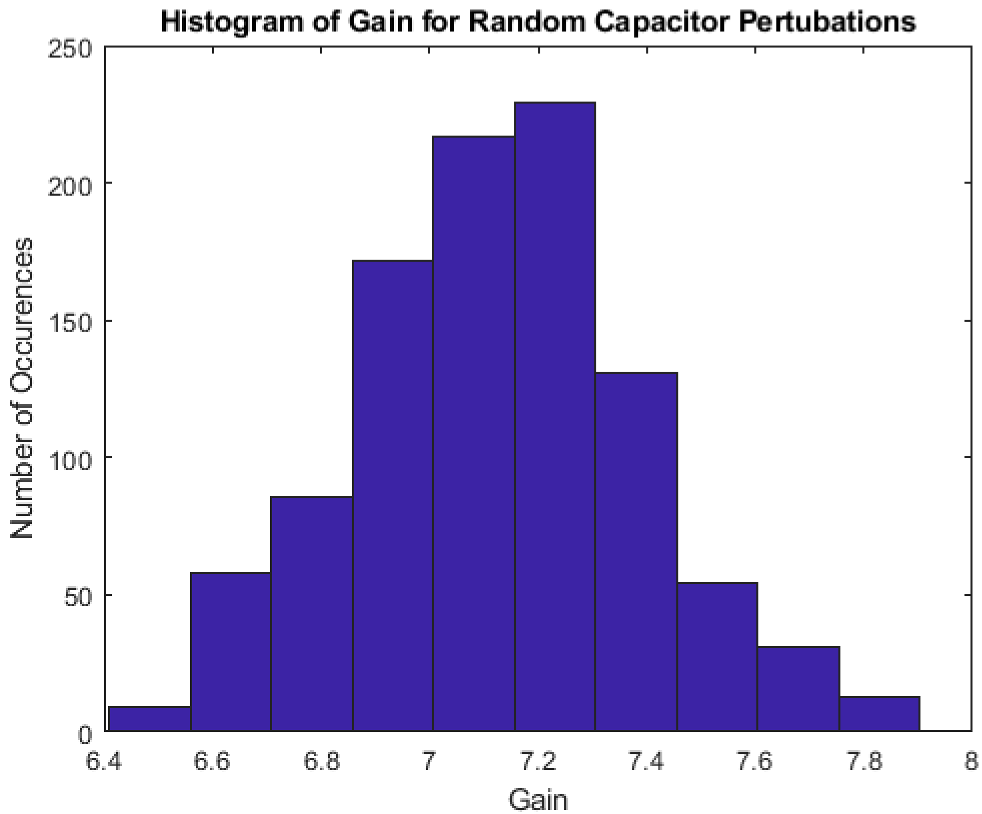
```
plot(0:1000,abs(Voac))
title('Ouput Voltage vs Frequency')
xlabel('Frequency (rad/s)')
ylabel('Vo')

figure(3)
semilogx(0:1000, log10(abs(Voac)))
title('Gain vs Frequency')
xlabel('Frequncy (rad/s)')
ylabel('Gain (dB)')
```





Now we will look at the gain as a function of random pertubations on C. The angular frequency will be fixed at w = pi.

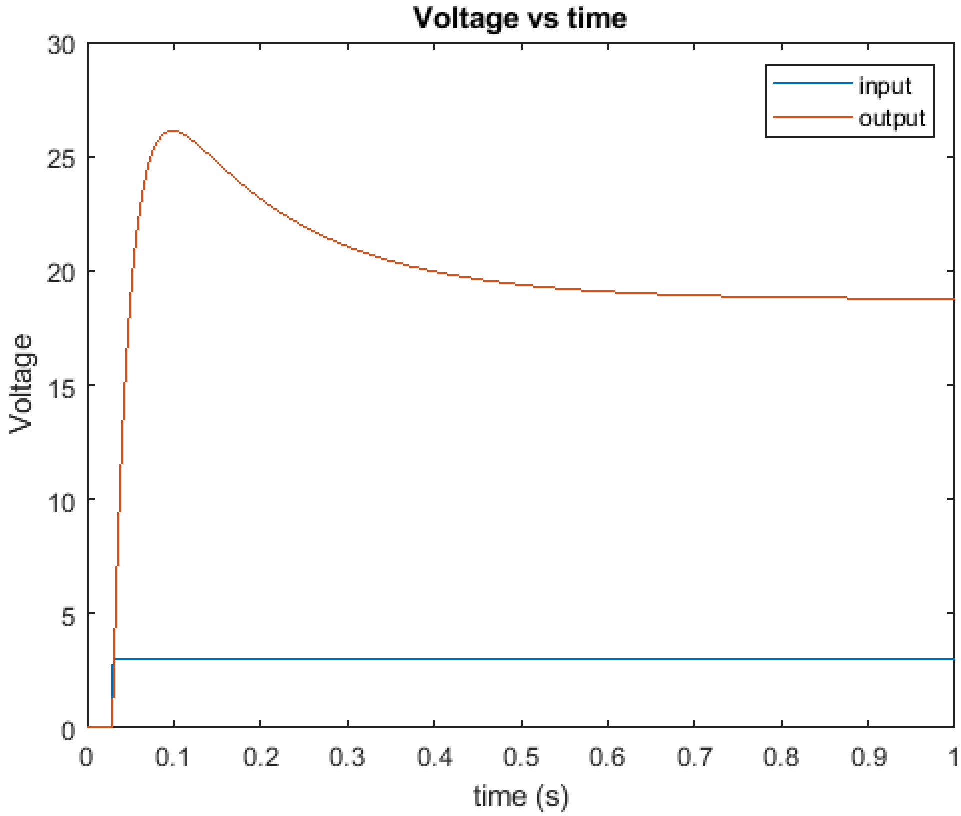
```
Crand = Cm;
for i = 1:1000
    Cr = normrnd(C,0.05);
    Crand(2,1) = -Cr;
    Crand(2,2) = Cr;
    V = (j*pi*Crand + G)\F;
    Vorand(i) = V(5);
end
hist(abs(Vorand));
title('Histogram of Gain for Random Capacitor Pertubations')
xlabel('Gain')
ylabel('Number of Occurences')
```

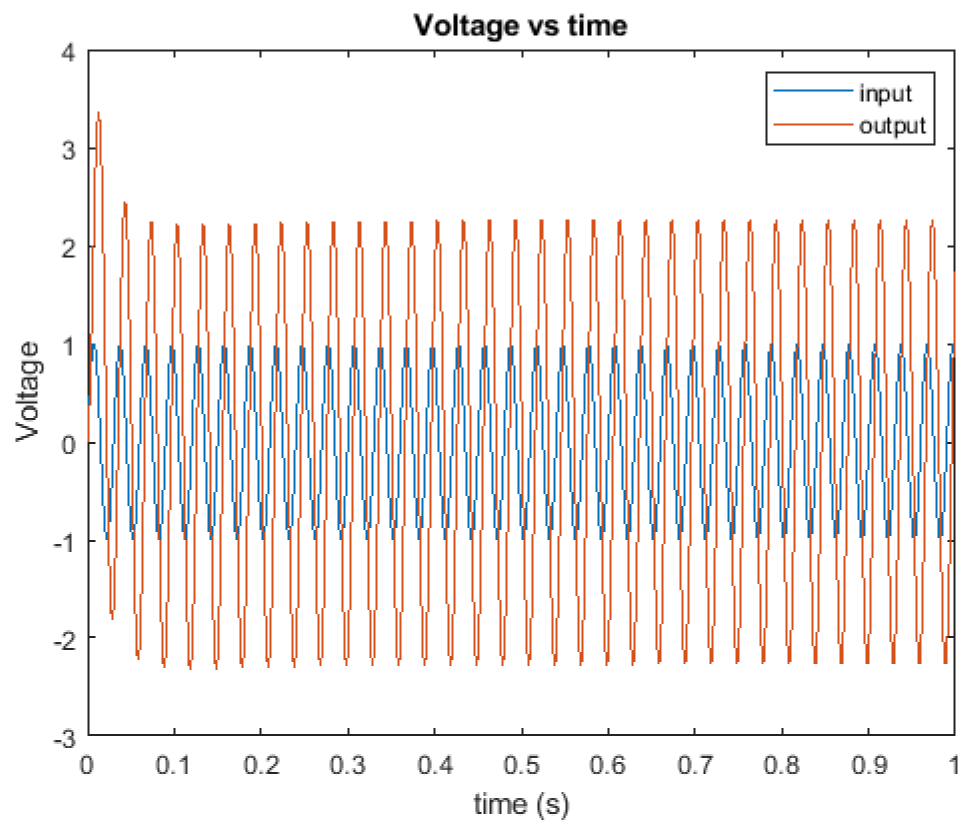


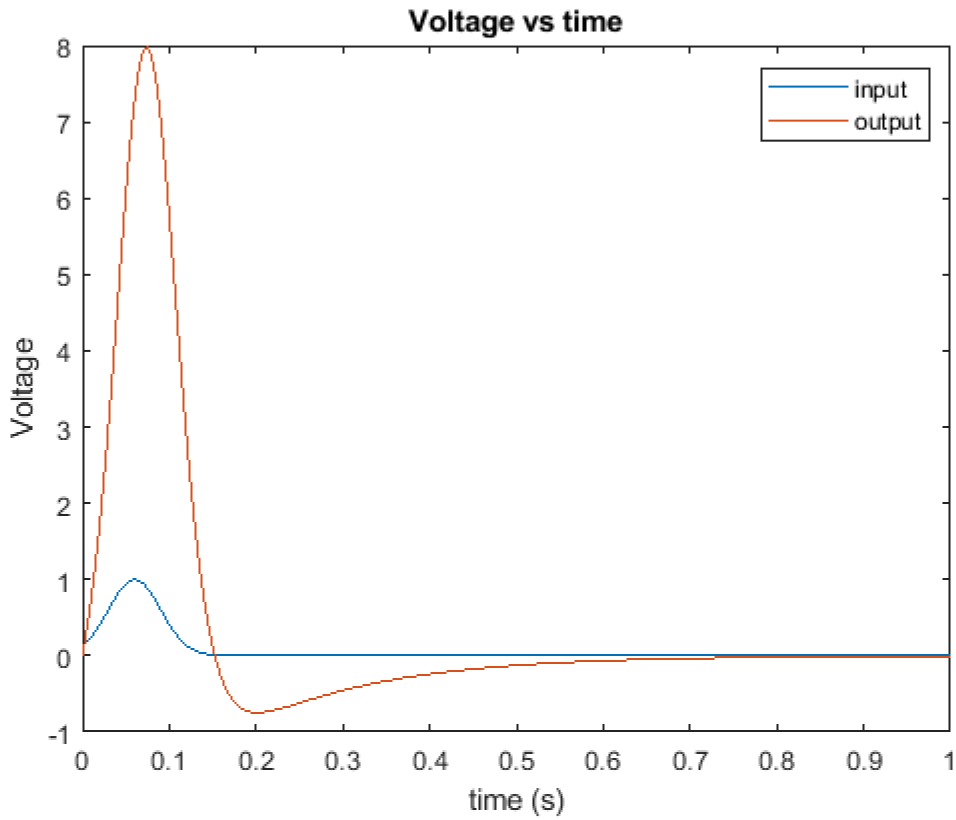
The next step is to perform a transient simulation of this circuit. In order to do this we must solve VdV/dt + GV = F. This can be done by approximating the derivative using the finite difference method. Doing this and rearanging we get (C/dt + G)V(j) = CV(j-1)/dt + F(t(j)) Where j deotes the time index. This circuit was simulated for a step input, a sinusoidal input and a gaussian pulse. The code for this simulation is shown below. It is worth noting that all initial Voltages and current are assumed to be 0.

```
dt = 0.001;

Atrans = Cm/dt + G;

V1 = zeros(7,1);
V2 = zeros(7,1);
V3 = zeros(7,1);
Vo1(1) = 0;
Vo2(1) = 0;
Vo3(1) = 0;
Vi1(1) = 0;
Vi2(1) = 0;
Vi3(1) = 0;
F1 = zeros(7,1);
F2 = zeros(7,1);
F3 = zeros(7,1);
count = 1;
for t = dt:dt:1
    if t >= 0.03
        F1(1) = 3;
```

```
    end
    F2(1) = sin(2*pi*t/0.03);
    F3(1) = exp(-0.5*((t - 0.06)/0.03)^2);
    V1 = Atrans\(Cm*V1/dt + F1);
    V2 = Atrans\(Cm*V2/dt + F2);
    V3 = Atrans\(Cm*V3/dt + F3);
    Vi1(count +1) = V1(1);
    Vi2(count +1) = V2(1);
    Vi3(count +1) = V3(1);
    Vo1(count +1) = V1(5);
    Vo2(count +1) = V2(5);
    Vo3(count +1) = V3(5);
    count = count+1;
end

figure(4)
plot(0:dt:1,Vi1)
hold on
plot(0:dt:1,Vo1)
title('Voltage vs time')
xlabel('time (s)')
ylabel('Voltage')
legend('input','output')

figure(5)
plot(0:dt:1,Vi2)
hold on
plot(0:dt:1,Vo2)
title('Voltage vs time')
xlabel('time (s)')
ylabel('Voltage')
legend('input','output')

figure(6)
plot(0:dt:1,Vi3)
hold on
plot(0:dt:1,Vo3)
title('Voltage vs time')
xlabel('time (s)')
ylabel('Voltage')
legend('input','output')
```

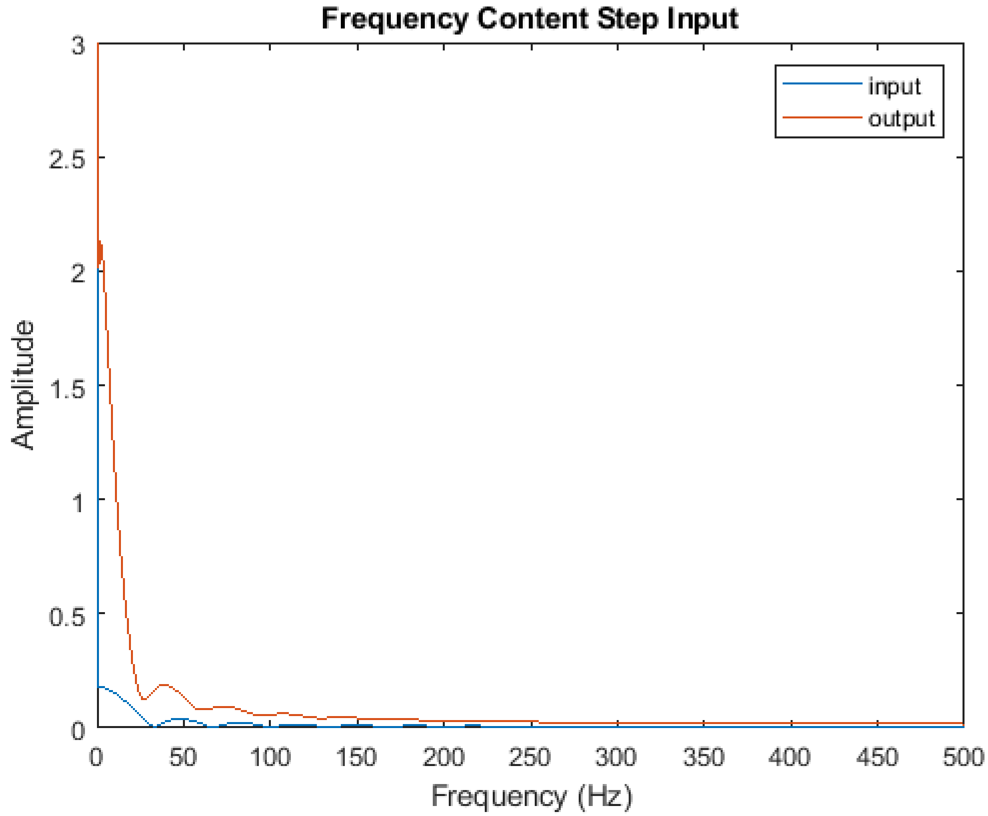

Voltage vs time
input
output
Voltage
time (s)
30
25
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Voltage vs time
input
output
Voltage
time (s)
4
3
2
1
0
-1
-2
-3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



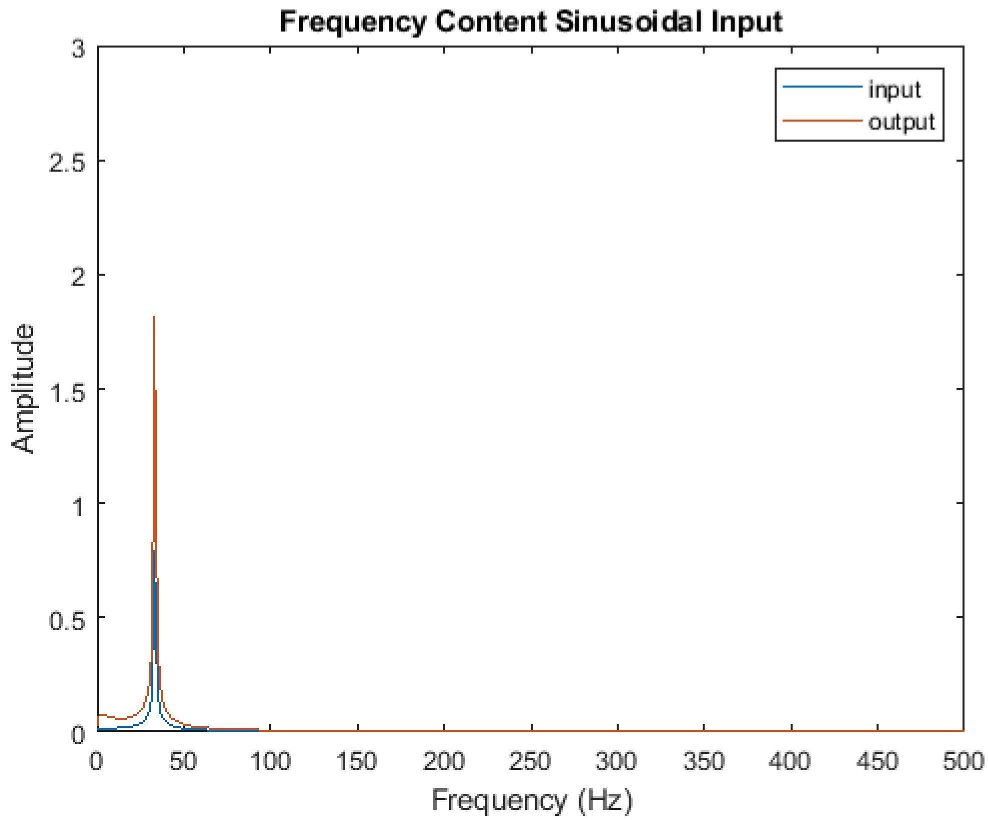
We can also take the fourier transform of the input and output signals to see what is happening in the frequency domain

```
Xin = fft(Vi1); %Take fourier transform
P2in = abs(Xin/1000);
P1in = P2in(1:1000/2+1);
P1in(2:end-1) = 2*P1in(2:end-1); % Calculate singel ended spectrum
f = (1/dt)*(0:(1000/2))/1000; % Sampling frequency
figure(7)
plot(f,P1in)

Xo = fft(Vo1);
P2o = abs(Xo/1000);
P1o = P2o(1:1000/2+1);
P1o(2:end-1) = 2*P1o(2:end-1);
f = (1/dt)*(0:(1000/2))/1000;
hold on
plot(f,P1o)
title('Frequency Content Step Input')
xlabel('Frequency (Hz)')
ylabel('Amplitude')
ylim([0 3])
legend('input','output')
```



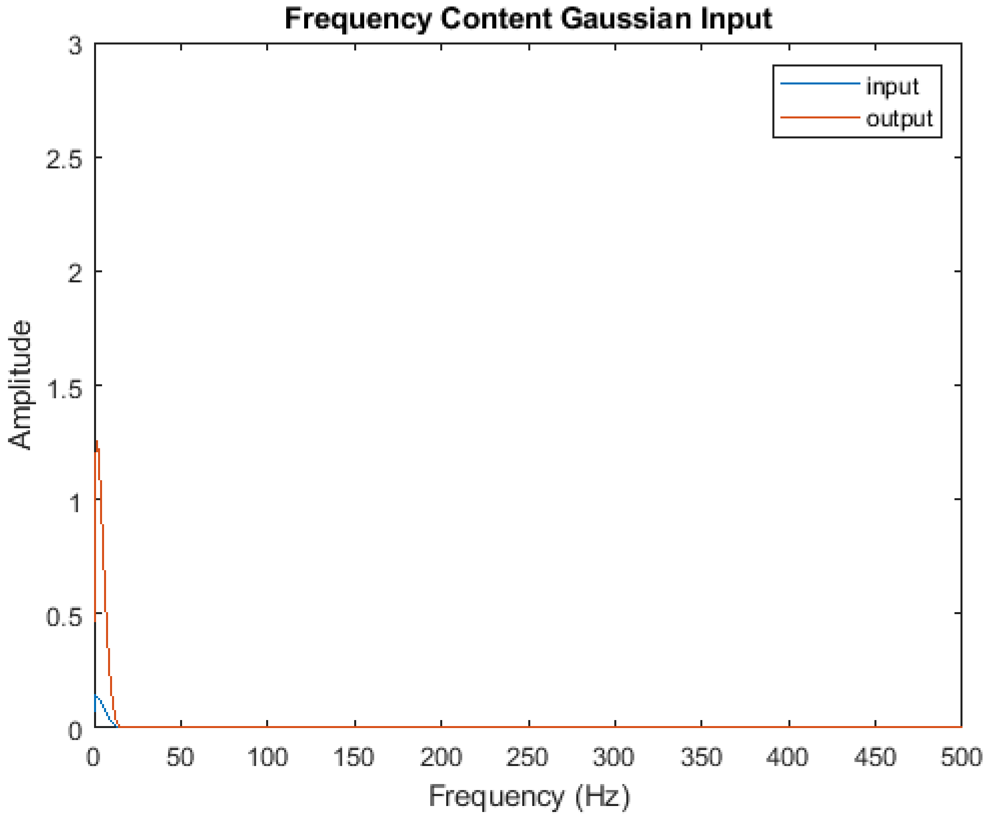
For the step input we see that the fourier transform of the step is the sinc function. We see at the output we have a slightly distorted sinc function. This distorion comes from the fact that the gain in the pass band is not constant. Furthermore we see that the higher frequency components are attenuated.

```
Xin = fft(Vi2); %Take fourier transform
P2in = abs(Xin/1000);
P1in = P2in(1:1000/2+1);
P1in(2:end-1) = 2*P1in(2:end-1); % Calculate singel ended spectrum
figure(8)
plot(f,P1in)

Xo = fft(Vo2);
P2o = abs(Xo/1000);
P1o = P2o(1:1000/2+1);
P1o(2:end-1) = 2*P1o(2:end-1);
hold on
plot(f,P1o)
title('Frequency Content Sinusoidal Input')
xlabel('Frequency (Hz)')
ylabel('Amplitude')
ylim([0 3])
legend('input','output')
```



For the sinusiodal input we see that the frequency response has two peaks at about 33Hz witch is the frequency of the input and output signal.

```
Xin = fft(Vi3); %Take fourier transform
P2in = abs(Xin/1000);
P1in = P2in(1:1000/2+1);
P1in(2:end-1) = 2*P1in(2:end-1); % Calculate singel ended spectrum
figure(9)
plot(f,P1in)

Xo = fft(Vo3);
P2o = abs(Xo/1000);
P1o = P2o(1:1000/2+1);
P1o(2:end-1) = 2*P1o(2:end-1);
hold on
plot(f,P1o)
title('Frequency Content Gaussian Input')
xlabel('Frequency (Hz)')
ylabel('Amplitude')
ylim([0 3])
legend('input','output')
```



For the gaussian input we see that the fourier transform of a gaussian signal is also a gaussian signal. Since the majority of the frequency response of the gaussian input falls within the pass band of this circuit, we see a gaussian frequency response at the output as well.

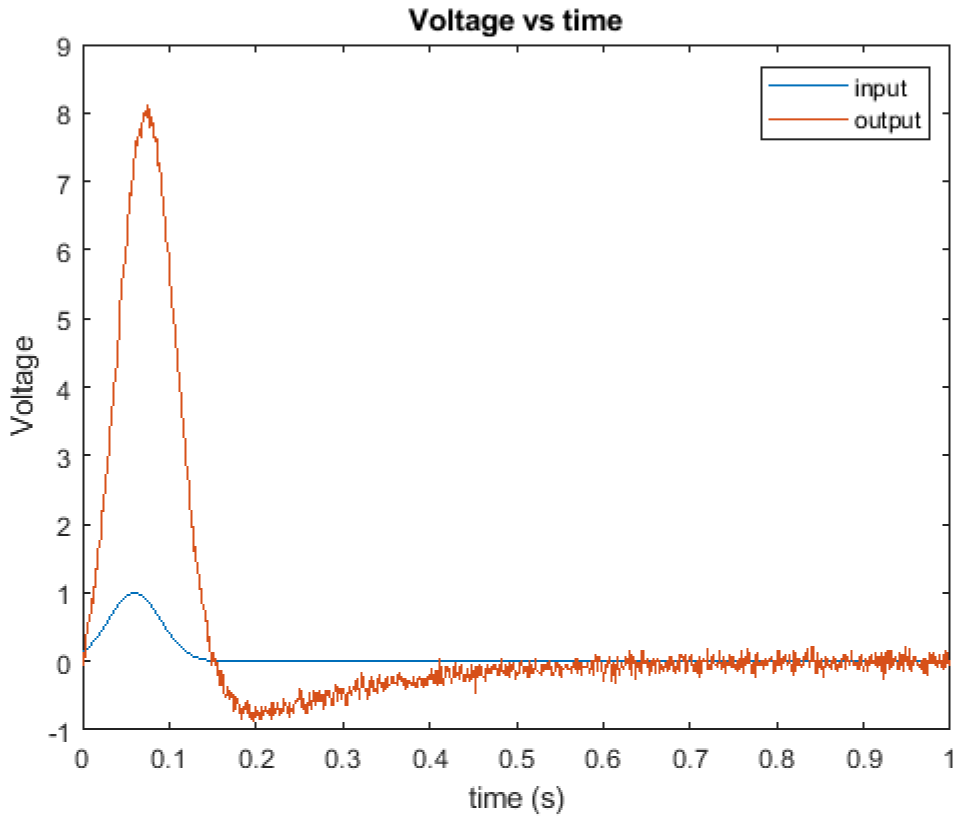
Part 3 Circuit With Noise

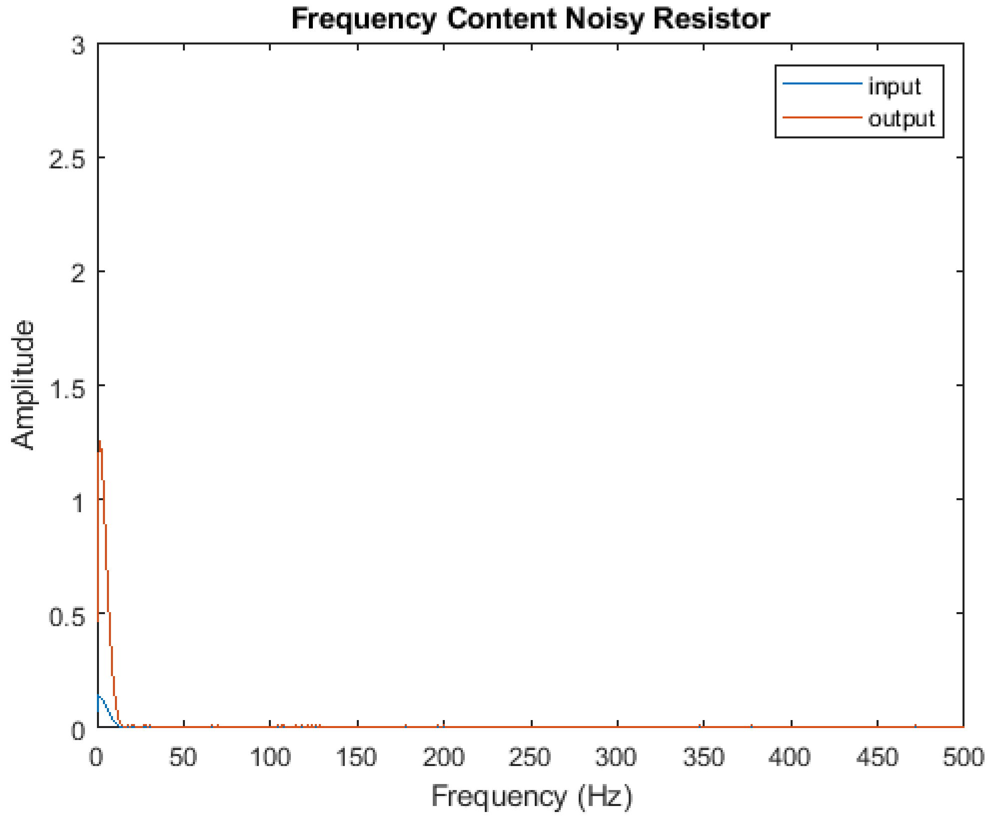
In this part, the model is imporved by adding a noise current source and a cpcacitor, to bandlimit the noise, in parallel with R3. This way R3 will behave more like a real resistor. This addition will alter the C matrix and the F input vector. To do this recall the third equation in part 1. In order to add the current source and capacitor the equation is changed as follows. $G3V3Cn\frac{dV3}{dt} - Il = In$ Where In is the noise current and Cn is the capacitor. From this we see that F(3) will need to be set to the noise current, In. And the C matrix needs to modified as folows.

```
In = 0.001;
Cn = 0.00001;
Cm(3,3) = Cn;
G
Cm

dt = 0.001;
Atrans = Cm/dt + G;

F = zeros(7,1);
V = zeros(7,1);
Vo(1) = 0;
```

```
Vi(1) = 0;

count = 1;
for t = dt:dt:1
    F(1) = exp(-0.5*((t - 0.06)/0.03)^2);
    F(3) = In*normrnd(0,1);
     V = Atrans\(Cm*V/dt + F);
     Vi(count + 1) = F(1);
     Vo(count + 1) = V(5);
     count = count + 1;
end

figure(10)
plot(0:dt:1,Vi)
hold on
plot(0:dt:1,Vo)
title('Voltage vs time')
xlabel('time (s)')
ylabel('Voltage')
legend('input','output')

Xin = fft(Vi);
P2in = abs(Xin/1000);
P1in = P2in(1:1000/2+1);
P1in(2:end-1) = 2*P1in(2:end-1);
f = (1/dt)*(0:(1000/2))/1000;
figure(11)
plot(f,P1in)

Xo = fft(Vo);
P2o = abs(Xo/1000);
P1o = P2o(1:1000/2+1);
P1o(2:end-1) = 2*P1o(2:end-1);
f = (1/dt)*(0:(1000/2))/1000;
hold on
plot(f,P1o)
title('Frequency Content Noisy Resistor')
xlabel('Frequency (Hz)')
ylabel('Amplitude')
ylim([0 3])
legend('input','output')


G =

    1.0000         0         0         0         0         0         0
   -1.0000    1.5000         0         0         0    1.0000         0
         0         0    0.1000         0         0   -1.0000         0
         0         0    0.1000         0         0         0   -1.0000
         0         0         0  -10.0000   10.0010         0         0
         0    1.0000   -1.0000         0         0         0         0
         0         0         0    1.0000         0         0 -100.0000
```

```
Cm =

         0         0         0         0         0         0         0
   -0.2500    0.2500         0         0         0         0         0
         0         0    0.0000         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         0
         0         0         0         0         0         0         0
```

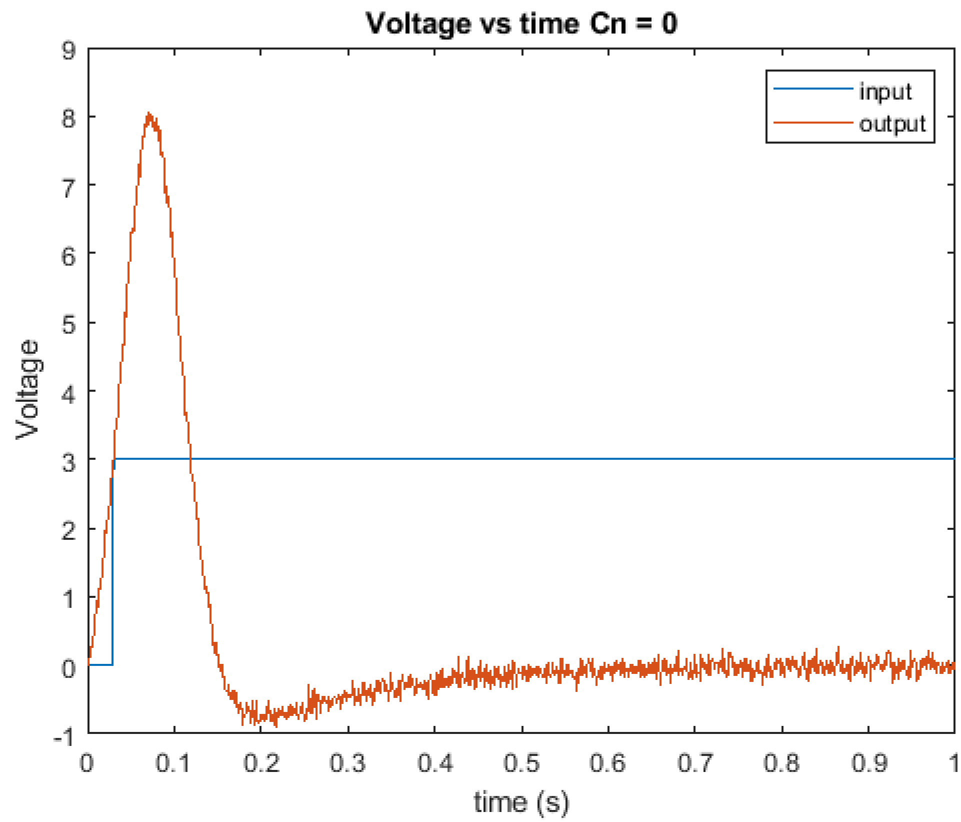


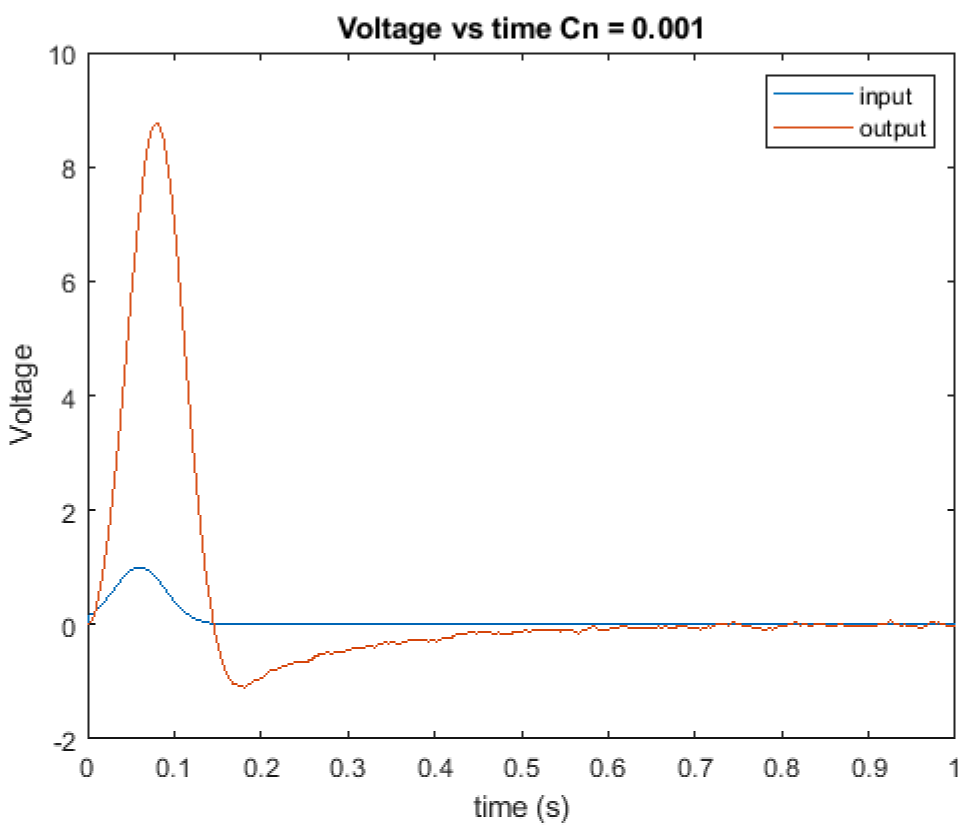


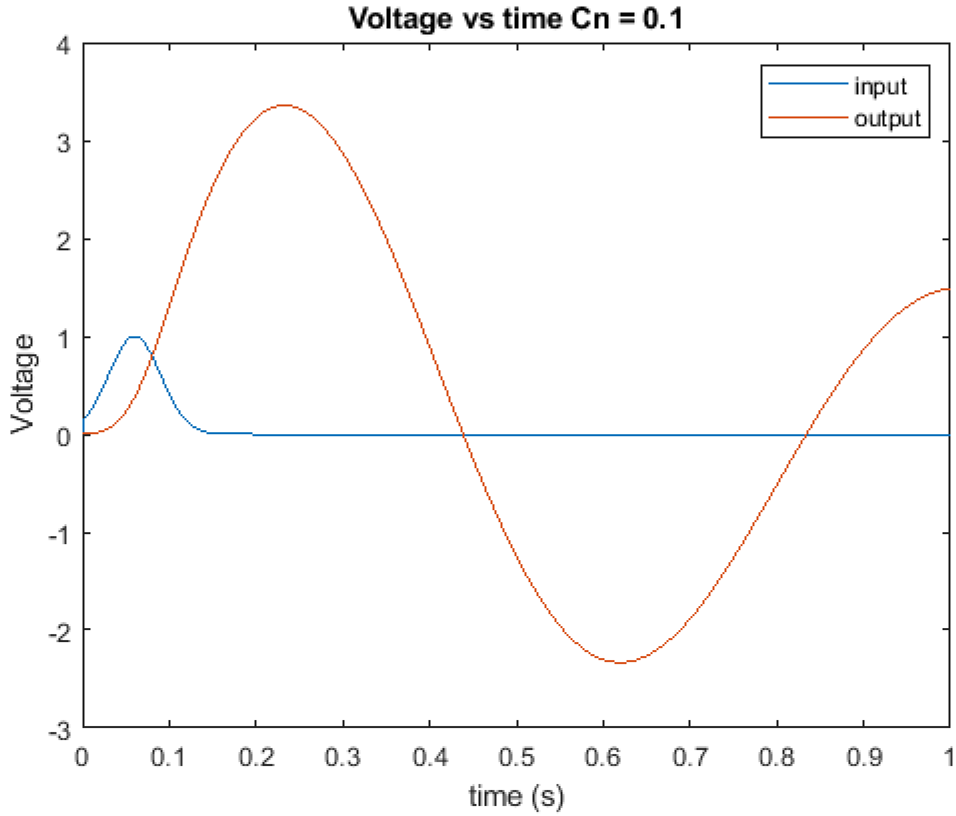
Now the effects of varying Cn will be explored. We are going to simulate this circuit for Cn = 0, Cn = 0.001 and Cn = 0.1

```
Csmall = Cm;
Cmed = Cm;
Clarge = Cm;
Csmall(3,3) = 0;
Cmed(3,3) = 0.001;
Clarge(3,3) = 0.1;


Vsmall = zeros(7,1);
Vmed = zeros(7,1);
Vlarge = zeros(7,1);
Vosmall(1) = 0;
Vomed(1) = 0;
Volarge(1) = 0;
Vi(1) = 0;
count = 1;
for t = dt:dt:1
     F(1) = exp(-0.5*((t - 0.06)/0.03)^2);
     F(3) = In*normrnd(0,1);
     Vsmall = (Csmall/dt + G)\(Csmall*Vsmall/dt + F);
     Vmed = (Cmed/dt + G)\(Cmed*Vmed/dt + F);
     Vlarge = (Clarge/dt + G)\(Clarge*Vlarge/dt + F);
     Vosmall(count + 1) = Vsmall(5);
```

```
        Vomed(count + 1) = Vmed(5);
        Volarge(count + 1) = Vlarge(5);
        Vi(count + 1) = F(1);
        count = count + 1;
    end

    figure(12)
    plot(0:dt:1,Vi1)
    hold on
    plot(0:dt:1,Vosmall)
    title('Voltage vs time Cn = 0')
    xlabel('time (s)')
    ylabel('Voltage')
    legend('input','output')

    figure(13)
    plot(0:dt:1,Vi)
    hold on
    plot(0:dt:1,Vomed)
    title('Voltage vs time Cn = 0.001')
    xlabel('time (s)')
    ylabel('Voltage')
    legend('input','output')

    figure(14)
    plot(0:dt:1,Vi)
    hold on
    plot(0:dt:1,Volarge)
    title('Voltage vs time Cn = 0.1')
    xlabel('time (s)')
    ylabel('Voltage')
    legend('input','output')
```


Voltage vs time Cn = 0
input
output
Voltage
time (s)
9
8
7
6
5
4
3
2
1
0
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Voltage vs time Cn = 0.001
input
output
Voltage
time (s)
10
8
6
4
2
0
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

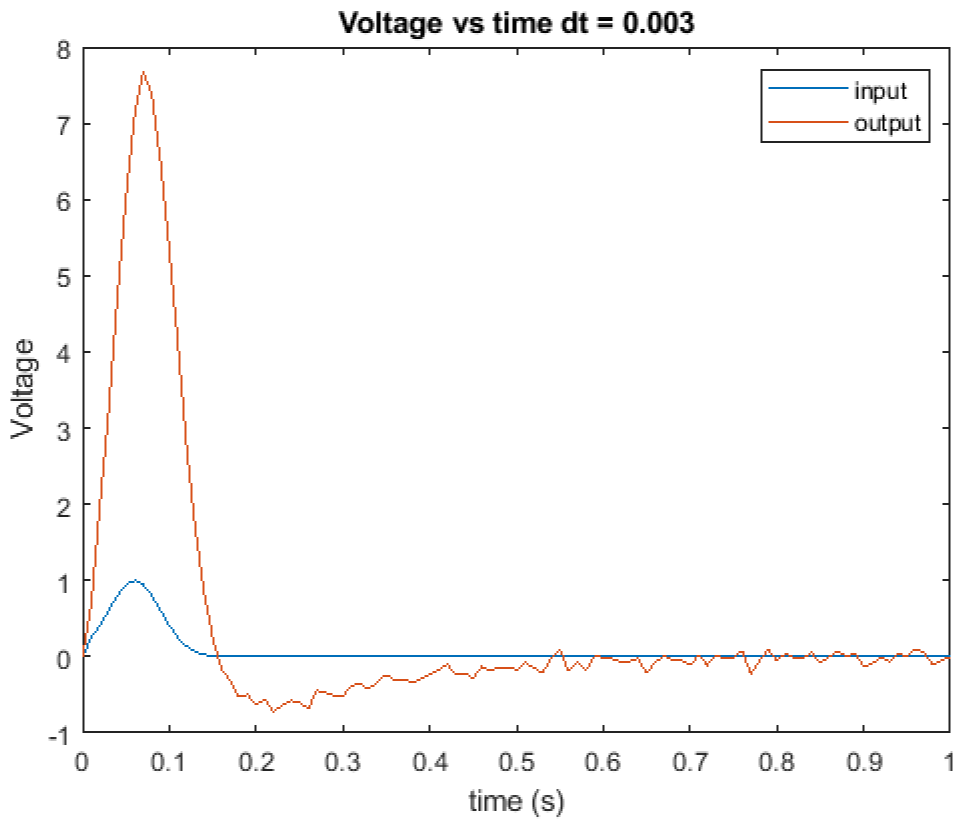


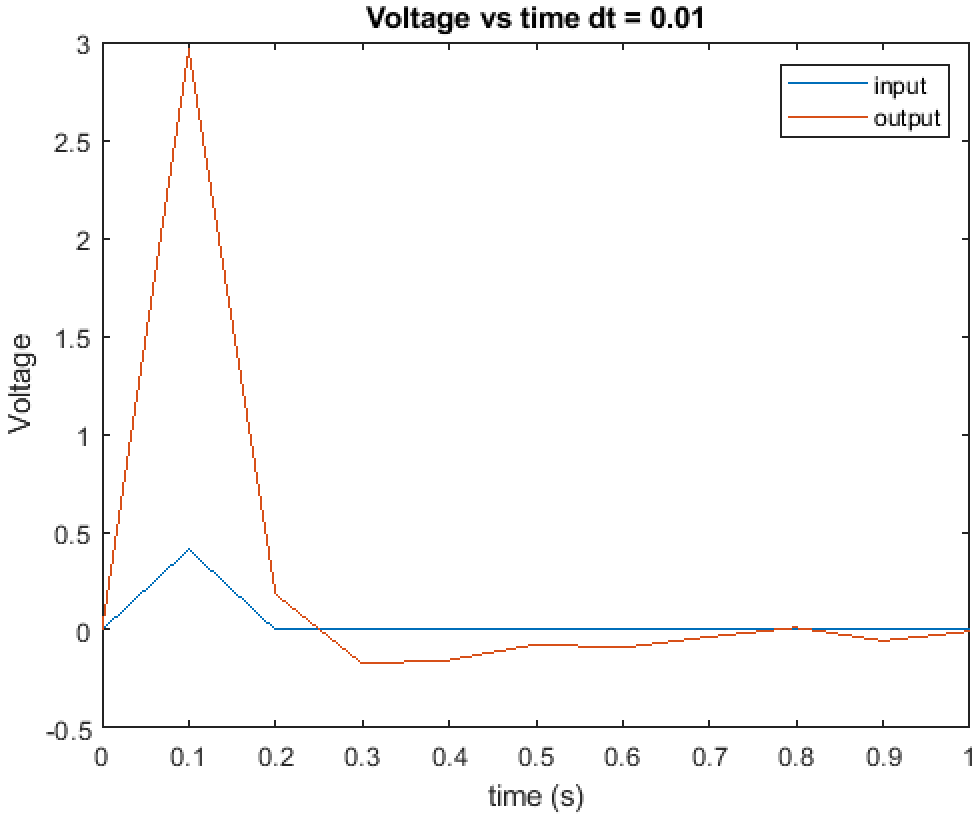
These plots show that initially, the noise is reduced as the capacitor is increased. However, as it is increased further the output signal becomes distorted as higher frequency harmonics become amplified.

Now the effect of varying the time step will be explored. The circuit will be simulated for dt = 0.003 and dt = 0.1

```
dt1 = 0.01;
ViSmallStep(1) = 0;
VoSmallStep(1) = 0;
V = zeros(7,1);
count = 1;
for t = dt1:dt1:1
     F(1) = exp(-0.5*((t - 0.06)/0.03)^2);
     F(3) = In*normrnd(0,1);
     V = (Cm/dt1 + G)\(Cm*V/dt1 + F);
     VoSmallStep(count + 1) = V(5);
     ViSmallStep(count + 1) = F(1);
     count = count + 1;
end

dt2 = 0.1;
ViLargeStep(1) = 0;
VoLargeStep(1) = 0;
V = zeros(7,1);
count = 1;
for t = dt2:dt2:1
     F(1) = exp(-0.5*((t - 0.06)/0.03)^2);
```

```
        F(3) = In*normrnd(0,1);
        V = (Cm/dt2 + G)\(Cm*V/dt2 + F);
        VoLargeStep(count + 1) = V(5);
        ViLargeStep(count + 1) = F(1);
        count = count + 1;
end

figure(15)
plot(0:dt1:1,ViSmallStep)
hold on
plot(0:dt1:1,VoSmallStep)
title('Voltage vs time dt = 0.003')
xlabel('time (s)')
ylabel('Voltage')
legend('input','output')

figure(16)
plot(0:dt2:1,ViLargeStep)
hold on
plot(0:dt2:1,VoLargeStep)
title('Voltage vs time dt = 0.01')
xlabel('time (s)')
ylabel('Voltage')
legend('input','output')
```





These results show that as the step is increased, the results of the simulation can become very erratic and unnacurate. This is because the accuracy of the finite difference approximation is inversly proportional to the step size.

Part 4 Non Linearity

If the transconductance of this device was non linear and was modeled by a thrid order polynomial, then the simulation would have to be modified in order to handle the nonlinearity. A Column Vector B(V) would be added to the left side of the matrix equation in order to deal with the non linearity. The updated system of equations would look like this.

$$V1 = Vin$$

$$G1(V2 - V1) + C1\frac{d(V2 - V1)}{dt} + Il = 0$$

$$G3V3 - Il = 0$$

$$G3V3 - I3 = 0$$

$$G4(Vo - V4) + GoVo = 0$$

$$V2 - V3 - L\frac{dIl}{dt} = 0$$

$$V4 - (aI3 + bI3^2 + cI3^3) = 0$$

This means that G(7,7) is changes to 1 and B(V)(7) = aI3 + bI3^2 + cI3^3. Because the system is non linear, it can't be solved by simple gaussian elimination. Instead the Newton Raphson algorithm must be used to solve this system. The Newton Raphson method is an iterative method that calculates the root of an equation based on an initial guess of the root as well as the value of the function and its derivative of the guess.

Published with MATLAB® R2018a