Table of Contents

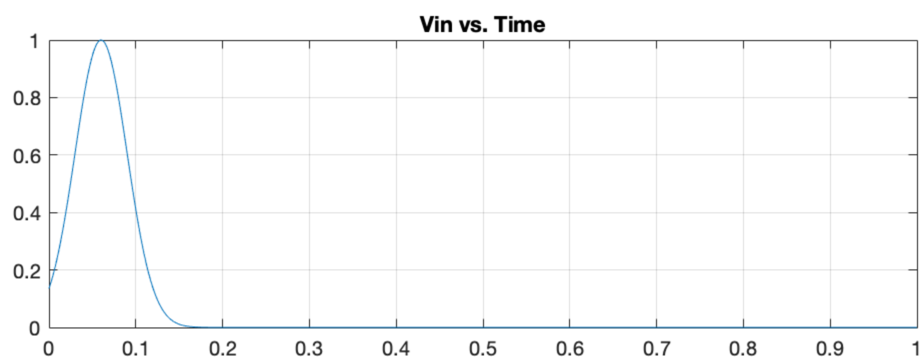
Assignment 4: Part 3

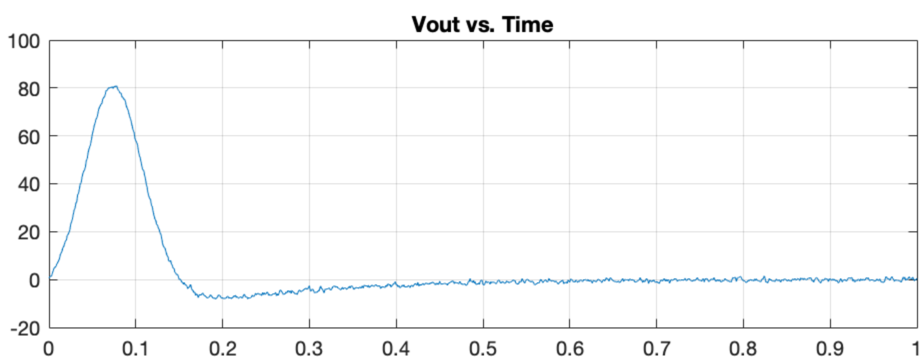
The third part of the assignemnt involved adding a current source to be modelled next to R3. The current source will generate thermal noise to be modelled. A capacitor was also modelled to be in parallel with both the newly added current source and the resistor R3. Refer to the lab manual for the updated circuit. The G, C and F matrix were then updated for the added components.

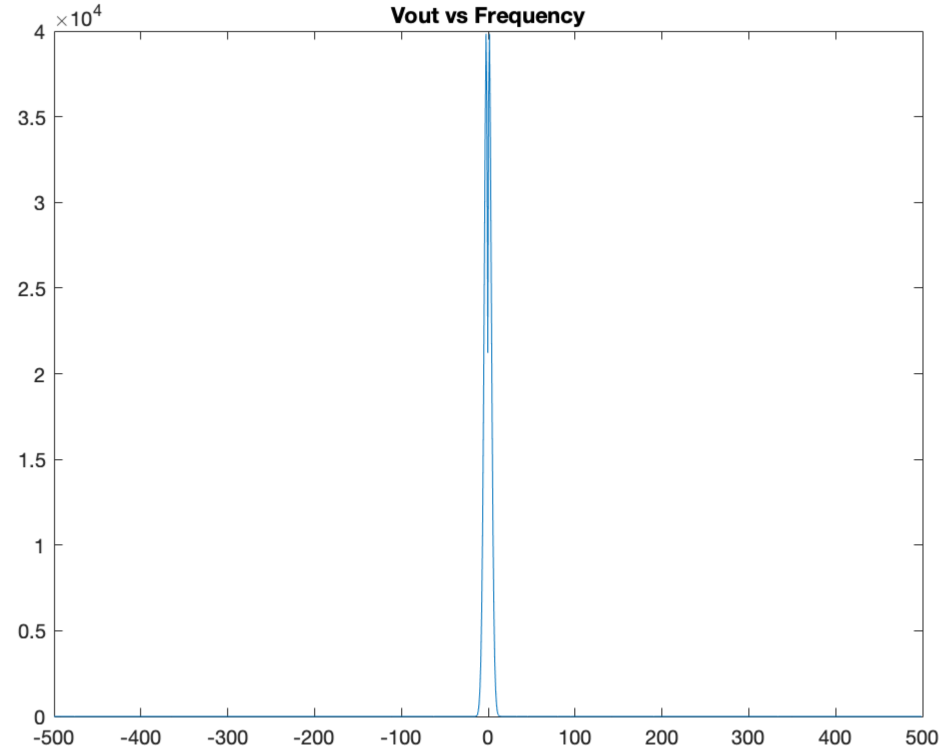
```
close all;
clear;

R1 =1;
R2 = 2;
R3 = 10;
R4 = 0.1;
R0 = 1000;
cap = 0.25;
L = 0.2;
alpha = 100;
Cn = 1E-4;

G1 = 1/R1;
G2 = 1/R2;
G3 = 1/R3;
G4 = 1/R4;
G0 = 1/R0;

G = zeros(9, 9);
C = zeros(9, 9);

G(1, 1) = -G1;
G(1, 2) =  G1;
G(2, 1) =  G1;
G(1, 3) =  G1;
G(2, 3) = -G1-G2;
G(3, 4) = -G3;
G(2, 7) = -1;
G(3, 7) = 1;
G(4, 3) = 1;
G(4, 4) = -1;
G(5, 6) = G4;
G(5, 4) = -alpha*G4;
G(5, 8) = 1;
G(6, 6) = -G4-G0;
G(6, 4) = alpha*G4;
G(8,4) = -alpha*G3;
```

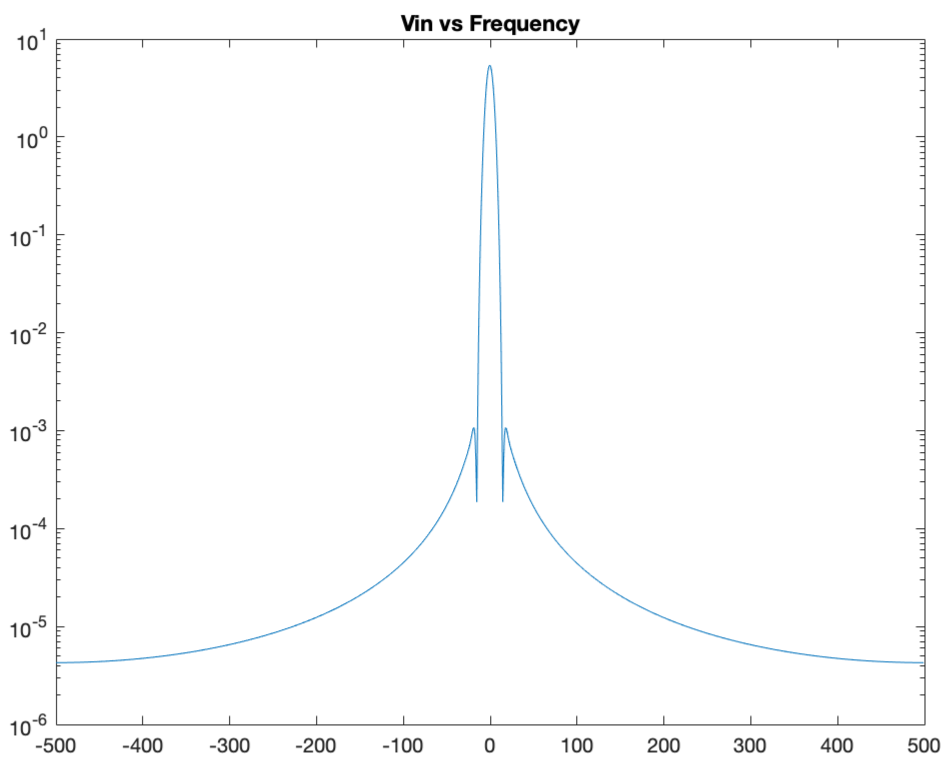
```
G(7, 1) = 1;
G(8, 5) = 1;
G(3,9) = -1;
G(9,9) = 1;

C(1,1)= -cap;
C(2,1)= cap;
C(1,3)= cap;
C(2,3)= -cap;
C(3,4) = -Cn;
C(4,7)= -L;

F = zeros(9,1);

time = 1;
number_steps = 1000;
time_step = time/number_steps;

A  = C/time_step + G;

V_old = zeros(9, 1);
V_new = zeros(9, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);

time = zeros(1000,1);

for i=0:time_step:1

    % Random noise generator
    In = randn(1)*0.001;

    time(count) = i;
    F(1,7) = Vt(i);
    F(1,9) = In;
    V_new = inv(A)*(C*V_old/time_step + F');
    V0(count) = V_new(6);
    Vin(count) = V_new(1);
    V_old = V_new;
    count = count+1;

end

figure(1)
subplot(2,1,2)
plot(time,V0)
title('Vout vs. Time')
grid on

subplot(2,1,1)
```

```
plot(time,Vin)
title('Vin vs. Time')
grid on

freq=1000;
fV0=fft(V0);
n=length(V0);

Yval=fftshift(fV0);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(2)
plot(fshift,power_shift)
title('Vout vs Frequency')

fVin=fft(Vin);
n=length(Vin);

Yval=fftshift(fVin);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(3)
semilogy(fshift, power_shift)
title('Vin vs Frequency')
```






Vout vs Frequency
×10^4
4
3.5
3
2.5
2
1.5
1
0.5
0
-500
-400
-300
-200
-100
0
100
200
300
400
500


Vin vs Frequency
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
-500
-400
-300
-200
-100
0
100
200
300
400
500

Capacitors effect on Band Width

The capacitor value was then varied to investigate the capacitors effect on the bandwidth of the circuit. The capacitor value was both raised and lowered and the resulting output and input voltages can be seen in the figures below. Three different capacitor values were chosen, 0.1, 0.001 and 0.00000001.

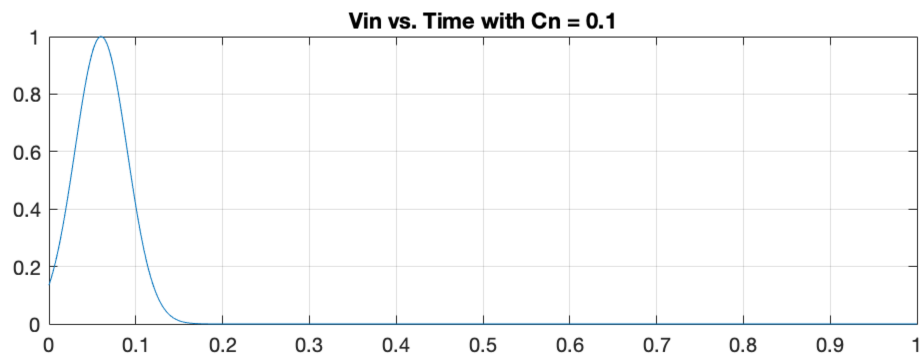
```
Cn = 0.1;
C = zeros(9,9);

C(1,1)= -cap;
C(2,1)= cap;
C(1,3)= cap;
C(2,3)= -cap;
C(3,4) = -Cn;
C(4,7)= -L;

time = 1;
number_steps = 1000;
time_step = time/number_steps;

A  = C/time_step + G;

V_old = zeros(9, 1);
V_new = zeros(9, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);

time = zeros(1000,1);

for i=0:time_step:1

    % Random noise generator
    In = randn(1)*0.001;

    time(count) = i;
    F(1,7) = Vt(i);
    F(1,9) = In;
    V_new = inv(A)*(C*V_old/time_step + F');
    V0(count) = V_new(6);
    Vin(count) = V_new(1);
    V_old = V_new;
    count = count+1;

end

figure(4)
subplot(2,1,2)
plot(time,V0)
title('Vout vs. Time with Cn = 0.1')
```

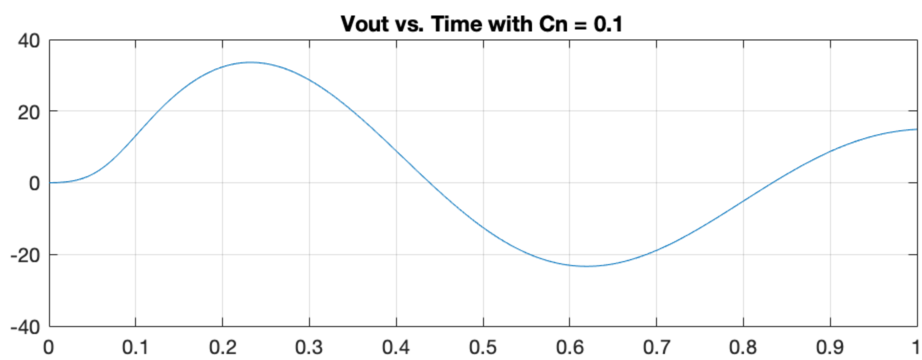
```
grid on

subplot(2,1,1)
plot(time,Vin)
title('Vin vs. Time with Cn = 0.1')
grid on

freq=1000;
fV0=fft(V0);
n=length(V0);

Yval=fftshift(fV0);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(5)
semilogy(fshift, power_shift)
title('Vout vs Frequency with Cn = 0.1')

fVin=fft(Vin);
n=length(Vin);

Yval=fftshift(fVin);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(6)
semilogy(fshift, power_shift)
title('Vin vs Frequency with Cn = 0.1')

% Second capacitor value
Cn = 0.001;
C = zeros(9,9);

C(1,1)= -cap;
C(2,1)= cap;
C(1,3)= cap;
C(2,3)= -cap;
C(3,4) = -Cn;
C(4,7)= -L;

time = 1;
number_steps = 1000;
time_step = time/number_steps;

A  = C/time_step + G;

V_old = zeros(9, 1);
V_new = zeros(9, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));
```

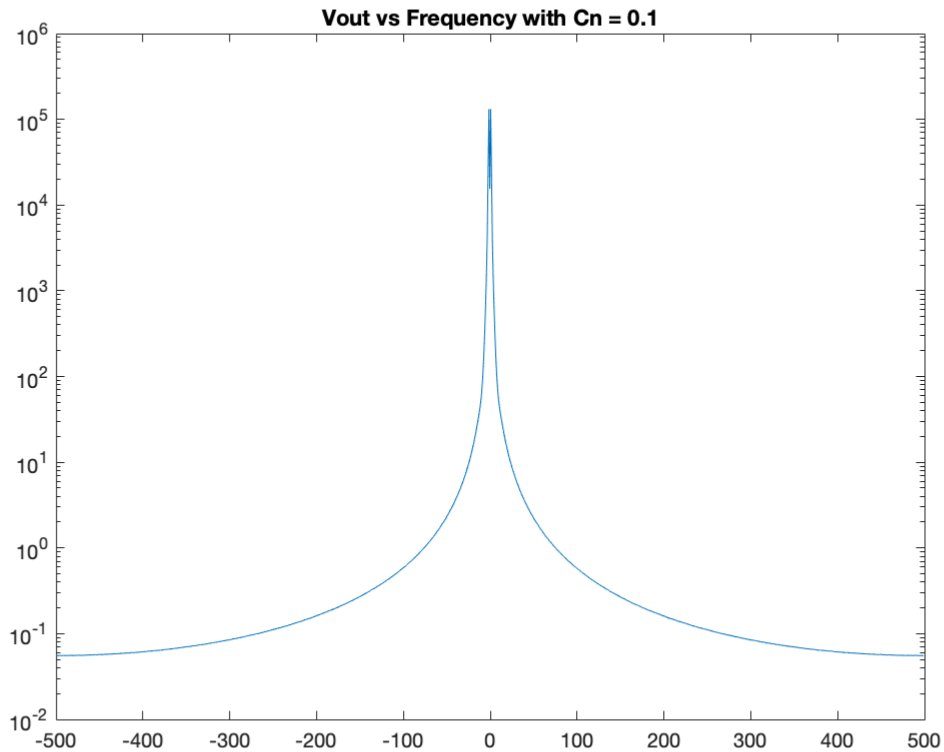
```
Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);

time = zeros(1000,1);

for i=0:time_step:1

    % Random noise generator
    In = randn(1)*0.001;

    time(count) = i;
    F(1,7) = Vt(i);
    F(1,9) = In;
    V_new = inv(A)*(C*V_old/time_step + F');
    V0(count) = V_new(6);
    Vin(count) = V_new(1);
    V_old = V_new;
    count = count+1;

end

figure(7)
subplot(2,1,2)
plot(time,V0)
title('Vout vs. Time with Cn = 0.001')
grid on

subplot(2,1,1)
plot(time,Vin)
title('Vin vs. Time with Cn = 0.001')
grid on

freq=1000;
fV0=fft(V0);
n=length(V0);

Yval=fftshift(fV0);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(8)
semilogy(fshift, power_shift)
title('Vout vs Frequency with Cn = 0.001')

fVin=fft(Vin);
n=length(Vin);

Yval=fftshift(fVin);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(9)
semilogy(fshift, power_shift)
title('Vin vs Frequency with Cn = 0.001')
```

```
%Third Capacitor Value

Cn = 1E-7;
C = zeros(9,9);

C(1,1)= -cap;
C(2,1)= cap;
C(1,3)= cap;
C(2,3)= -cap;
C(3,4) = -Cn;
C(4,7)= -L;

time = 1;
number_steps = 1000;
time_step = time/number_steps;

A  = C/time_step + G;

V_old = zeros(9, 1);
V_new = zeros(9, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);

time = zeros(1000,1);

for i=0:time_step:1

    % Random noise generator
    In = randn(1)*0.001;

    time(count) = i;
    F(1,7) = Vt(i);
    F(1,9) = In;
    V_new = inv(A)*(C*V_old/time_step + F');
    V0(count) = V_new(6);
    Vin(count) = V_new(1);
    V_old = V_new;
    count = count+1;

end

figure(10)
subplot(2,1,2)
plot(time,V0)
title('Vout vs. Time with Cn = 0.00000001')
grid on

subplot(2,1,1)
plot(time,Vin)
```

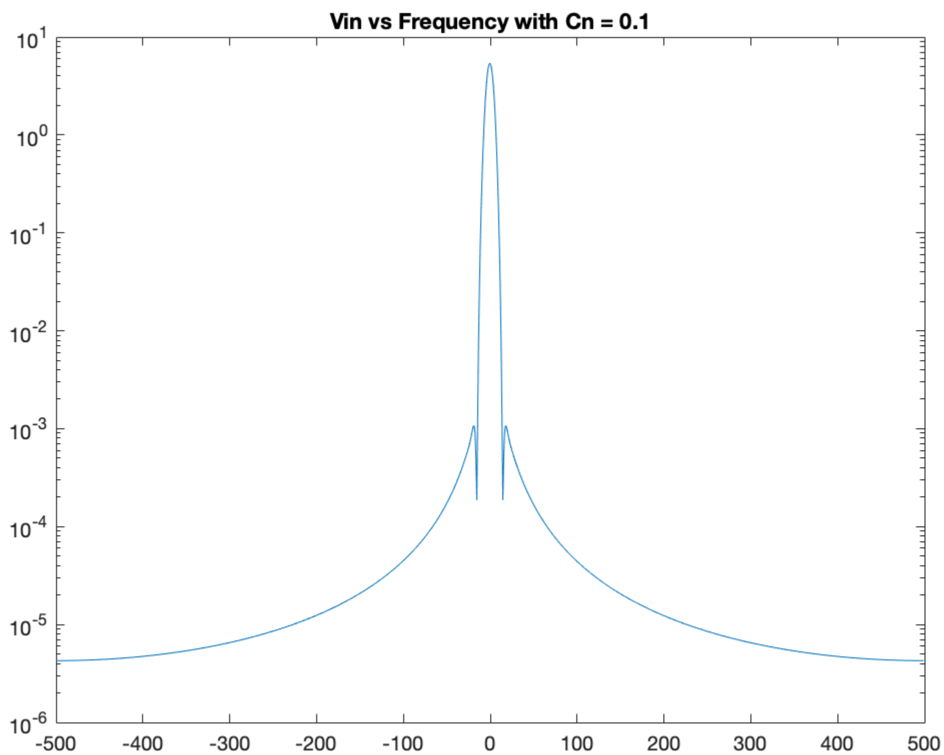
```
title('Vin vs. Time with Cn = 0.00000001')
grid on

freq=1000;
fV0=fft(V0);
n=length(V0);

Yval=fftshift(fV0);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(11)
semilogy(fshift, power_shift)
title('Vout vs Frequency with Cn = 0.00000001')

fVin=fft(Vin);
n=length(Vin);

Yval=fftshift(fVin);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(12)
semilogy(fshift, power_shift)
title('Vin vs Frequency with Cn = 0.00000001')
```



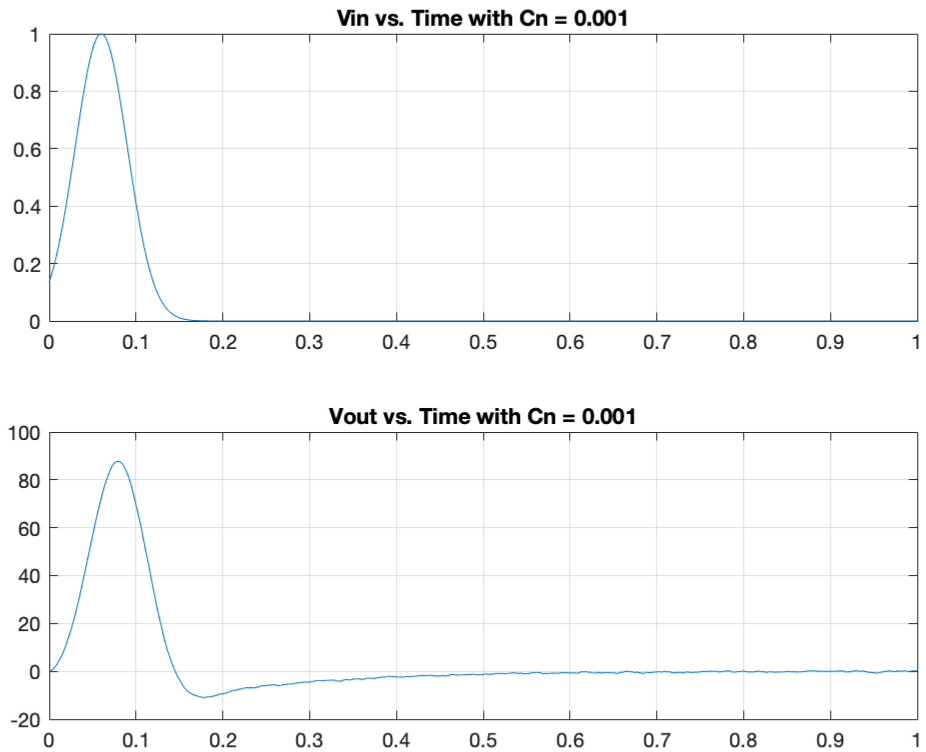


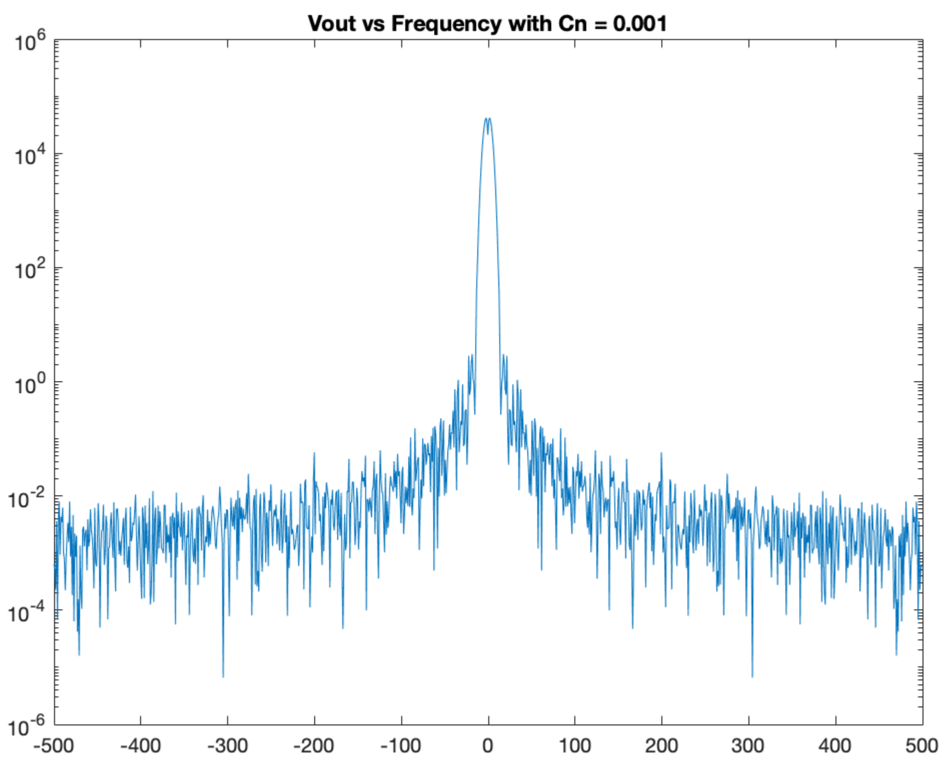


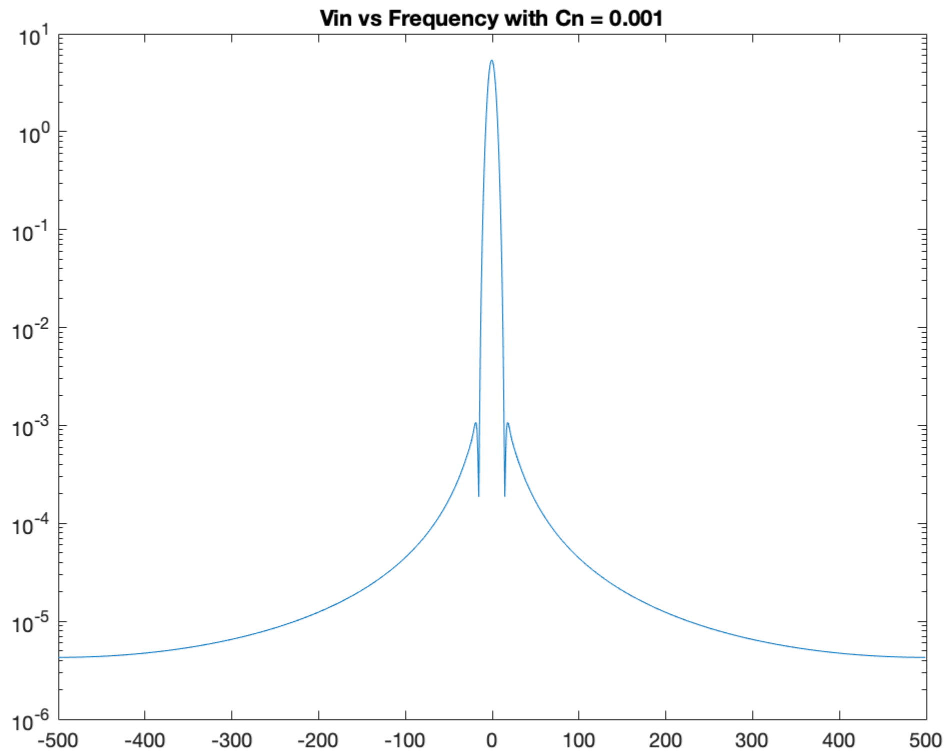
Vout vs Frequency with Cn = 0.1
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
-500
-400
-300
-200
-100
0
100
200
300
400
500

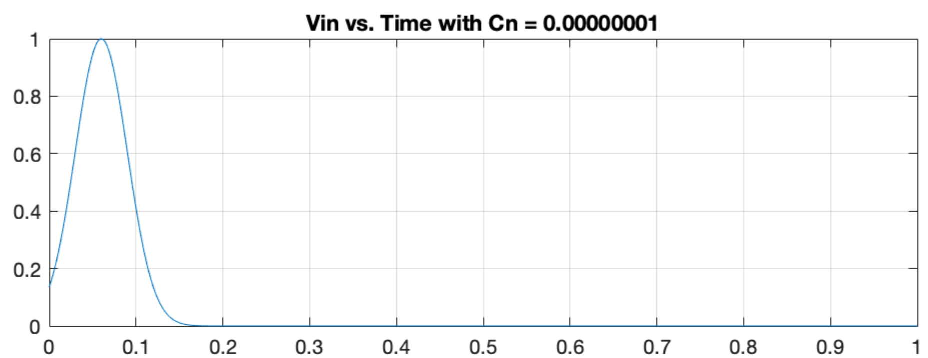


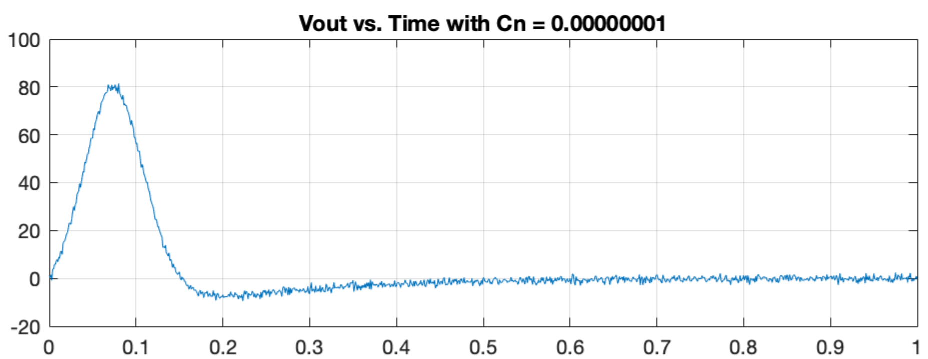
Vin vs Frequency with Cn = 0.1
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
-500
-400
-300
-200
-100
0
100
200
300
400
500

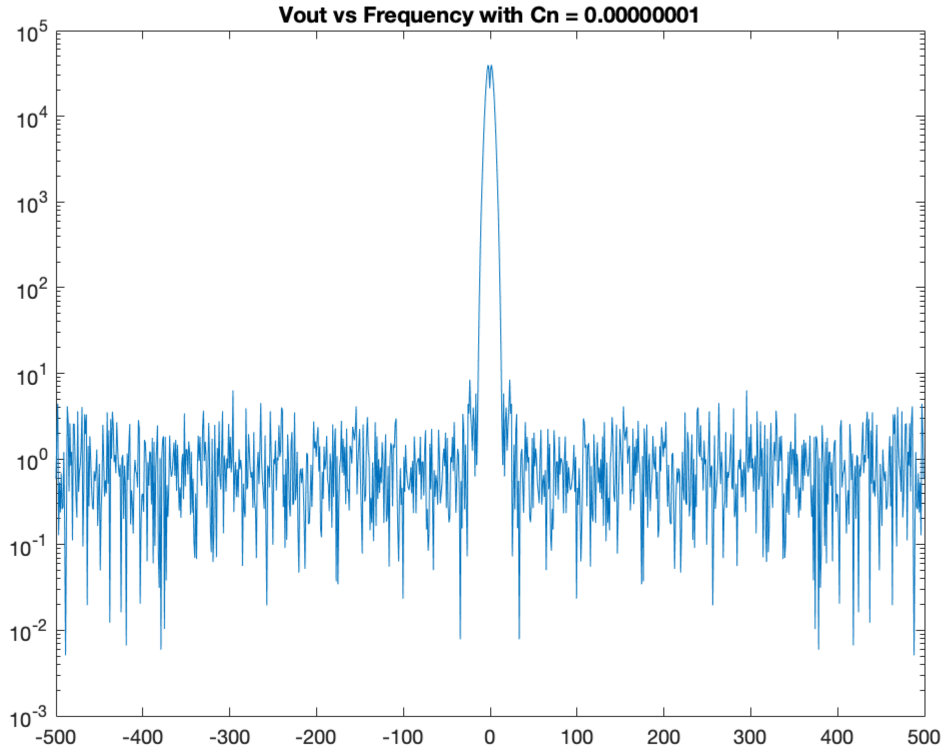

Vin vs. Time with Cn = 0.001
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Vout vs. Time with Cn = 0.001
100
80
60
40
20
0
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

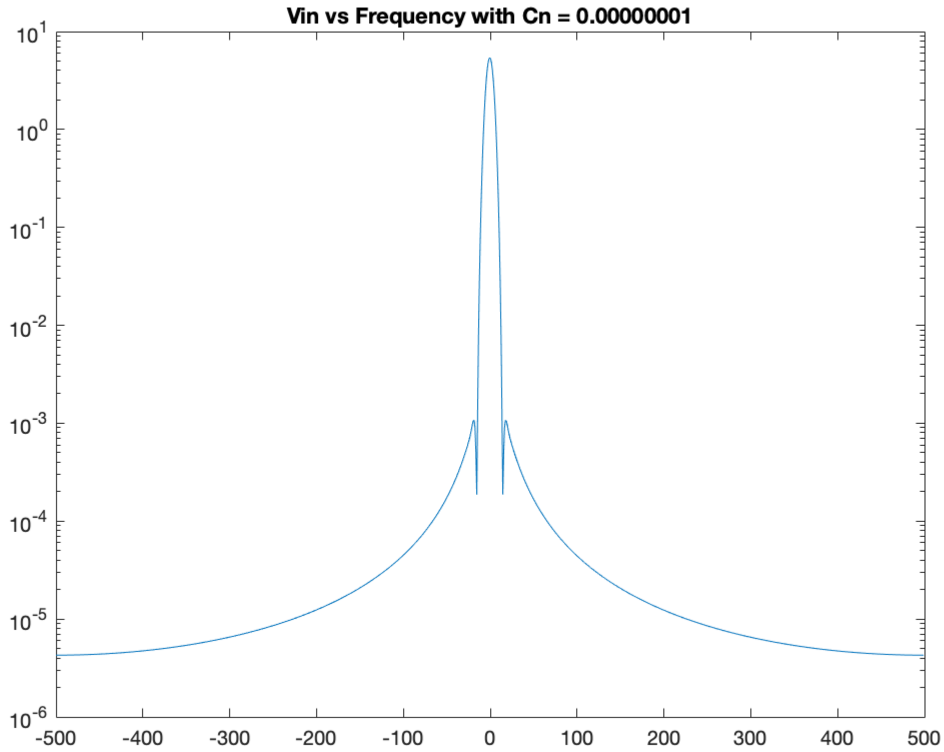

Vout vs Frequency with Cn = 0.001
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
-500
-400
-300
-200
-100
0
100
200
300
400
500


Vin vs Frequency with Cn = 0.001
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
-500
-400
-300
-200
-100
0
100
200
300
400
500


Vin vs. Time with Cn = 0.00000001
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Vout vs. Time with Cn = 0.00000001
100
80
60
40
20
0
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Vout vs Frequency with Cn = 0.00000001
10^5
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
-500
-400
-300
-200
-100
0
100
200
300
400
500


Vin vs Frequency with Cn = 0.00000001
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
-500
-400
-300
-200
-100
0
100
200
300
400
500

The smaller capacitor value produced an output voltage -3dB band width of approximately 10 Hz, taken from the plot. The largest of the capacitance values, 0.1F, produced a -dB bandwidth of approximately 4Hz. This result suggests that increaseing the capacitor value reduces the bandwidth of the system.

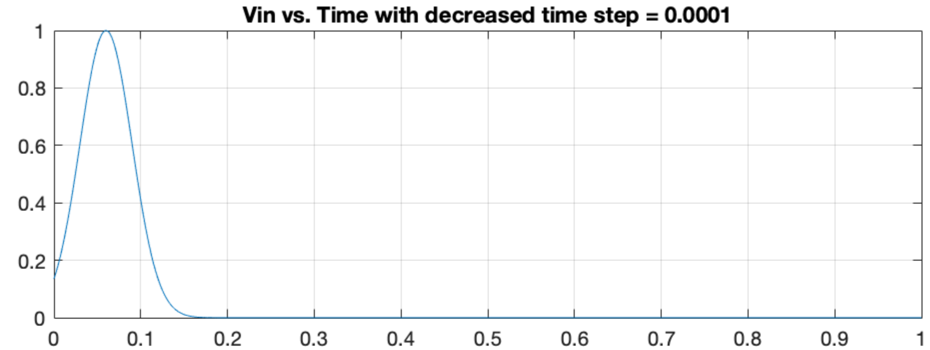
Varying the Time Step

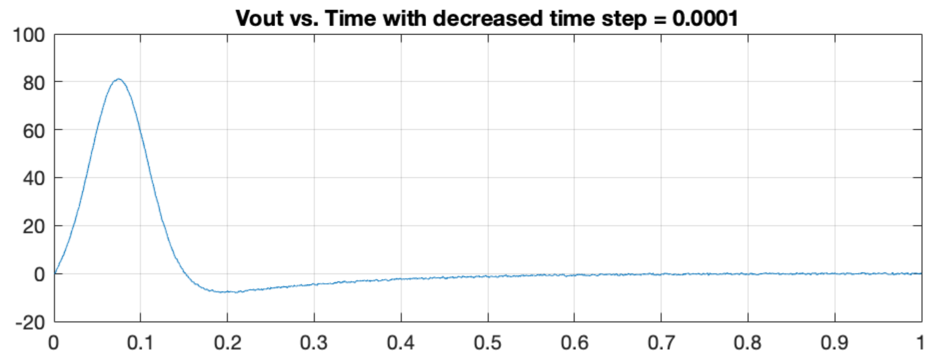
Finally, the time step was varied to determine how this would alter the simulation. The time step was both increased and decreased, with the resulting simulations displayed below.

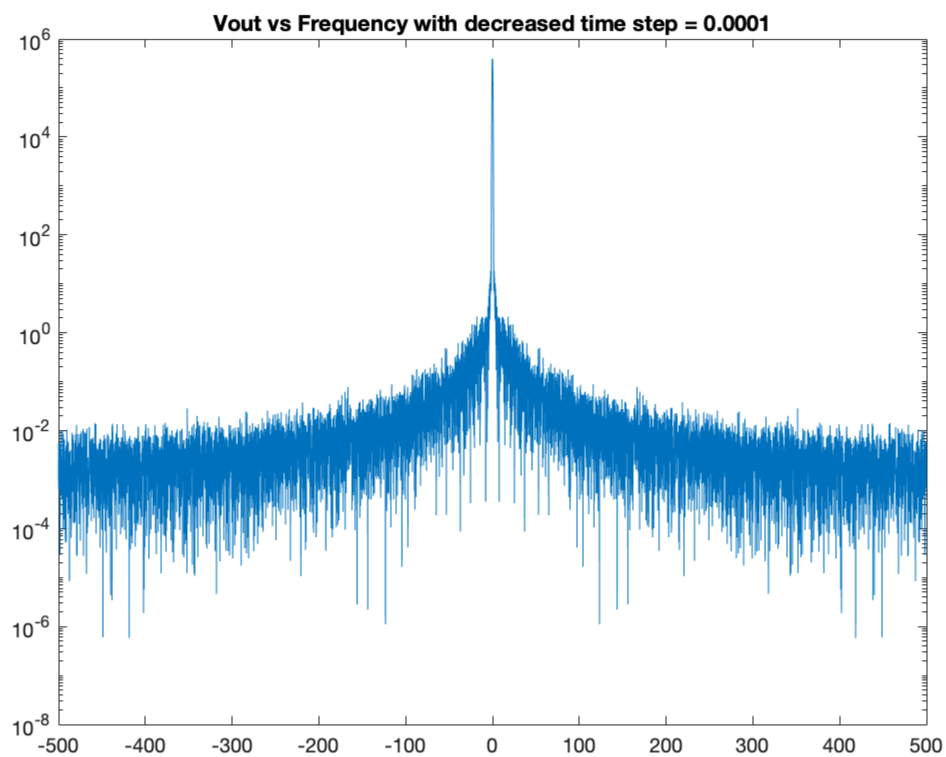
The time step was first decreased, the result can be seen in the figures below.

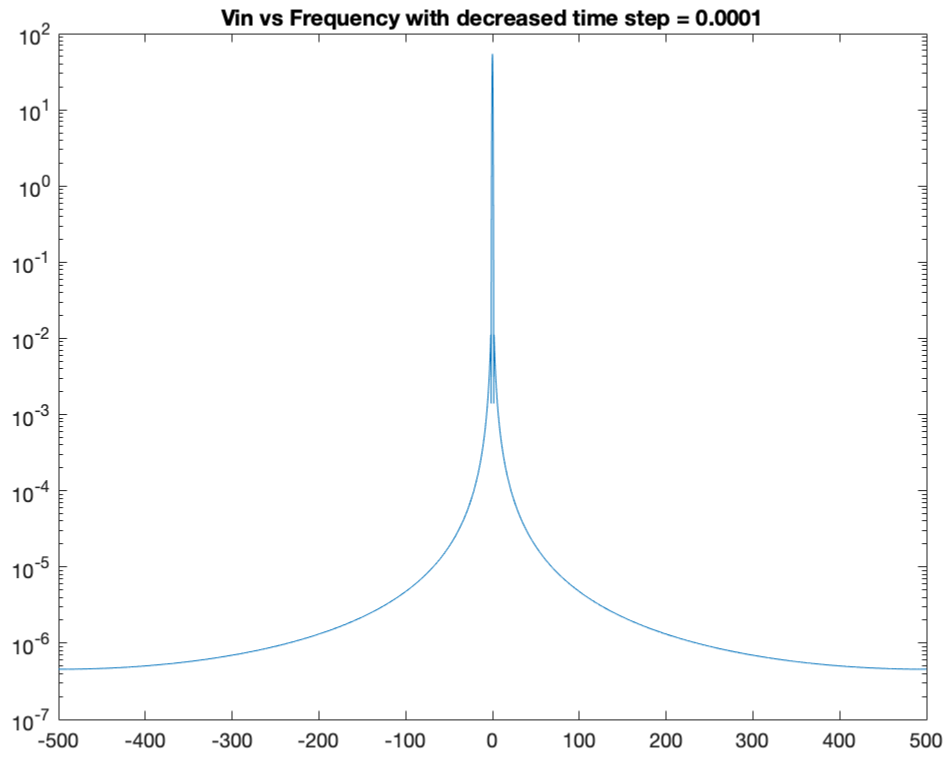
```
Cn = 1E-4;
C = zeros(9,9);

C(1,1)= -cap;
C(2,1)= cap;
C(1,3)= cap;
C(2,3)= -cap;
C(3,4) = -Cn;
C(4,7)= -L;

time = 1;
number_steps = 10000;
time_step = time/number_steps;

A  = C/time_step + G;

V_old = zeros(9, 1);
V_new = zeros(9, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);

time = zeros(10000,1);

for i=0:time_step:1

    % Random noise generator
    In = randn(1)*0.001;

    time(count) = i;
    F(1,7) = Vt(i);
    F(1,9) = In;
    V_new = inv(A)*(C*V_old/time_step + F');
    V0(count) = V_new(6);
    Vin(count) = V_new(1);
    V_old = V_new;
    count = count+1;

end
```

```
figure(13)
subplot(2,1,2)
plot(time,V0)
title('Vout vs. Time with decreased time step = 0.0001')
grid on

subplot(2,1,1)
plot(time,Vin)
title('Vin vs. Time with decreased time step = 0.0001')
grid on

freq=1000;
fV0=fft(V0);
n=length(V0);

Yval=fftshift(fV0);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(14)
semilogy(fshift, power_shift)
title('Vout vs Frequency with decreased time step = 0.0001')

fVin=fft(Vin);
n=length(Vin);

Yval=fftshift(fVin);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(15)
semilogy(fshift, power_shift)
title('Vin vs Frequency with decreased time step = 0.0001')
```


Vin vs. Time with decreased time step = 0.0001
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Vout vs. Time with decreased time step = 0.0001
100
80
60
40
20
0
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

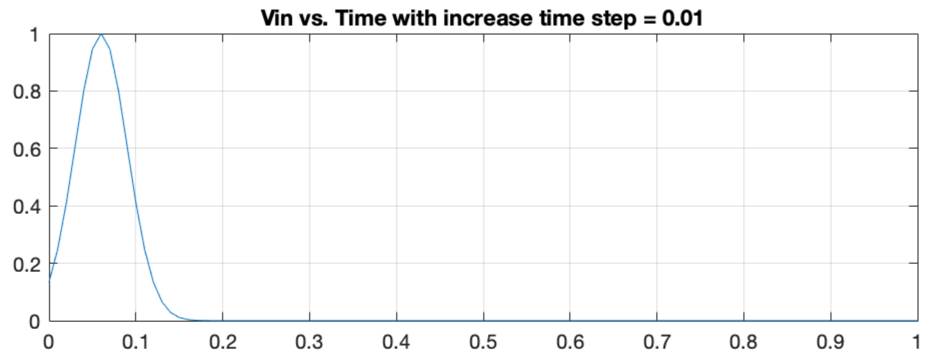

Vout vs Frequency with decreased time step = 0.0001
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
-500
-400
-300
-200
-100
0
100
200
300
400
500

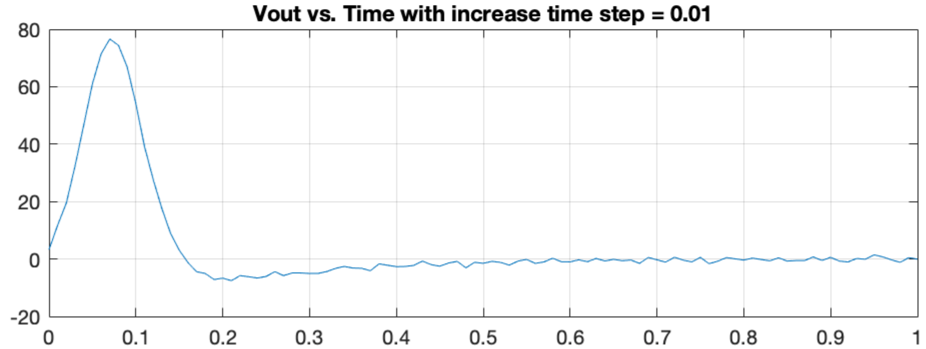


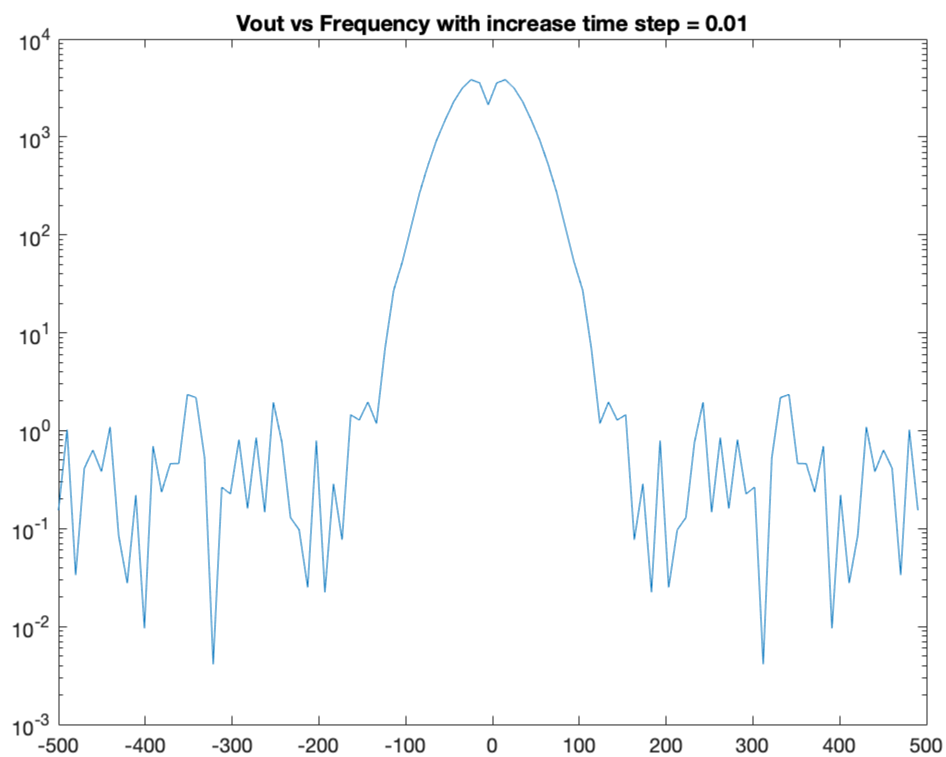
The time step was then increased, the results can be seen in the figure below.

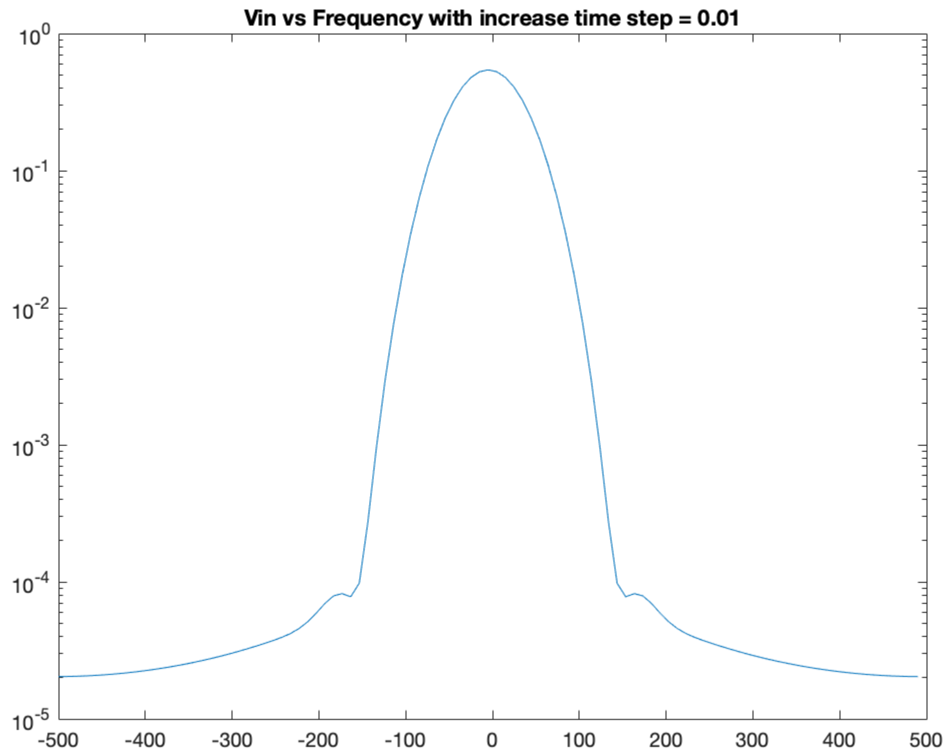
```
Cn = 1E-4;
C = zeros(9,9);

C(1,1)= -cap;
C(2,1)= cap;
C(1,3)= cap;
C(2,3)= -cap;
C(3,4) = -Cn;
C(4,7)= -L;

time = 1;
number_steps = 100;
time_step = time/number_steps;

A  = C/time_step + G;

V_old = zeros(9, 1);
V_new = zeros(9, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);
```

```
time = zeros(100,1);

for i=0:time_step:1

    % Random noise generator
    In = randn(1)*0.001;

    time(count) = i;
    F(1,7) = Vt(i);
    F(1,9) = In;
    V_new = inv(A)*(C*V_old/time_step + F');
    V0(count) = V_new(6);
    Vin(count) = V_new(1);
    V_old = V_new;
    count = count+1;

end

figure(16)
subplot(2,1,2)
plot(time,V0)
title('Vout vs. Time with increase time step = 0.01')
grid on

subplot(2,1,1)
plot(time,Vin)
title('Vin vs. Time with increase time step = 0.01')
grid on

freq=1000;
fV0=fft(V0);
n=length(V0);

Yval=fftshift(fV0);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(17)
semilogy(fshift, power_shift)
title('Vout vs Frequency with increase time step = 0.01')

fVin=fft(Vin);
n=length(Vin);

Yval=fftshift(fVin);
fshift = (-n/2:n/2-1)*(freq/n);
power_shift = abs(Yval).^2/n;

figure(18)
semilogy(fshift, power_shift)
title('Vin vs Frequency with increase time step = 0.01')
```


Vin vs. Time with increase time step = 0.01
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Vout vs. Time with increase time step = 0.01
80
60
40
20
0
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Vout vs Frequency with increase time step = 0.01
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
-500
-400
-300
-200
-100
0
100
200
300
400
500



Altering the time steps produces drastically different results, specifically in the frequency domain. With more time steps, the freqency specturm is much larger and therefore the plot is composed of more accurate frequencies. For the time domain, the larger time step resulted in small alterations in the output voltage signal. It is advantageous to use a smaller time step to produce a more accurate result.

Assignment 4: Part 4

The fourth part of the assignment discusses the implementation of an non-linear voltage generator. Through team research, it was determined the jacobian method could be used to solve for the current controller voltage source. To solve for the voltage during each iteration, the Newton Ralphson method can be implemented. Similarly to the rest of the simulations in the assignment, a voltage matrix will be created and solved for to estimate the output voltage.

Published with MATLAB® R2017a