## Table of Contents



ELEC 4700 Assignment 4 W2019 Baldeep Kooner 101004107

# Part 1: Diff. Eq & Matrix Formulation

```
%{

N1:     i1 + i2 = 0
        i1 + [(V1-V2)/R1 + C(d(V1-V2)/dt)] = 0
        V1 = Vin (1)
N2:    i2 + i3 + i4 = 0
       [(Vin-V2)/R1 + C(d(Vin-V2)/dt)] + V2/R2 + iL (2)
N3:    iL + i3 = 0
       iL + V3/R3 = 0 (3)
N4:    alpha*i3 + i4 = 0
       alpha*i3 + (V4-V5)/R4 = 0 (4)
       V4 = alpha*i3 (5)
N5:    i4 + i0 = 0
       (V5-V4)/R4 + V0/R0 = 0 (6)

N1:
       i1 + [(V1-V2)G1 + (V1-V2)sC] = 0 (1)
       V1 = Vin (2)
N2:
       [(V2-V1)G1 + (V2-V1)sC] + V2G2 + (V2-V3)sL (3)
N3:
       (V3-V2)sL + V3G3 = 0 (4)
N4:
       alpha*i3 + (V4-V5)G4 = 0 (5)
       V4 = alpha*i3 (6)
N5:
       (V5-V4)G4 + V0G0 = 0 (7)

Where s = jw

    V1,     V2, IL,    V3,       V4,    Vo  <-- V matrix

G:
      1,      0,  0,      0,        0,     0
    -G1, G1+G2,  0,      0,        0,     0
      0,      1,  0,    -1,        0,     0
      0,      0, -1,     G3,       0,     0
      0,      0,  0, -a*G3,        1,     0
      0,      0,  0,      0,     -G4, G4+GO
```

```
C:
      0,        0,  0,  0,      0,    0,
     -C,        C,  0,  0,      0,    0,
      0,        0,  L,  0,      0,    0,
      0,        0,  0,  0,      0,    0,
      0,        0,  0,  0,      0,    0,
      0,        0,  0,  0,      0,    0,

%}

clear all
close all
clc

G = zeros(6, 6);

%Resistances:
R1 = 1;
R2 = 2;
R3 = 10;
R4 = 0.1;
R0 = 1000;

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

%Additional Parameters:
alpha = 100;
Cval = 0.25;
L = 0.2;
vin = zeros(1, 20);
vo = zeros(1, 20);
v3 = zeros(1, 20);

G(1, 1) = 1;                                            % 1
G(2, 1) = -G1; G(2, 2) = G1 + G2;                       % 2
G(3 ,2) = -1; G(3, 4) = 1;                              % iL
G(4, 3) = -1; G(4, 4) = G3;                             % 3
G(5, 5) = 1; G(5, 4) = -alpha*G3;                       % 4
G(6, 6) = G4 + G0; G(6, 5) = -G4;                       % 5

C = zeros(6);

C(2, 1) = -Cval; C(2, 2) = Cval;
C(3, 3) = L;
```

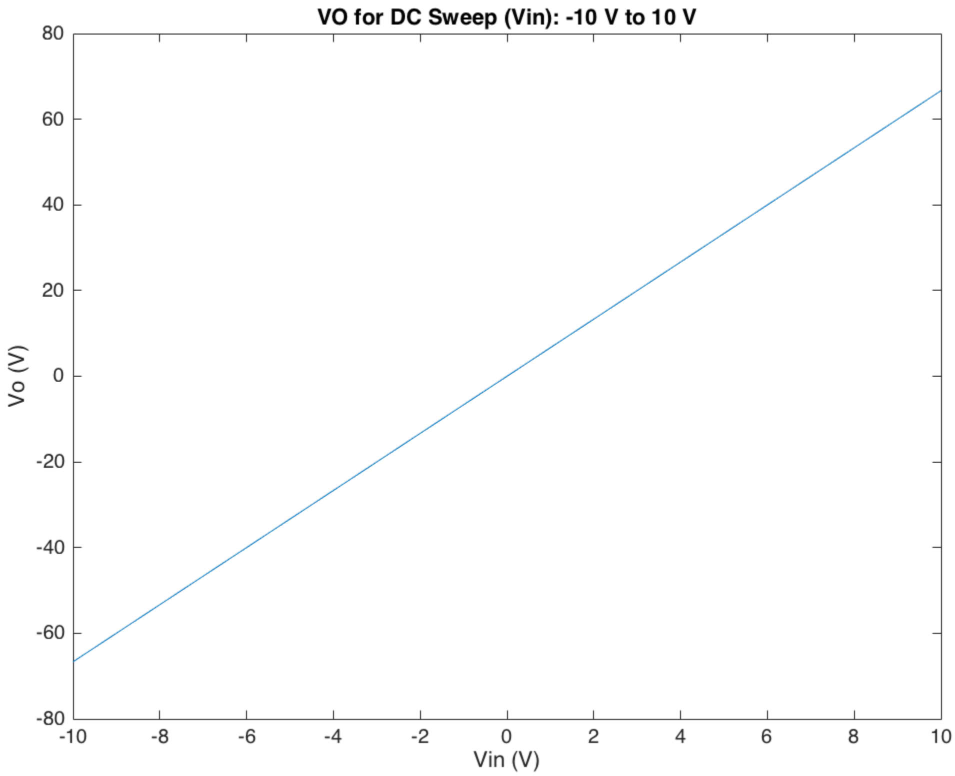
The G and C matrices were set as follows:

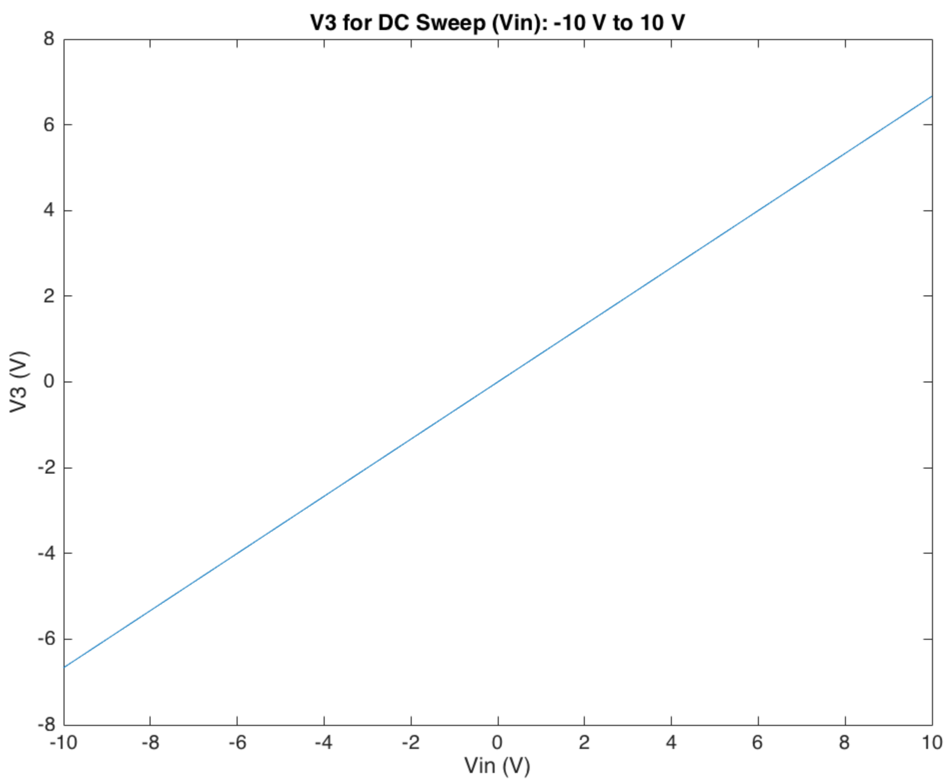
```
G
C
```

```
G =

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


C =

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

The input was swept as a DC input from -10V to 10V. Both the output voltage and the voltage, V3, were plotted over this DC sweep.

```
F = zeros(1, 6);
v = -10;

for i = 1:21
    vin(i) = v;
    F(1) = vin(i);

    Vm = G\F';

    vo(i) = Vm(6);
    v3(i) = Vm(4);
    v = v + 1;
end


figure(1)
plot(vin, vo);
title('VO for DC Sweep (Vin): -10 V to 10 V');
xlabel('Vin (V)')
ylabel('Vo (V)')

figure(2)
plot(vin, v3)
title('V3 for DC Sweep (Vin): -10 V to 10 V')
xlabel('Vin (V)')
ylabel('V3 (V)')
```

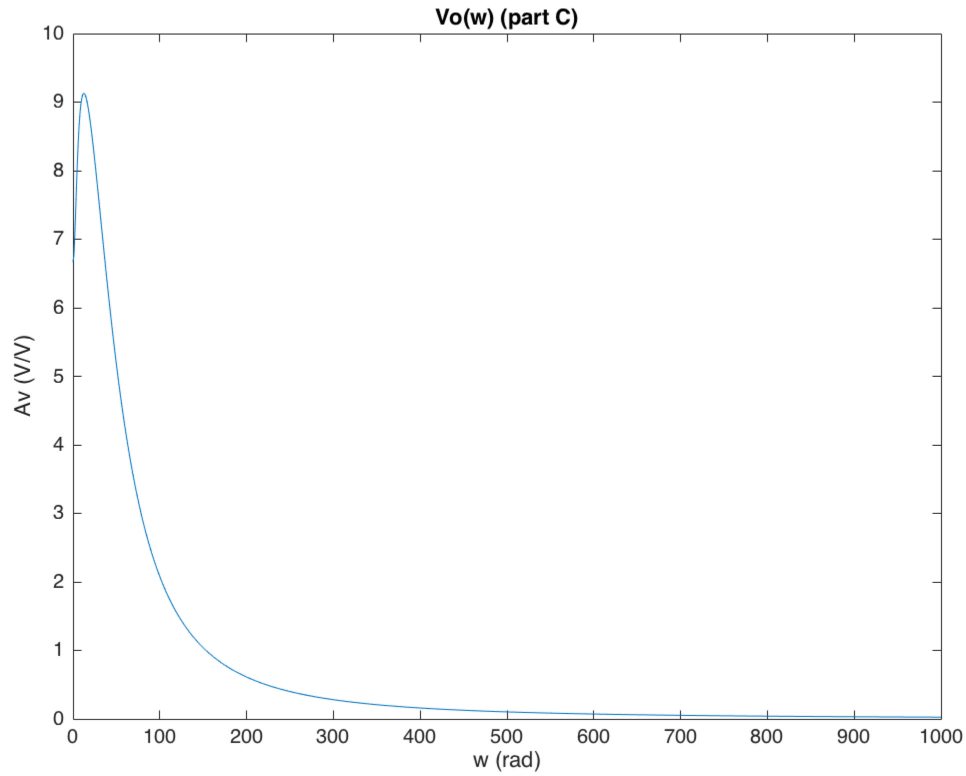

VO for DC Sweep (Vin): -10 V to 10 V
Vo (V)
Vin (V)
80
60
40
20
0
-20
-40
-60
-80
-10
-8
-6
-4
-2
0
2
4
6
8
10


V3 for DC Sweep (Vin): -10 V to 10 V
V3 (V)
Vin (V)
8
6
4
2
0
-2
-4
-6
-8
-10
-8
-6
-4
-2
0
2
4
6
8
10

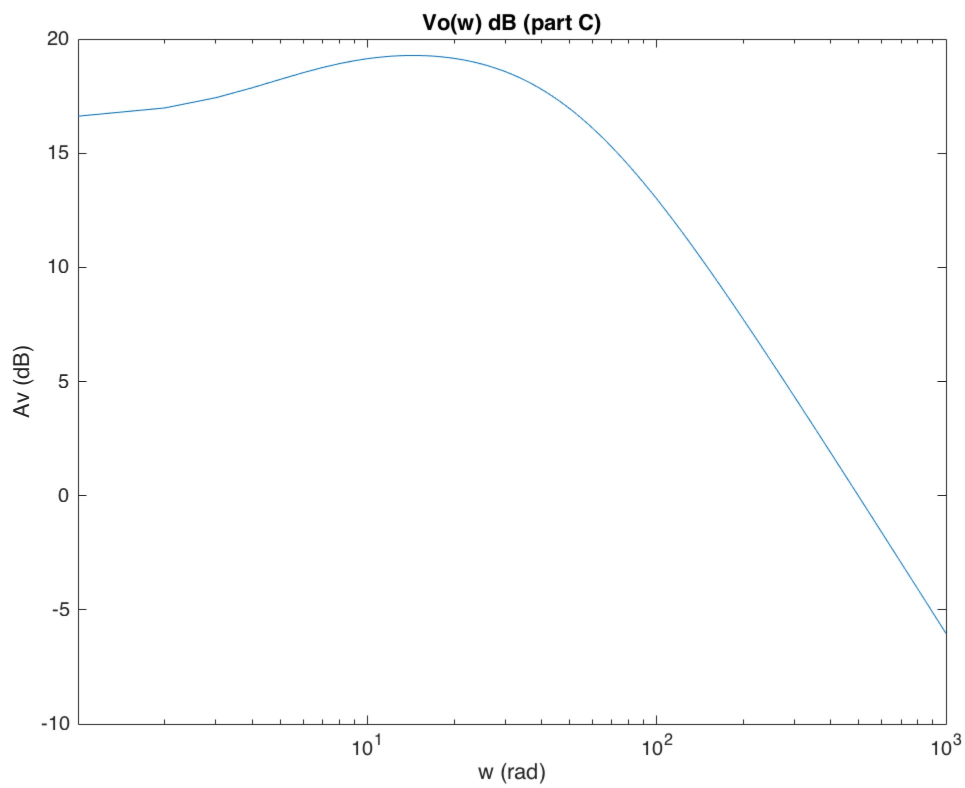
Next, Vo was analyzed for an AC case. Therefore, the output was plotted against the angular frequency, and the dB gain was plotted.

```
F(1) = 1;
vo2 = zeros(1, 1000);
freq = linspace(0, 1000, 1000); % note: in radians
Av = zeros(1, 1000);
Avlog = zeros(1, 1000);

for i = 1:1000
    Vm2 = (G+1i*freq(i)*C)\F';
    vo2(i) = Vm2(6);
    Av(i) = vo2(i)/F(1);
    Avlog(i) = 20*log10(Av(i));
end
figure(3)
plot(freq, Av)
title('Vo(w) (part C)')
xlabel('w (rad)')
ylabel('Av (V/V)')

figure(4)
semilogx(freq, Avlog)
xlim([0 1000])
title('Vo(w) dB (part C)')
xlabel('w (rad)')
ylabel('Av (dB)')
```

*Warning: Imaginary parts of complex X and/or Y arguments ignored*
*Warning: Imaginary parts of complex X and/or Y arguments ignored*



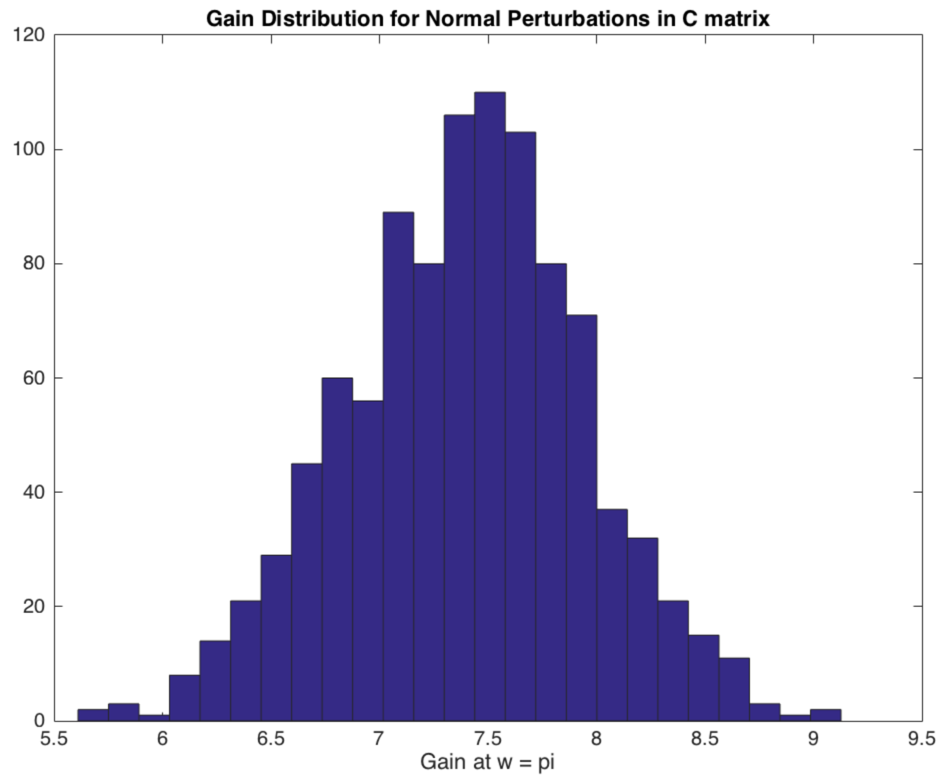
Vo(w) (part C)
Av (V/V)
w (rad)
10
9
8
7
6
5
4
3
2
1
0
0
100
200
300
400
500
600
700
800
900
1000

Vo(w) dB (part C)
Av (dB)
w (rad)
20
15
10
5
0
-5
-10
$10^1$
$10^2$
$10^3$

The AC case was plotted again where the gain was plotted as function of random perturbations on C using a normal distribution with std = .05 at ? = ?. A histogram was made to demonstrate the changes in the gain.

```
w = pi;
Av2 = zeros(15, 1);
Cper = zeros(15, 1);
vo3 = zeros(1, 15);

for i = 1:1000
    C(2, 1) = normrnd(-Cval, 0.05);
    C(2, 2) = normrnd(Cval, 0.05);
    C(3, 3) = normrnd(L, 0.05);
    Vm3 = (G+1i*w*C)\F';
    vo3(i) = Vm3(6);
    Av2(i) = vo3(i)/F(1);
end

figure(5)
hist(real(Av2), 25)
title('Gain Distribution for Normal Perturbations in C matrix')
xlabel('Gain at w = pi')
```

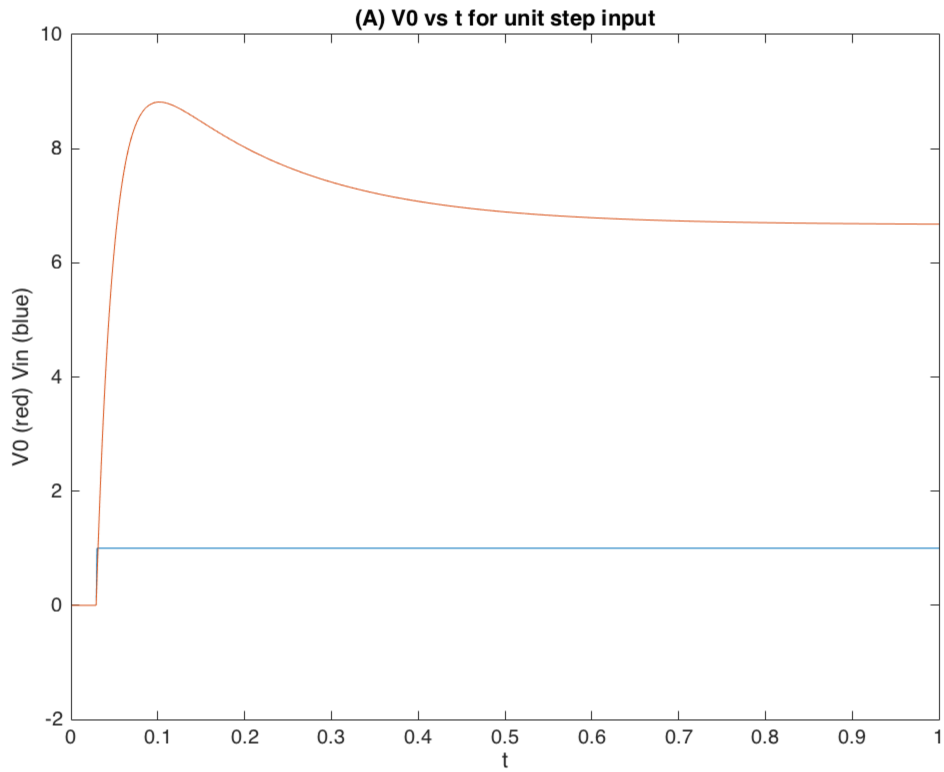



# Part 2: Transient Circuit Simulation

In this part, the circuit was analyzed in the time domain for various inputs for a time period of 1 second. The first input was a simple unit step input that goes high after 0.03 seconds.

```
vin = zeros(1, 1000);
vo = zeros(1, 1000);
v3 = zeros(1, 1000);

G(1, 1) = 1;                                        % 1
G(2, 1) = -G1; G(2, 2) = G1 + G2;                   % 2
G(3 ,2) = -1; G(3, 4) = 1;                          % iL
G(4, 3) = -1; G(4, 4) = G3;                         % 3
G(5, 5) = 1; G(5, 4) = -alpha*G3;                   % 4
G(6, 6) = G4 + G0; G(6, 5) = -G4;                   % 5

C = zeros(6);

C(2, 1) = -Cval; C(2, 2) = Cval;
C(3, 3) = L;

F = zeros(1, 6);

ii = 1;


V = zeros(6,1);
for t = 0.001:0.001:1
    Vold = V;
    if t < 0.03
        vin(ii) = 0;
        F(1) = 1;
    else
        vin(ii) = 1;
        F(1) = 0;
    end

    F(1) = vin(ii);
    A = (C / 0.001) + G;
    V = A \ ((C * Vold / 0.001) + F');

    vo(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.001 : 1;
figure(6);
plot(t, vin);
%title('(A) V1 vs t');
%ylabel('V1');
%xlabel('t');
hold on
plot(t, vo);
title('(A) V0 vs t for unit step input');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
ylim([-2 10])
```

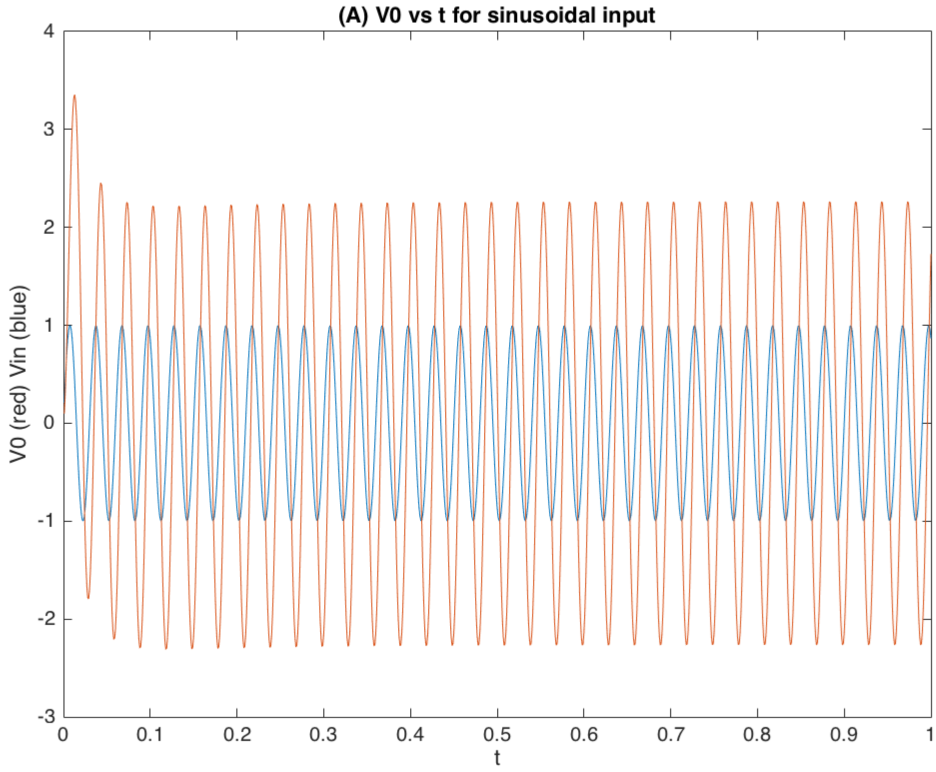
```
F = zeros(1, 6);

ii = 1;
```




The second input was a sinusoidal input at a frequency of 1/0.03 Hz.

```
V = zeros(6,1);

for t = 0.001:0.001:1
    Vold = V;
    vin2(ii) = sin(2*pi*(1/0.03) * t);

    F(1) = vin2(ii);
    A = (C / 0.001) + G;
    V = A \ ((C * Vold / 0.001) + F');

    vo2(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.001 : 1;
figure(7);
plot(t, vin2);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
```

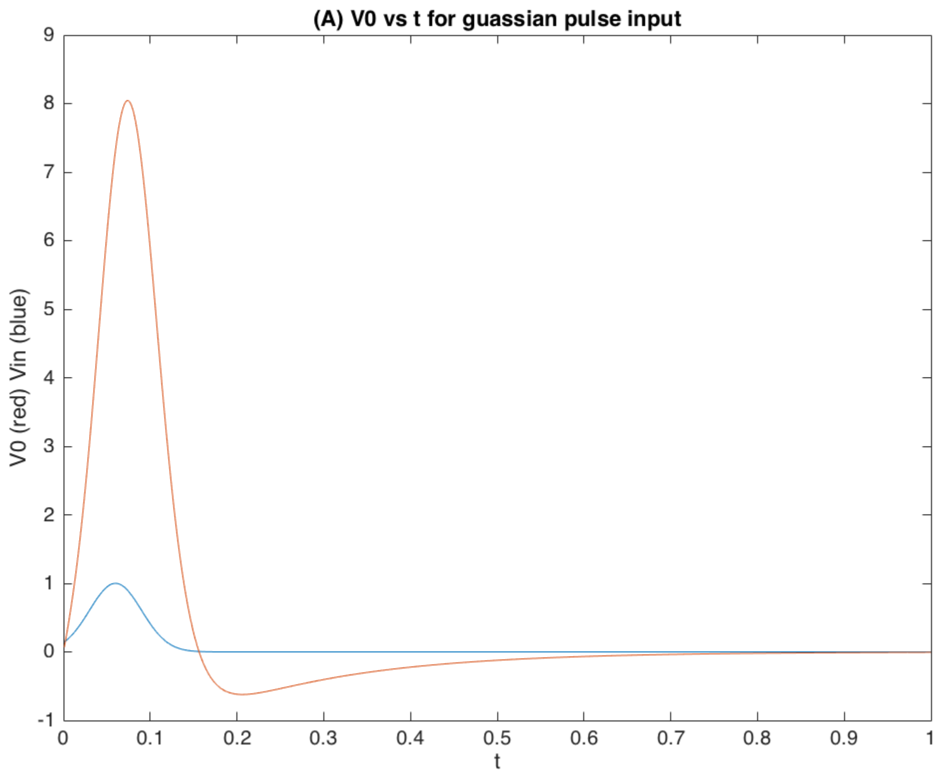
```
hold on
plot(t, vo2);
title('(A) V0 vs t for sinusoidal input');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
```




The third input was a guassian pulse with a magnitude of 1, std dev. of 0.03 seconds and a delay of 0.06 seconds.

```
A = zeros(6);
F = zeros(1, 6);
v = -10;
ii = 1;
dt = 1.0/1000;
Vold = zeros(6,1);
V = zeros(6,1);

for t = 0.001:0.001:1
    Vold = V;
    vin3(ii) = exp(-0.5 * power(((t - 0.06) / (0.03)), 2)); % guassian
 pulse

    F(1) = vin3(ii);
    A = (C / 0.001) + G;
    V = A \ ((C * Vold / 0.001) + F');

    vo3(ii) = V(6);
```

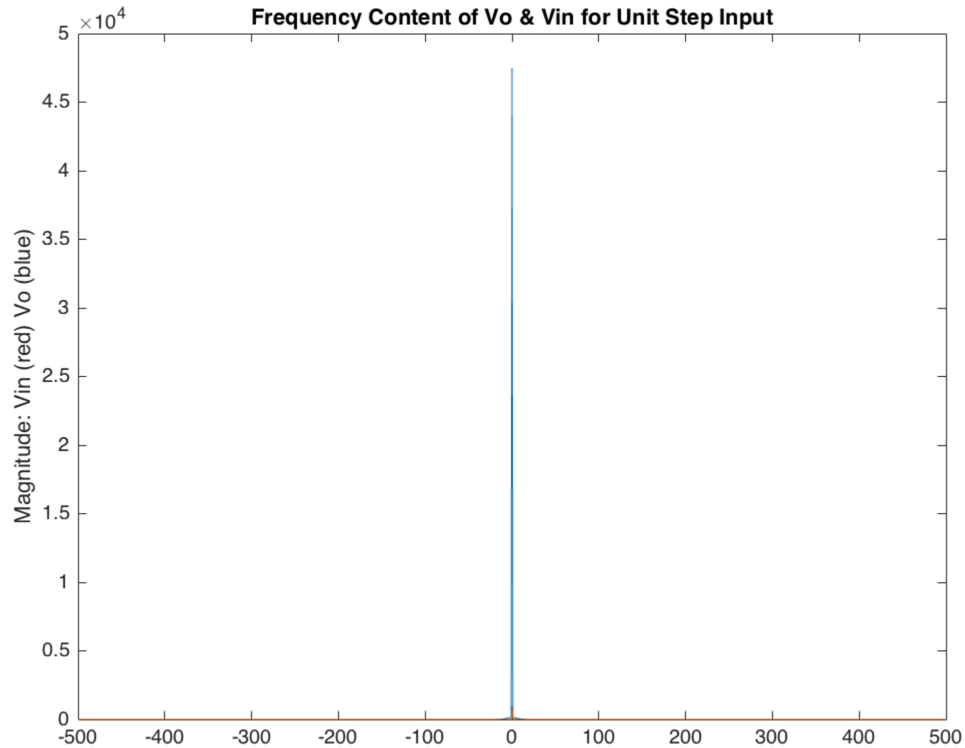
```
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.001 : 1;
figure(8);
plot(t, vin3);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
hold on
plot(t, vo3);
title('(A) V0 vs t for guassian pulse input');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
```

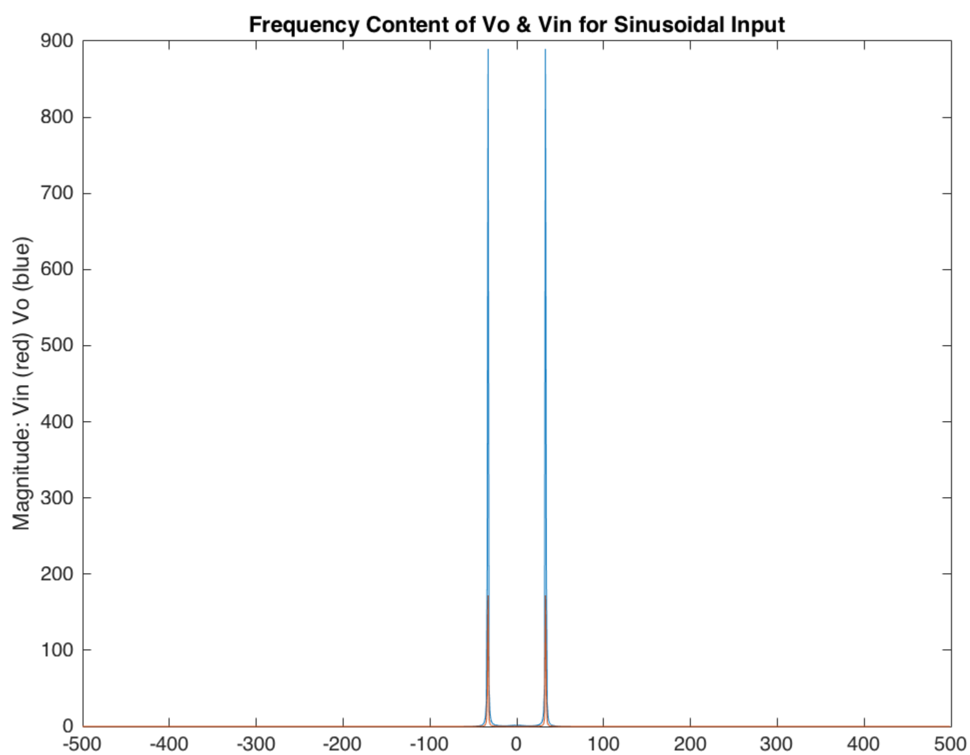


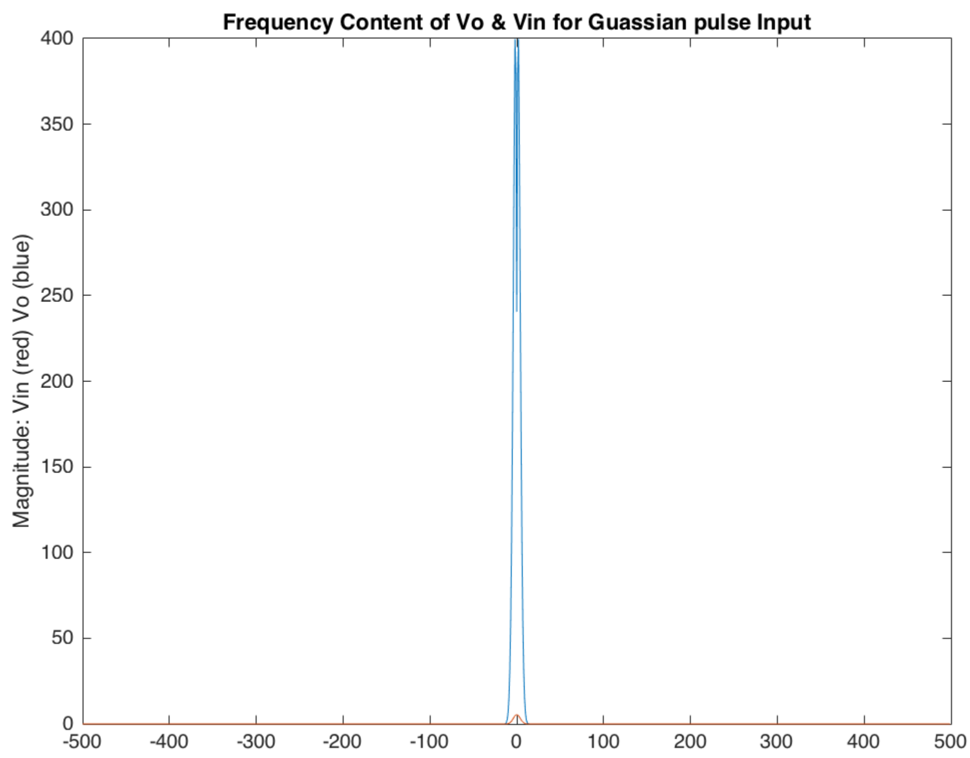

Next, the frequency content of each output and input that was analyzed in the previous part was plotted.

```
% Frequency content for each input type:
% Unit step:
fo = fft(vo);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
figure(9);
plot(fs,p);
title('Frequency Content of Vo for Unit Step Input');
hold on
```

```
fo = fft(vin);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
plot(fs,p);
title('Frequency Content of Vo & Vin for Unit Step Input');
ylabel('Magnitude: Vin (red) Vo (blue)')
hold off

% Sine:
fo = fft(vo2);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
figure(10);
plot(fs,p);
hold on
fo = fft(vin2);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
plot(fs,p);
title('Frequency Content of Vo & Vin for Sinusoidal Input');
ylabel('Magnitude: Vin (red) Vo (blue)')
hold off

% Guassian
fo = fft(vo3);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
figure(11);
plot(fs,p);
hold on
fo = fft(vin3);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
plot(fs,p);
title('Frequency Content of Vo & Vin for Guassian pulse Input');
ylabel('Magnitude: Vin (red) Vo (blue)')
hold off
```


Frequency Content of Vo & Vin for Unit Step Input
Magnitude: Vin (red) Vo (blue)
×10^4
5
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


Frequency Content of Vo & Vin for Sinusoidal Input
Magnitude: Vin (red) Vo (blue)
900
800
700
600
500
400
300
200
100
0
-500
-400
-300
-200
-100
0
100
200
300
400
500

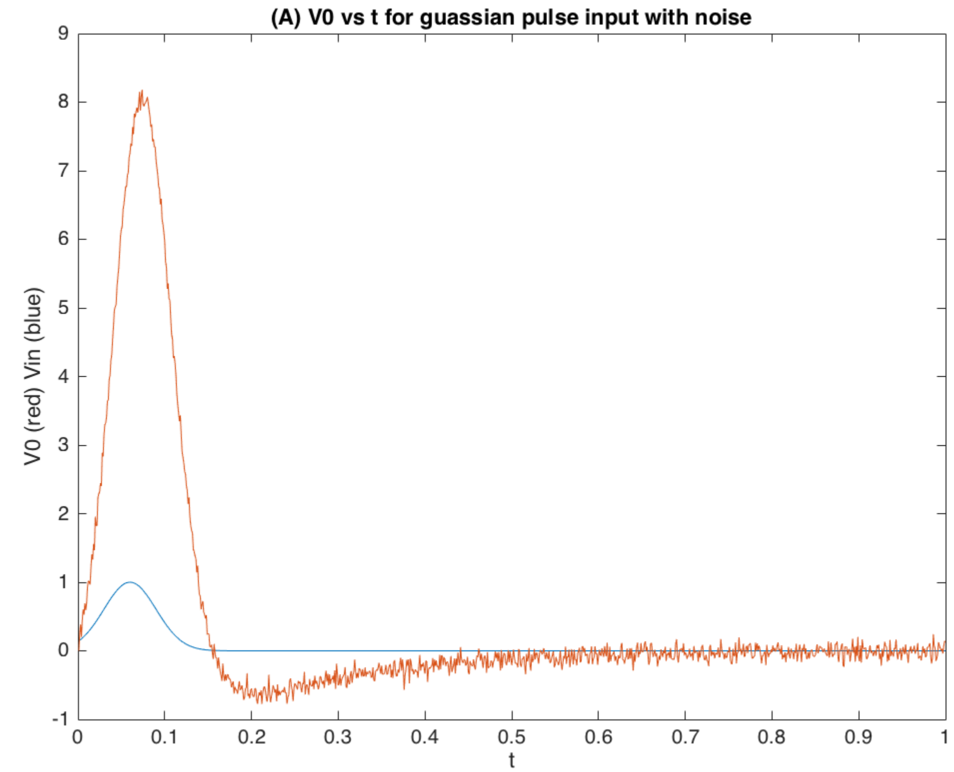


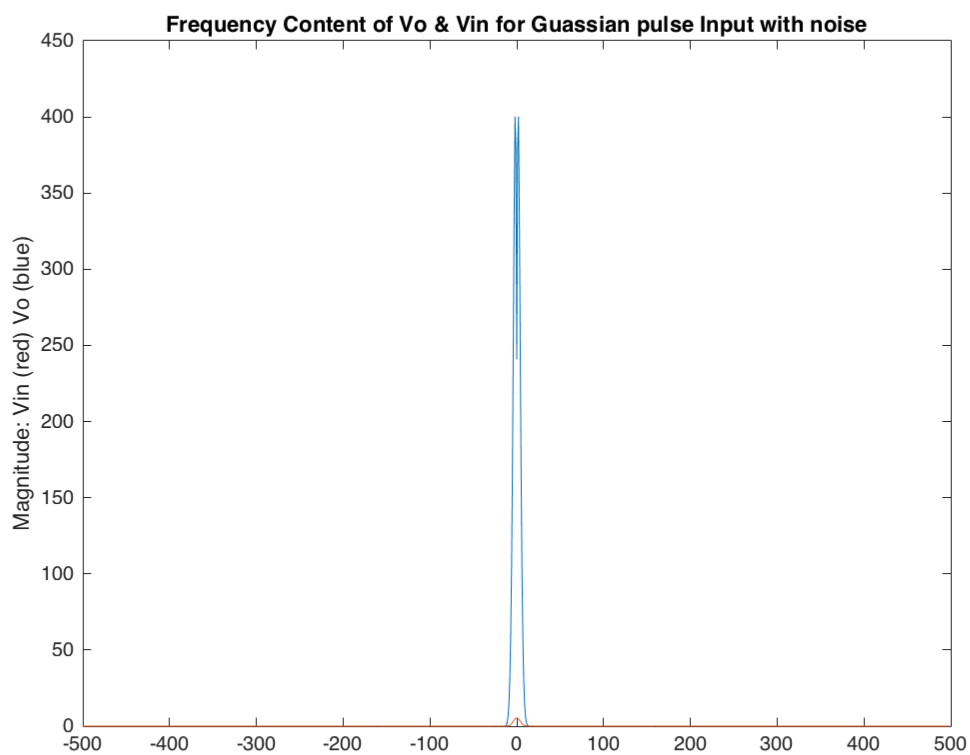

# Part 3: Circuit With Noise

In this section, a current source and capacitor was added in parallel with R3 to simulate noise. Therefore, a new G and C matrix was formulated.

```
Cn = 0.00001;

G(1, 1) = 1;                                    % 1
G(2, 1) = -G1; G(2, 2) = G1 + G2;               % 2
G(3 ,2) = -1; G(3, 4) = 1;                      % iL
G(4, 3) = -1; G(4, 4) = G3; G(4, 7) = 1;        % 3
G(5, 5) = 1; G(5, 4) = -alpha*G3;               % 4
G(6, 6) = G4 + G0; G(6, 5) = -G4;               % 5
G(7, 7) = 1;                                    % In

C = zeros(7);

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

A = zeros(7);
F = zeros(1, 7);
ii = 1;
dt = 1.0/1000;
Vold = zeros(7,1);
```

```
V = zeros(7,1);

for t = 0.001:0.001:1
    Vold = V;
    vin4(ii) = exp(-0.5 * power(((t - 0.06) / (0.03)), 2)); % guassian
 pulse
    In = randn * 0.001;
    F(7) = In;
    F(1) = vin4(ii);
    A = (C / 0.001) + G;
    V = A \ ((C * Vold / 0.001) + F');

    vo4(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.001 : 1;
figure(12);
plot(t, vin4);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
hold on
plot(t, vo4);
title('(A) V0 vs t for guassian pulse input with noise');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
hold off

% Guassian with noise
fo = fft(vo4);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
figure(13);
plot(fs,p);
hold on
fo = fft(vin4);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
plot(fs,p);
title('Frequency Content of Vo & Vin for Guassian pulse Input with
 noise');
ylabel('Magnitude: Vin (red) Vo (blue)')
hold off
```


(A) V0 vs t for guassian pulse input with noise
V0 (red) Vin (blue)
t
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

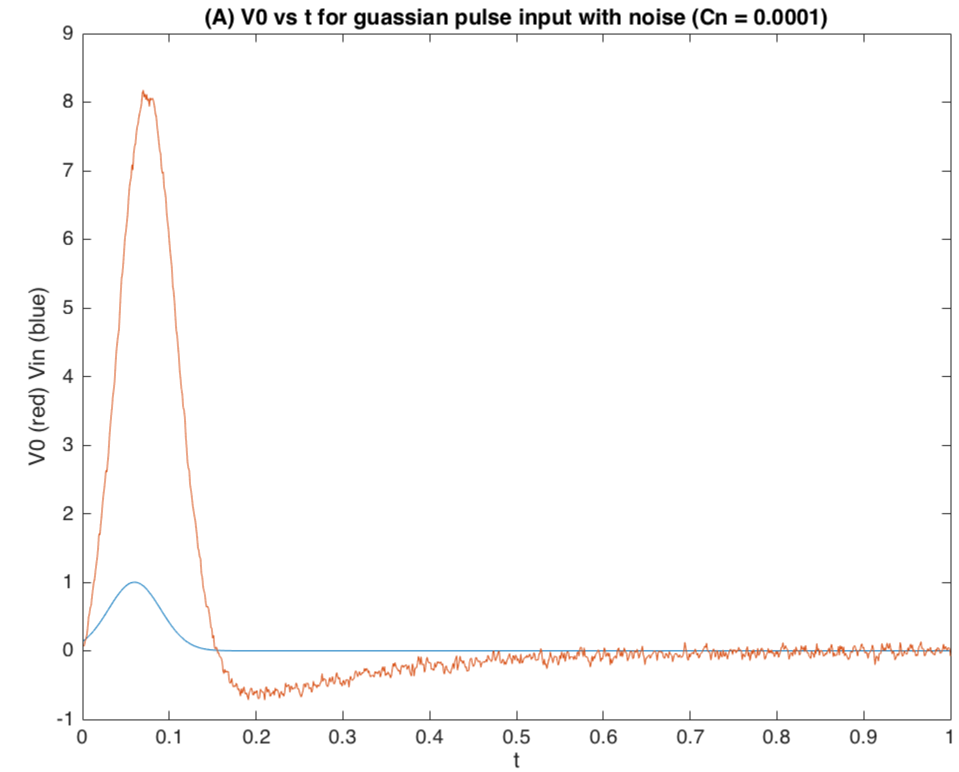

Frequency Content of Vo & Vin for Guassian pulse Input with noise
Magnitude: Vin (red) Vo (blue)
450
400
350
300
250
200
150
100
50
0
-500
-400
-300
-200
-100
0
100
200
300
400
500

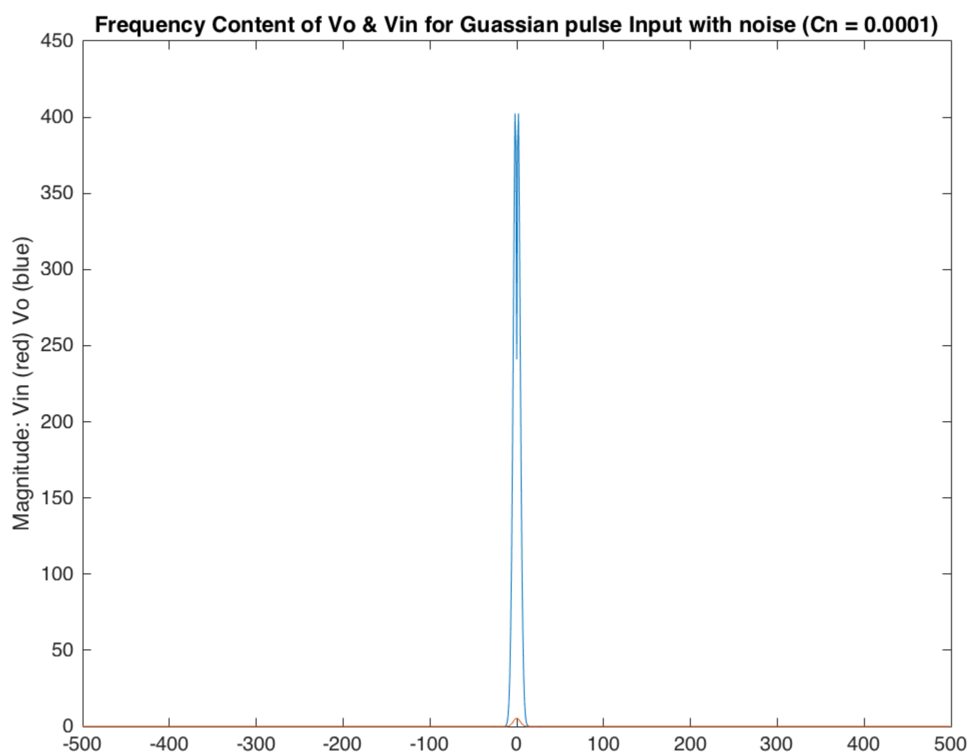
Next, the value of the newly added capacitor was varied to analyze its effect on the bandwidth.

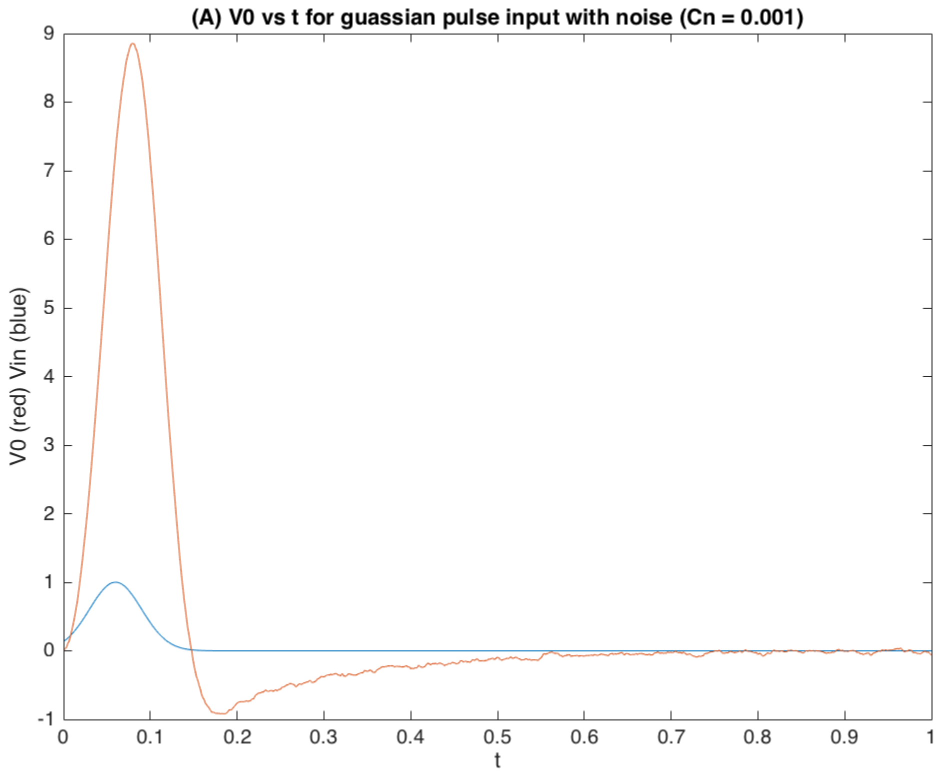
```
C(4, 4) = 0.0001;

A = zeros(7);
F = zeros(1, 7);
ii = 1;
dt = 1.0/1000;
Vold = zeros(7,1);
V = zeros(7,1);

for t = 0.001:0.001:1
    Vold = V;
    vin5(ii) = exp(-0.5 * power(((t - 0.06) / (0.03)), 2)); % guassian
 pulse
    In = randn * 0.001;
    F(7) = In;
    F(1) = vin5(ii);
    A = (C / 0.001) + G;
    V = A \ ((C * Vold / 0.001) + F');

    vo5(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.001 : 1;
figure(14);
plot(t, vin5);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
hold on
plot(t, vo5);
title('(A) V0 vs t for guassian pulse input with noise (Cn =
 0.0001)');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
hold off

% Guassian with noise
fo = fft(vo5);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
figure(15);
plot(fs,p);
hold on
fo = fft(vin5);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
plot(fs,p);
title('Frequency Content of Vo & Vin for Guassian pulse Input with
 noise (Cn = 0.0001)');
ylabel('Magnitude: Vin (red) Vo (blue)')
```

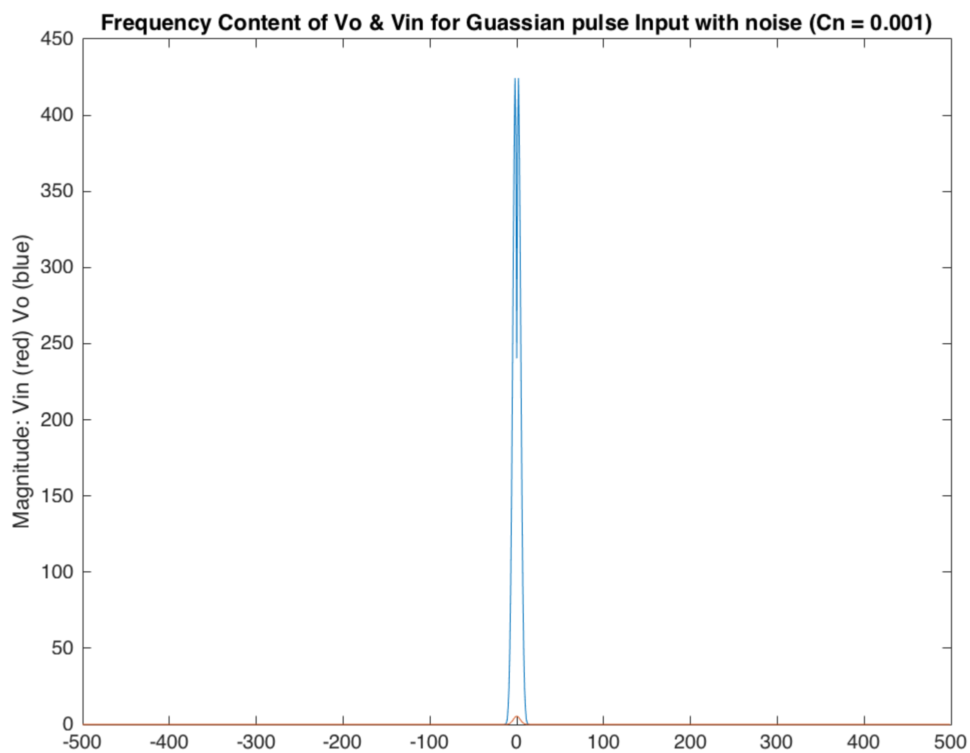
```
hold off

C(4, 4) = 0.001;

A = zeros(7);
F = zeros(1, 7);
ii = 1;
dt = 1.0/1000;
Vold = zeros(7,1);
V = zeros(7,1);

for t = 0.001:0.001:1
    Vold = V;
    vin6(ii) = exp(-0.5 * power(((t - 0.06) / (0.03)), 2)); % guassian
 pulse
    In = randn * 0.001;
    F(7) = In;
    F(1) = vin6(ii);
    A = (C / 0.001) + G;
    V = A \ ((C * Vold / 0.001) + F');

    vo6(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.001 : 1;
figure(16);
plot(t, vin4);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
hold on
plot(t, vo6);
title('(A) V0 vs t for guassian pulse input with noise (Cn = 0.001)');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
hold off

% Guassian with noise
fo = fft(vo6);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
figure(17);
plot(fs,p);
hold on
fo = fft(vin6);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
plot(fs,p);
title('Frequency Content of Vo & Vin for Guassian pulse Input with
 noise (Cn = 0.001)');
ylabel('Magnitude: Vin (red) Vo (blue)')
hold off
```

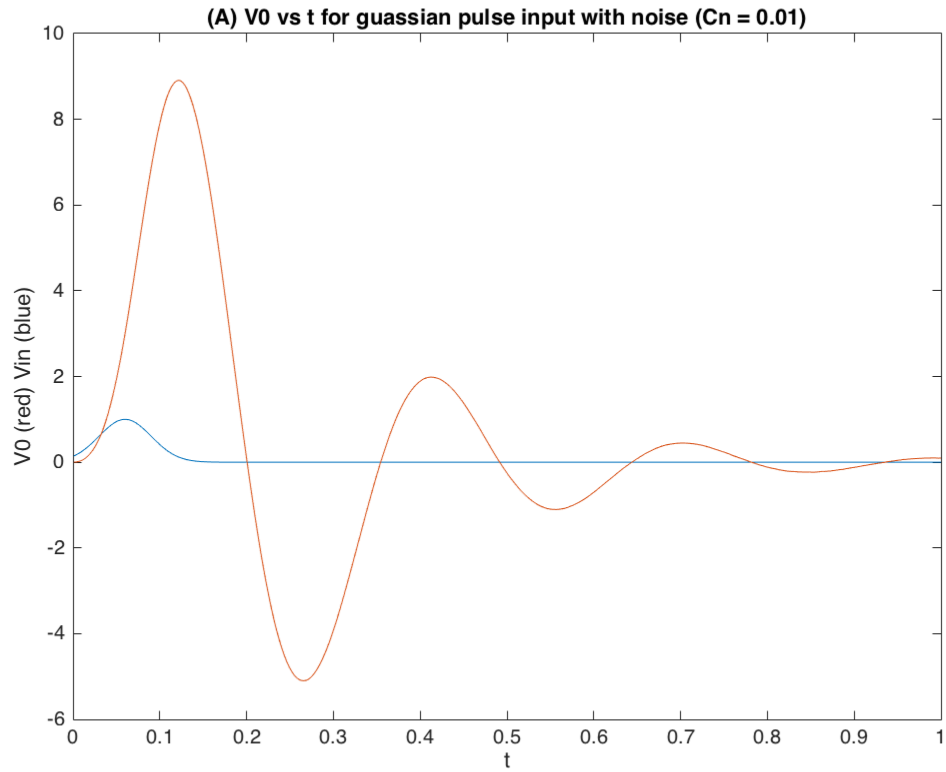
```
C(4, 4) = 0.01;

A = zeros(7);
F = zeros(1, 7);
ii = 1;
dt = 1.0/1000;
Vold = zeros(7,1);
V = zeros(7,1);

for t = 0.001:0.001:1
    Vold = V;
    vin7(ii) = exp(-0.5 * power(((t - 0.06) / (0.03)), 2)); % guassian
 pulse
    In = randn * 0.001;
    F(7) = In;
    F(1) = vin7(ii);
    A = (C / 0.001) + G;
    V = A \ ((C * Vold / 0.001) + F');

    vo7(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.001 : 1;
figure(18);
plot(t, vin7);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
hold on
plot(t, vo7);
title('(A) V0 vs t for guassian pulse input with noise (Cn = 0.01)');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
hold off

% Guassian with noise
fo = fft(vo7);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
figure(19);
plot(fs,p);
hold on
fo = fft(vin7);
fs = (-1000/2:1000/2-1);
p = abs(fftshift(fo)) .^ 2/1000;
plot(fs,p);
title('Frequency Content of Vo & Vin for Guassian pulse Input with
 noise (Cn = 0.01)');
ylabel('Magnitude: Vin (red) Vo (blue)')
hold off
```


(A) V0 vs t for guassian pulse input with noise (Cn = 0.0001)
V0 (red) Vin (blue)
t
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

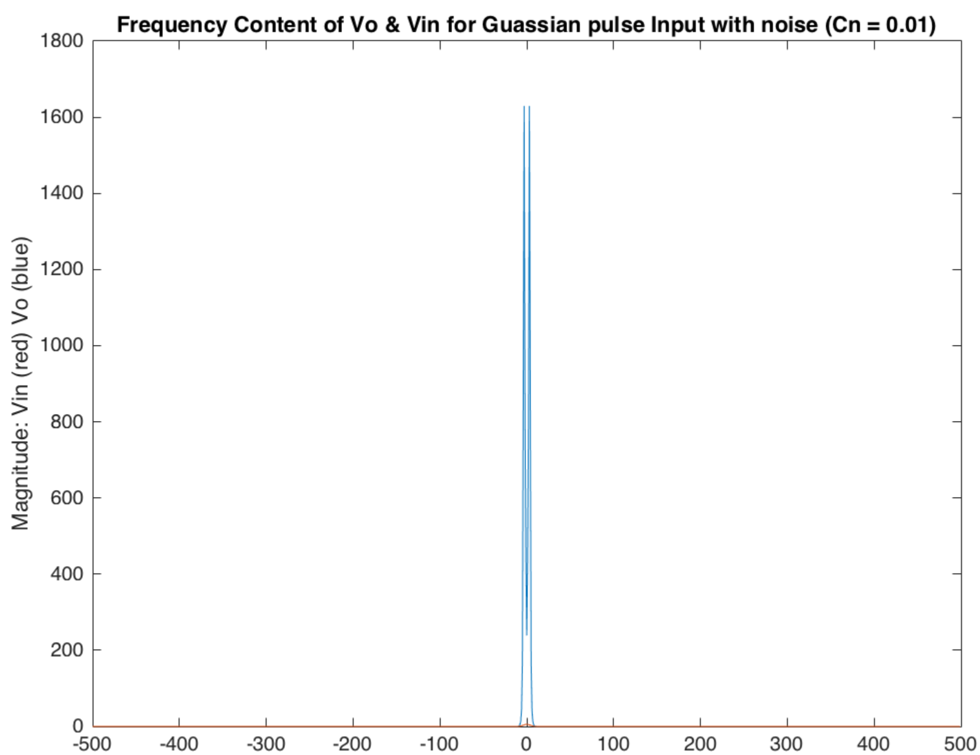

Frequency Content of Vo & Vin for Guassian pulse Input with noise (Cn = 0.0001)
Magnitude: Vin (red) Vo (blue)
450
400
350
300
250
200
150
100
50
0
-500
-400
-300
-200
-100
0
100
200
300
400
500


(A) V0 vs t for guassian pulse input with noise (Cn = 0.001)
V0 (red) Vin (blue)
t
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


Frequency Content of Vo & Vin for Guassian pulse Input with noise (Cn = 0.001)
Magnitude: Vin (red) Vo (blue)
450
400
350
300
250
200
150
100
50
0
-500
-400
-300
-200
-100
0
100
200
300
400
500



(A) V0 vs t for guassian pulse input with noise (Cn = 0.01)
V0 (red) Vin (blue)
t
10
8
6
4
2
0
-2
-4
-6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

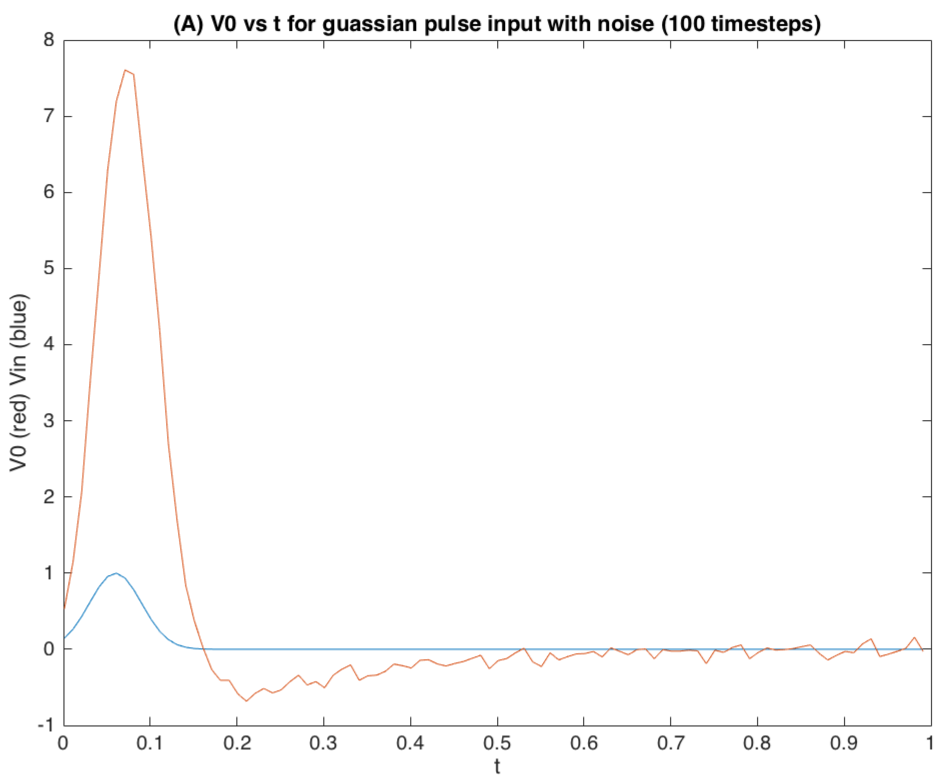
Frequency Content of Vo & Vin for Guassian pulse Input with noise (Cn = 0.01)
Magnitude: Vin (red) Vo (blue)
1800
1600
1400
1200
1000
800
600
400
200
0
-500
-400
-300
-200
-100
0
100
200
300
400
500

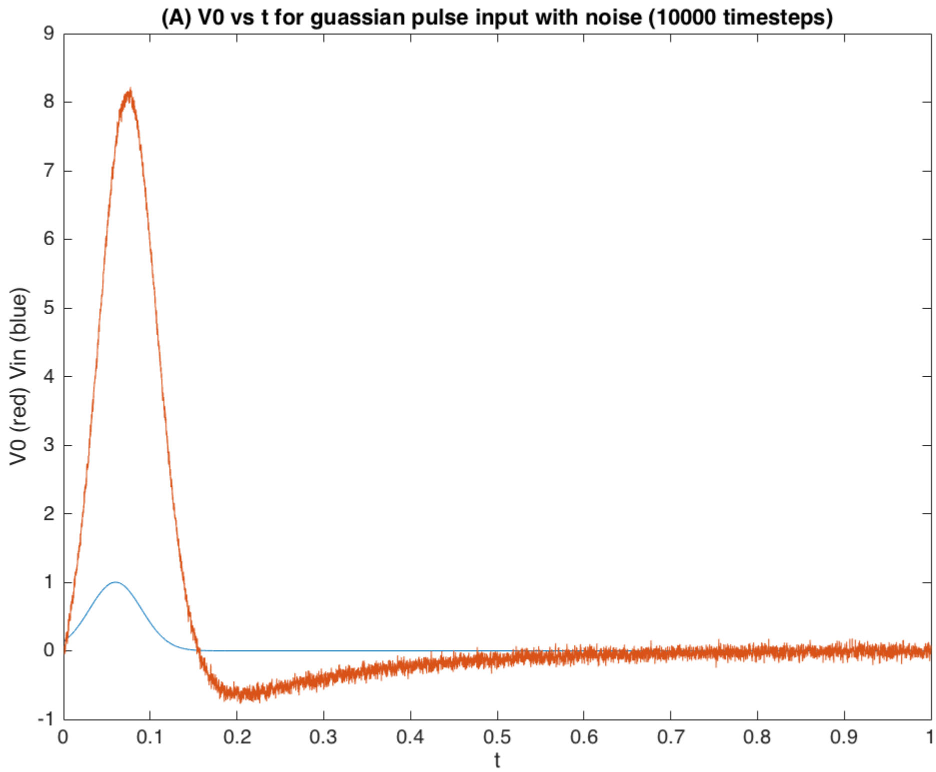
The capacitance begins to more clearly affect the bandwidth when the value approached 0.01 F. For this value, the bandwidth decreased, evident in comparing the frequency spectrums for this case (see figure 19) and the previous cases (see figures 17 and 15). The simulation results suggest that the bandwidth decreases as the capacitance increases.

Next, the number of timesteps were varied.

```
C(4, 4) = 0.00001;

A = zeros(7);
F = zeros(1, 7);
ii = 1;
dt = 1.0/100;
Vold = zeros(7,1);
V = zeros(7,1);

for t = 0.001:0.01:1
    Vold = V;
    vin8(ii) = exp(-0.5 * power(((t - 0.06) / (0.03)), 2)); % guassian
 pulse
    In = randn * 0.001;
    F(7) = In;
    F(1) = vin8(ii);
    A = (C / 0.01) + G;
    V = A \ ((C * Vold / 0.01) + F');

    vo8(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.01 : 1;
figure(20);
plot(t, vin8);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
hold on
plot(t, vo8);
title('(A) V0 vs t for guassian pulse input with noise (100
 timesteps)');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
hold off

A = zeros(7);
F = zeros(1, 7);
ii = 1;
dt = 1.0/10000;
Vold = zeros(7,1);
V = zeros(7,1);

for t = 0.001:0.0001:1
    Vold = V;
```

```
    vin9(ii) = exp(-0.5 * power(((t - 0.06) / (0.03)), 2)); % guassian
 pulse
    In = randn * 0.001;
    F(7) = In;
    F(1) = vin9(ii);
    A = (C / 0.0001) + G;
    V = A \ ((C * Vold / 0.0001) + F');

    vo9(ii) = V(6);
    %v3(ii) = V(4);
    ii = ii + 1;
end

t = 0.001 : 0.0001 : 1;
figure(21);
plot(t, vin9);
%title('(A) V1 vs t');
%ylabel('V1');
xlabel('t');
hold on
plot(t, vo9);
title('(A) V0 vs t for guassian pulse input with noise (10000
 timesteps)');
ylabel('V0 (red) Vin (blue)');
xlabel('t');
hold off
```






# Non-linearity

In order to solve for a non-linear transconductance equation for the voltage source on the output stage, a new B matrix would need to be implemented. In addition, a jacobian matrix may be implemented in the program for the derivatives, which may present nested loops. Also, a H matrix would be required in order to solve for the V matrix.

*Published with MATLAB® R2015b*