Table of Contents

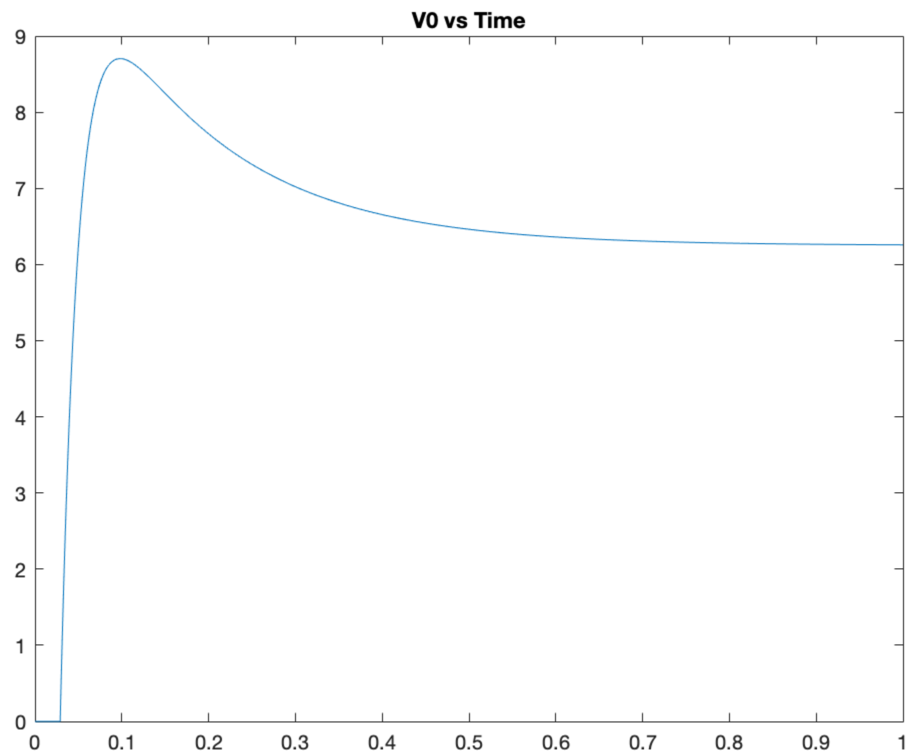
Assignment 4: Part 2

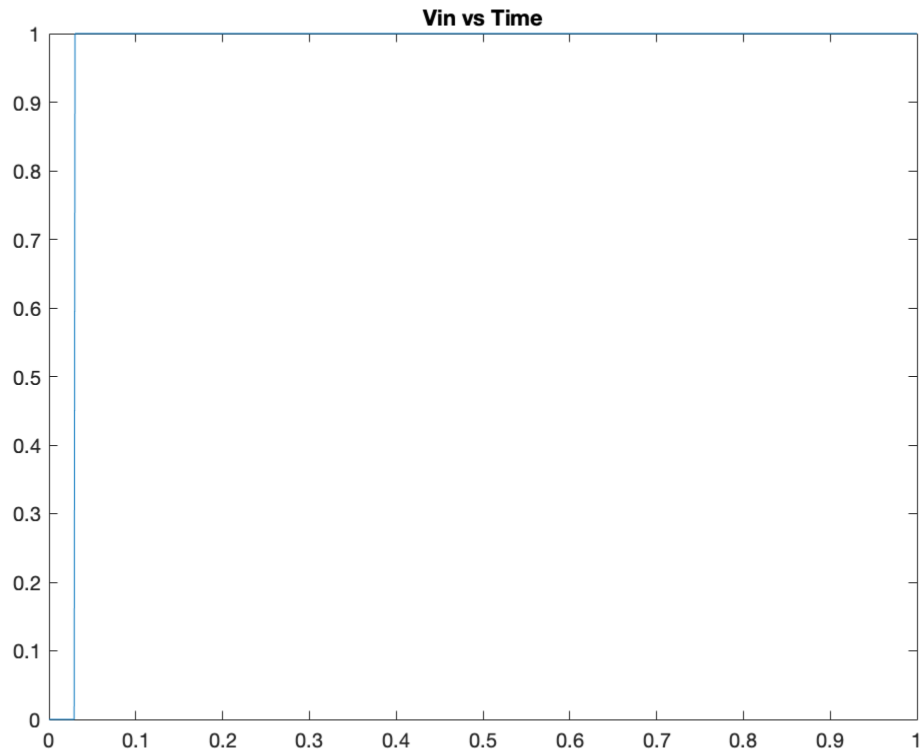
The second part of the assignment involved implementing a model for a low pass filter circuit. A low pass filter is aimed to provide gain to signals at low frequencies and attenuate signals at higher frequencies. The drop off point is defined by the capacitor in the system. The C, G and F matricies need to be redefined for the simulation. A DC sweep was conudcted by inputing a unit step function. This can be seen in the figures below.

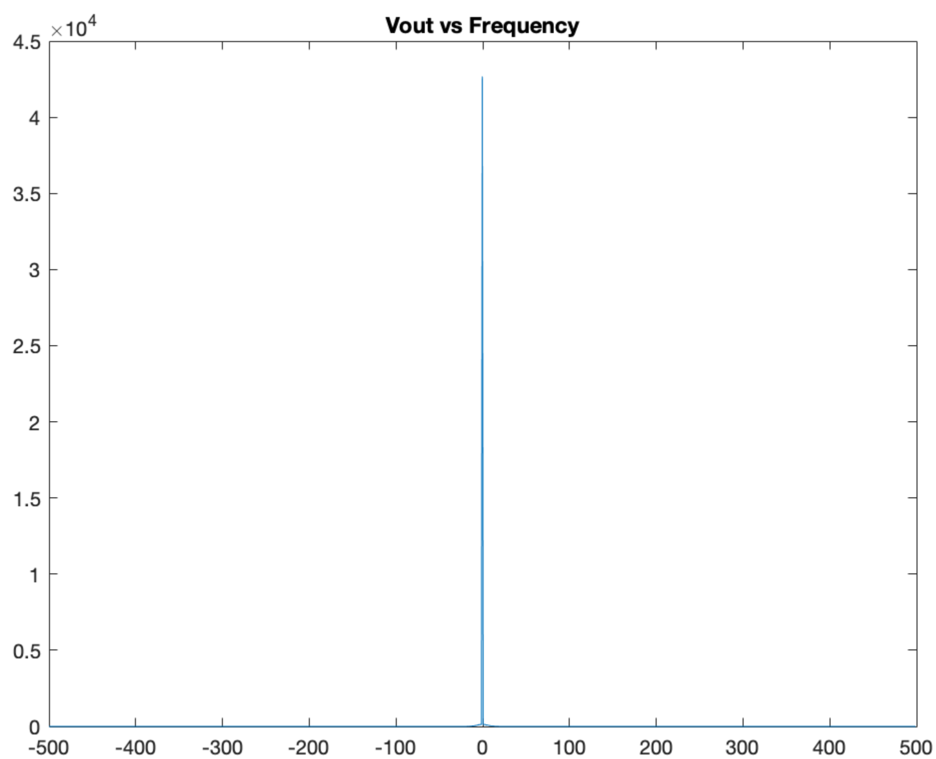
```
close all;
clear;

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

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

G = zeros(8, 8);
C = zeros(8, 8);

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, 7) = -alpha*G4;
G(5, 8) = 1;
G(6, 6) = -G4-G0;
G(6, 7) = alpha*G4;
```

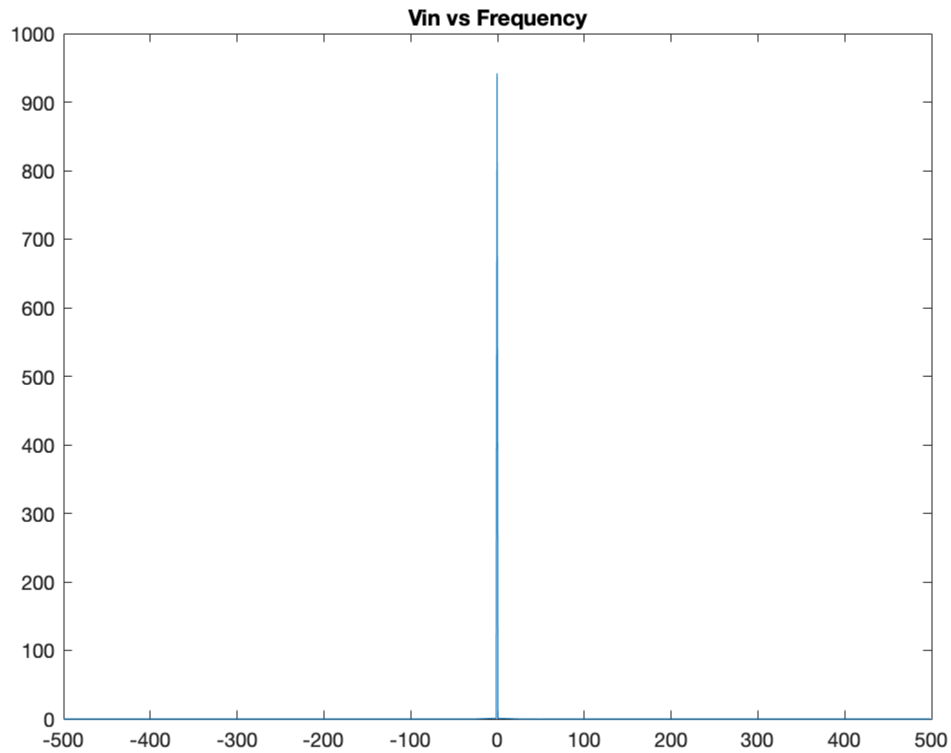
```
G(7, 1) = 1;
G(8, 5) = 1;
G(8, 7) = -alpha;

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

F = zeros(8,1);

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

A  = C/time_step + G;

V_old = zeros(8, 1);
V_new = zeros(8, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

for i=0:time_step:1
    if (i >= 0.03)
        F=[0 0 0 0 0 0 1 0];
    else
        F=[0 0 0 0 0 0 0 0];
    end

    time(count) = i;
    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)
plot(time,V0)
title('V0 vs Time')

figure(2)
plot(time,Vin)
title('Vin vs Time')

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(3)
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(4)
plot(fshift,power_shift)
title('Vin vs Frequency')
```




Vin vs Time
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Vout vs Frequency
×10^4
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-500
-400
-300
-200
-100
0
100
200
300
400
500

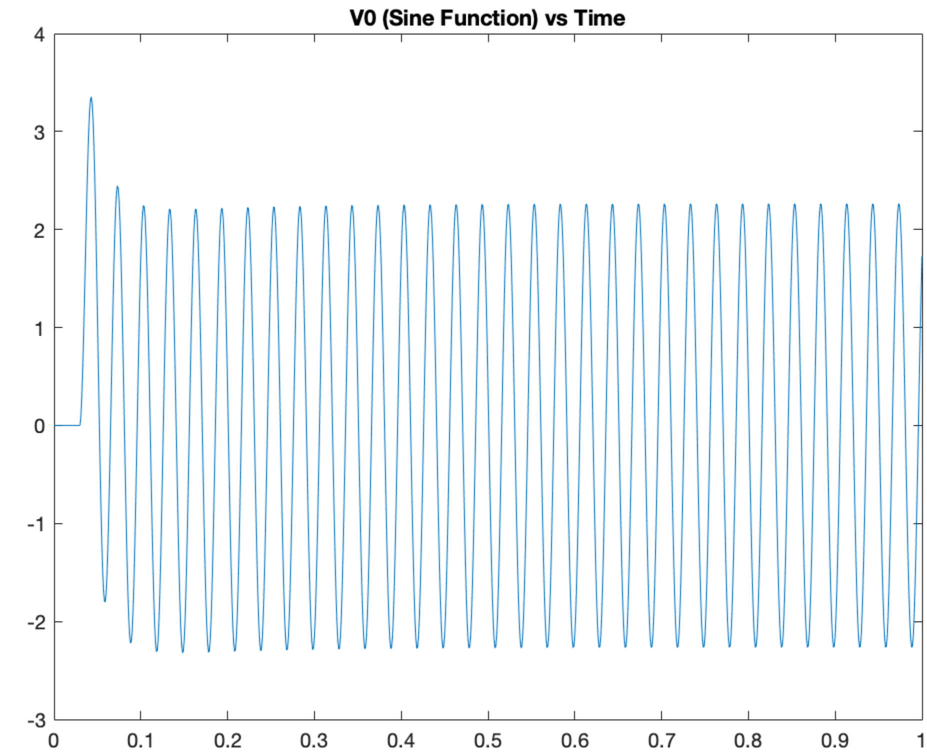


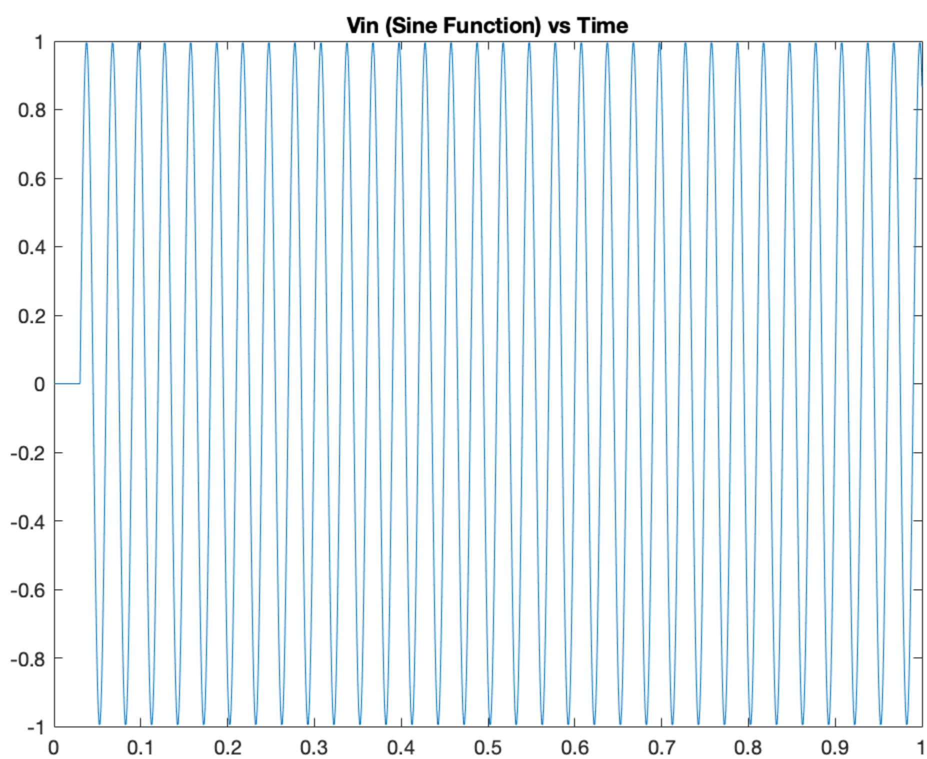
Sine Wave Signal

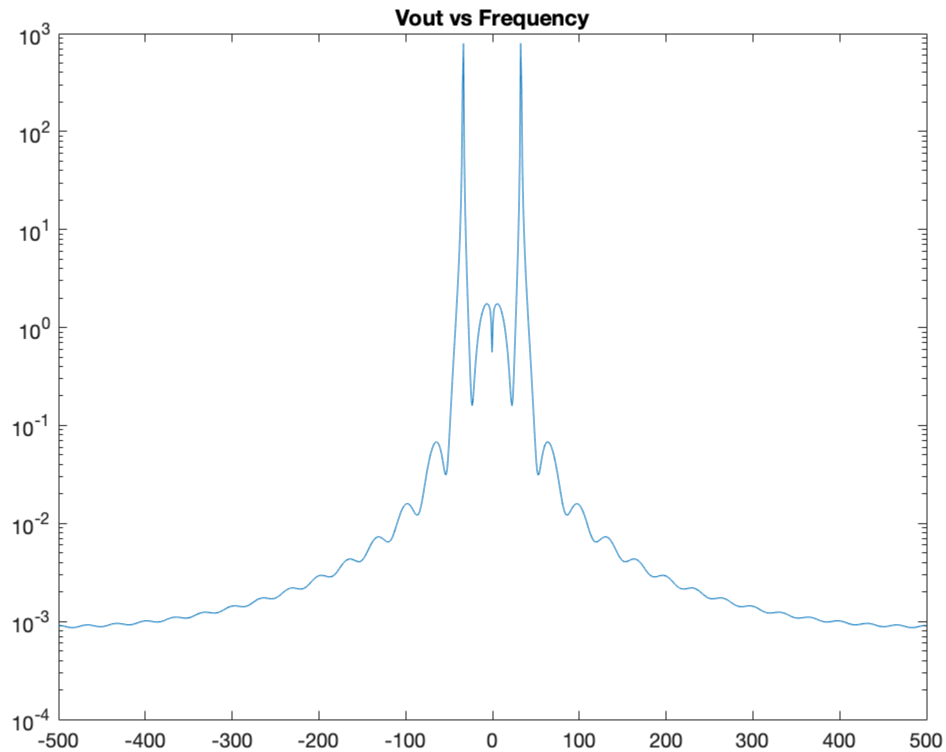
An AC signal was then input into the system. A sine function was defined and used as the input signal. The resulting plots can be seen in the plots below. Included are the input and output voltages is both the time domain and the frequency domain.

```
f = 1/0.03;
input = @(i) sin(2*pi*f*i);

V_old = zeros(8, 1);
V_new = zeros(8, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

for i=0:time_step:1
    if (i >= 0.03)
        F=[0 0 0 0 0 0 input(i) 0];
    else
        F=[0 0 0 0 0 0 0 0];
    end

    time(count) = i;
    V_new = inv(A)*(C*V_old/time_step + F');
    V0(count) = V_new(6);
```

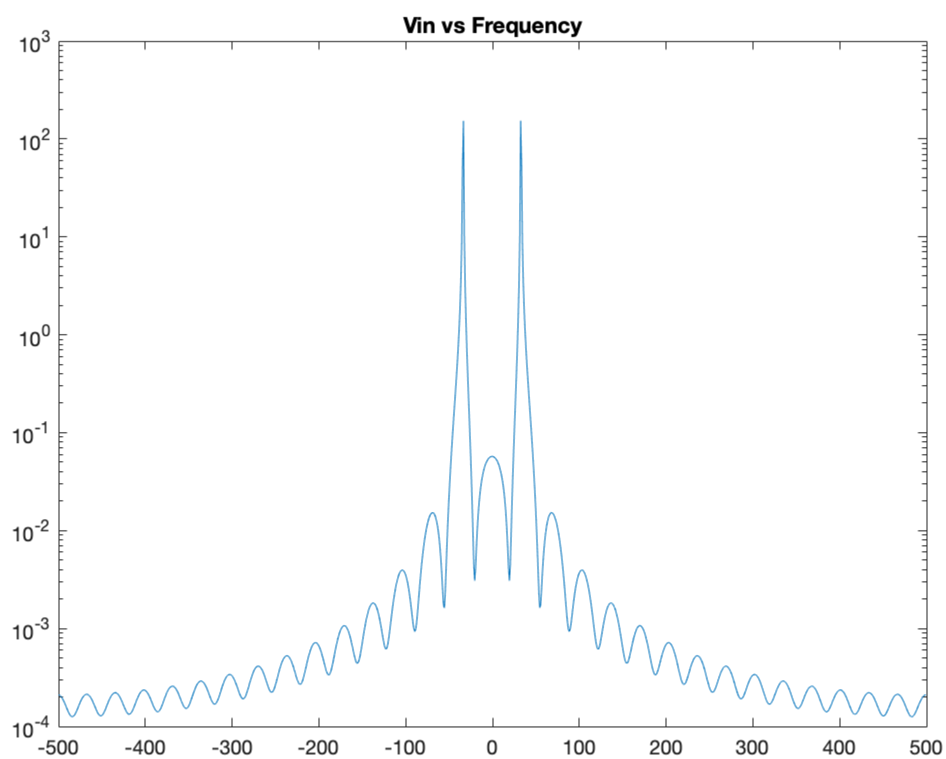
```
    Vin(count)=V_new(1);
    V_old=V_new;
    count = count + 1;
end

figure(5)
plot(time,V0)
title('V0 (Sine Function) vs Time')

figure(6)
plot(time,Vin)
title('Vin (Sine Function) vs Time')

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(7)
semilogy(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(8)
semilogy(fshift, power_shift)
title('Vin vs Frequency')
```


V0 (Sine Function) vs Time
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


Vin (Sine Function) vs Time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Vout vs Frequency
10^3
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
-500
-400
-300
-200
-100
0
100
200
300
400
500



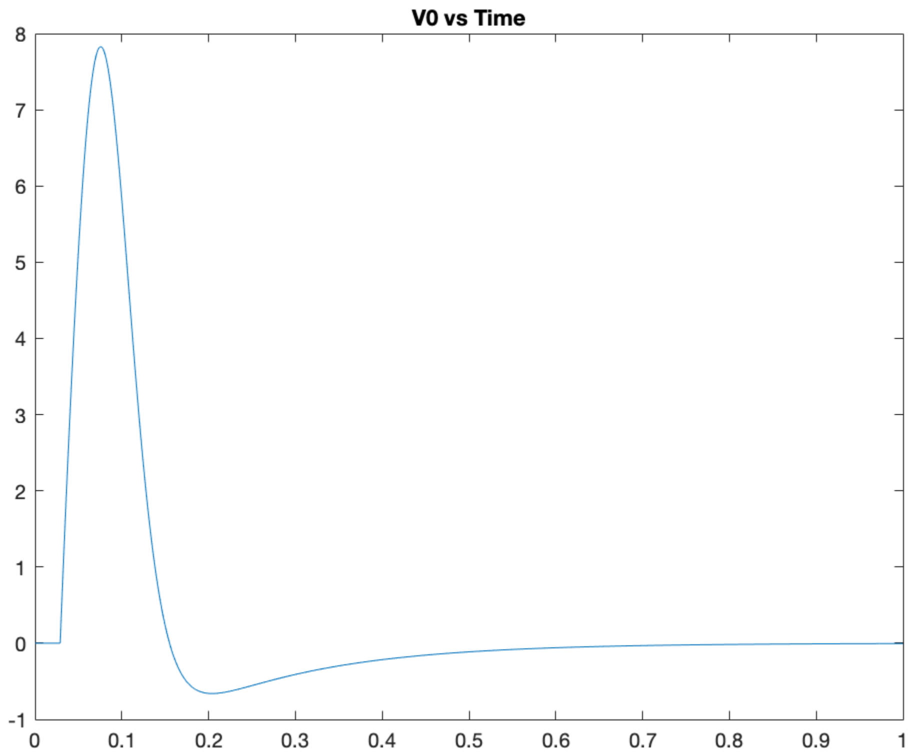
Vin vs Frequency
10^3
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
-500
-400
-300
-200
-100
0
100
200
300
400
500

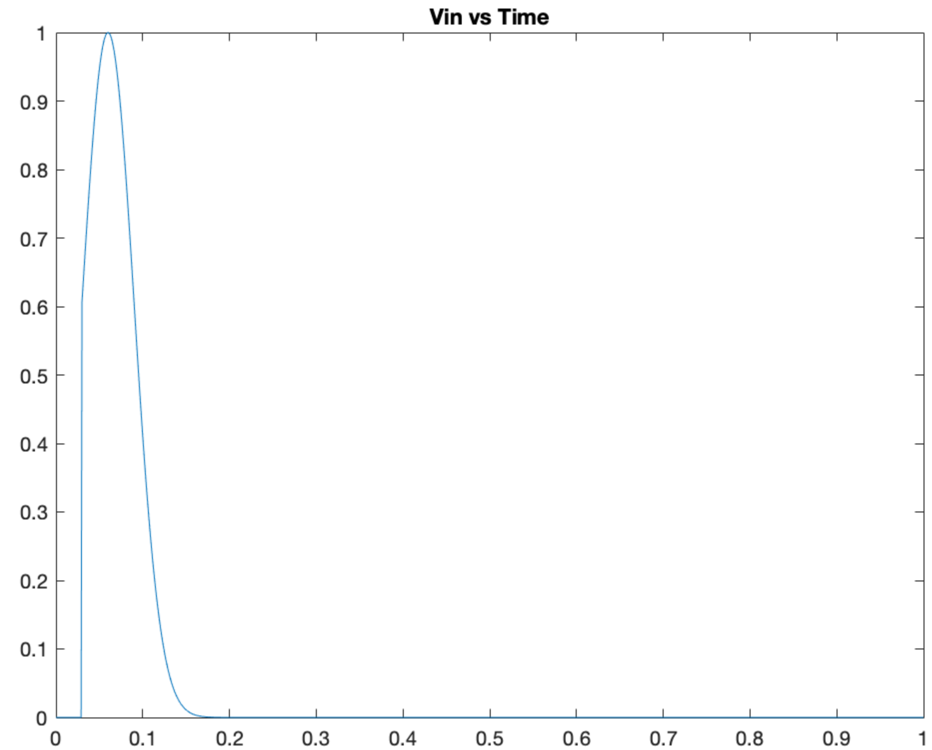
Gaussian Pulse Signal

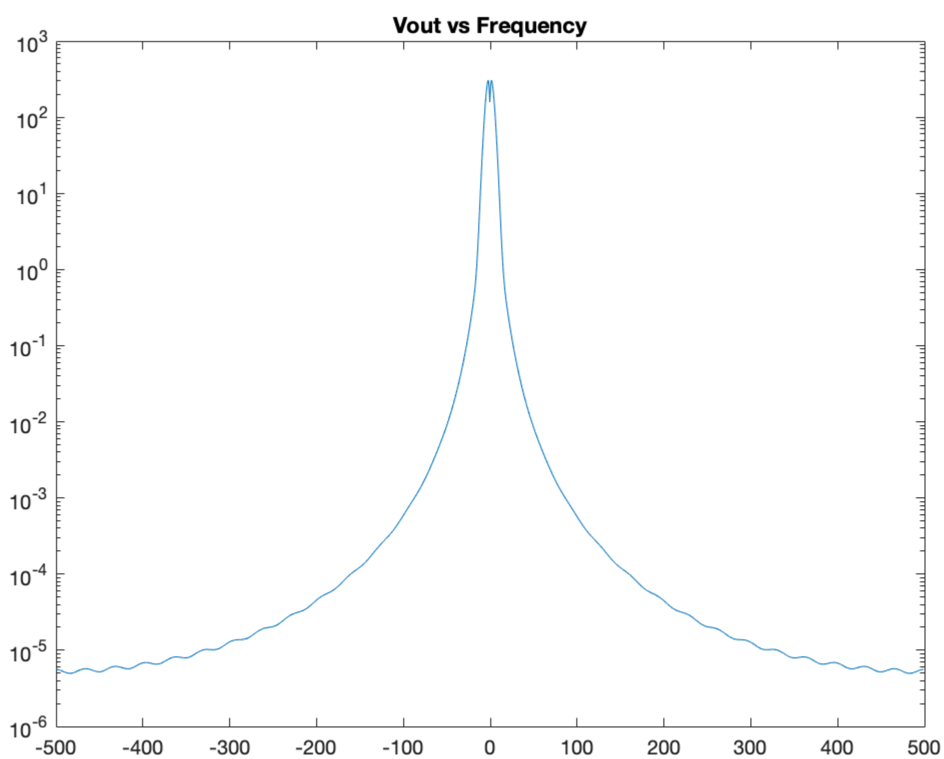
A gaussian pulse was then used as an input into the system. The function was defined and served as the input signal. The resulting plots can be seen in the figures below.

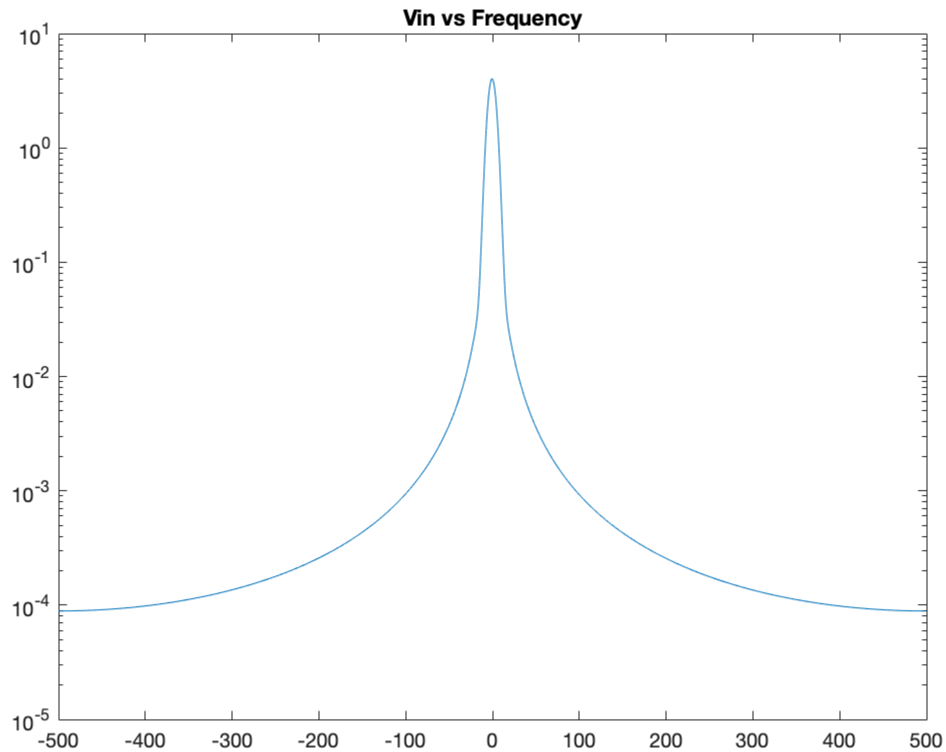
```
input =@(i) exp(-(1/2)*((i-0.06)/(0.03))^2);
V_old = zeros(8, 1);
V_new = zeros(8, 1);
count = 1;
time = zeros(1, length(time_step));
V0 = zeros(1, length(time_step));
Vin = zeros(1, length(time_step));

for i=0:time_step:1
    if (i >= 0.03)
        F=[0 0 0 0 0 0 input(i) 0];
    else
        F=[0 0 0 0 0 0 0 0];
    end

    time(count) = i;
    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(9)
plot(time,V0)
title('V0 vs Time')

figure(10)
plot(time,Vin)
title('Vin vs Time')

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')

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')
```




Vin vs Time
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Vout vs Frequency
10^{3}
10^{2}
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



Published with MATLAB® R2017a