Assignment 4 Part 3 - Andrew Paul 100996250

Table of Contents

Updated G matrix 1
Updated C matrix 1
Assignment 4 Part 4 - Andrew Paul 15

In the third section of this assignment a current source and capcitor were added to create noise in the system. The G and C matricies were redefined in order to accomidate these new components.

Updated G matrix

	1	2	3	4	5	6	7	8	9
1	-1	1	1	0	0	0	0	0	0
2	1	0	-1.5000	0	0	0	-1	0	0
3	0	0	0	-0.1000	0	0	1	0	-1
4	0	0	1	-1	0	0	0	0	0
5	0	0	0	-1000	0	10	0	1	0
6	0	0	0	1000	0	-10.0010	0	0	0
7	1	0	0	0	0	0	0	0	0
8	0	0	0	-10	1	0	0	0	0
9	0	0	0	0	0	0	0	0	1

Updated C matrix

	1	2	3	4	5	6	7	8	9
1	-0.2500	0	0.2500	0	0	0	0	0	0
2	0.2500	0	-0.2500	0	0	0	0	0	0
3	0	0	0	-1.0000e-05	0	0	0	0	0
4	0	0	0	0	0	0	-0.2000	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

```
close all;
clear;

%Initialize variables and matricies

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

```
F = zeros(9,1);

R1 =1;
R2 = 2;
R3 = 10;
R4 = 0.1;
R0 = 1000;
cap = 0.25;
L = 0.2;
alpha = 100;
Cn = 0.00001;

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



% Redefined G matrix rows [V1 Iin V2 V3 V4 V5 IL I4 Iin]
% IL and I3 are no longer the same value with the added components
% Updated C and G Matrix

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;

% Create C matrix
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(1,9);

time_step = 0.001;

Vsolp = zeros(9,1);
A = C/(time_step) + G;

Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);
Vout = zeros(1000,1);
Vin = zeros(1000,1);

i = 1;
time = zeros(1000,1);

for t=0:time_step:1

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

    time(i) = t;
    F(1,7) = Vt(t);
    F(1,9) = In;
    Vsol = inv(A)*(C*Vsolp/time_step + F');
    Vout(i) = Vsol(6);
    Vin(i) = Vsol(1);
    Vsolp = Vsol;
    i = i+1;

end

figure(1)
subplot(2,1,2)
plot(time,Vout)
title('Guass Function - Vout vs. Time w/ Noise')
grid on

subplot(2,1,1)
plot(time,Vin)
title('Guass Function - Vin vs. Time w/ Noise')
grid on

% Vout Frequency plots

freq = 1000;
freqOut = fft(Vout);
x = length(Vout);
y = fftshift(freqOut);
freqShift = (-x/2:x/2-1)*(freq/x);
shift = abs(y).^2/x;

figure(2)
semilogy(freqShift,shift)
title('Gauss frquecny spectrum - Vout w/ Noise')
```

```
% Vin freq plots

freqIn = fft(Vin);
x = length(Vin);
t = (0:x-1)*(freq/x);
power = abs(freqIn).^2/x;
y=fftshift(freqIn);
freqShift = (-x/2:x/2-1)*(freq/x);
shift = abs(y).^2/x;

figure(3)
semilogy(freqShift,shift)
title('Gauss frquecny spectrum - Vin w/ Noise')



%Change Cap value 1
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;


A = C/(time_step) + G;
Vsolp = zeros(9,1);
Vt=@(t) exp(-(1/2)*((t-0.06)/(0.03))^2);
Vout=zeros(500,1);
Vin=zeros(500,1);
i = 1;
time=zeros(500,1);

for t=0:time_step:0.5
    In = randn(1)*0.001;
    time(i)=t;
    F(1,7) = Vt(t);
    F(1,9) = In;
    Vsol = inv(A)*(C*Vsolp/time_step + F');
    Vout(i) = Vsol(6);
    Vin(i) = Vsol(1);
    Vsolp = Vsol;
    i = i+1;
end

figure(4)
subplot(2,1,2)
plot(time,Vout)
title('Guass Function - Vout vs. Time w/ Noise - Cn = 0.1')
grid on
```

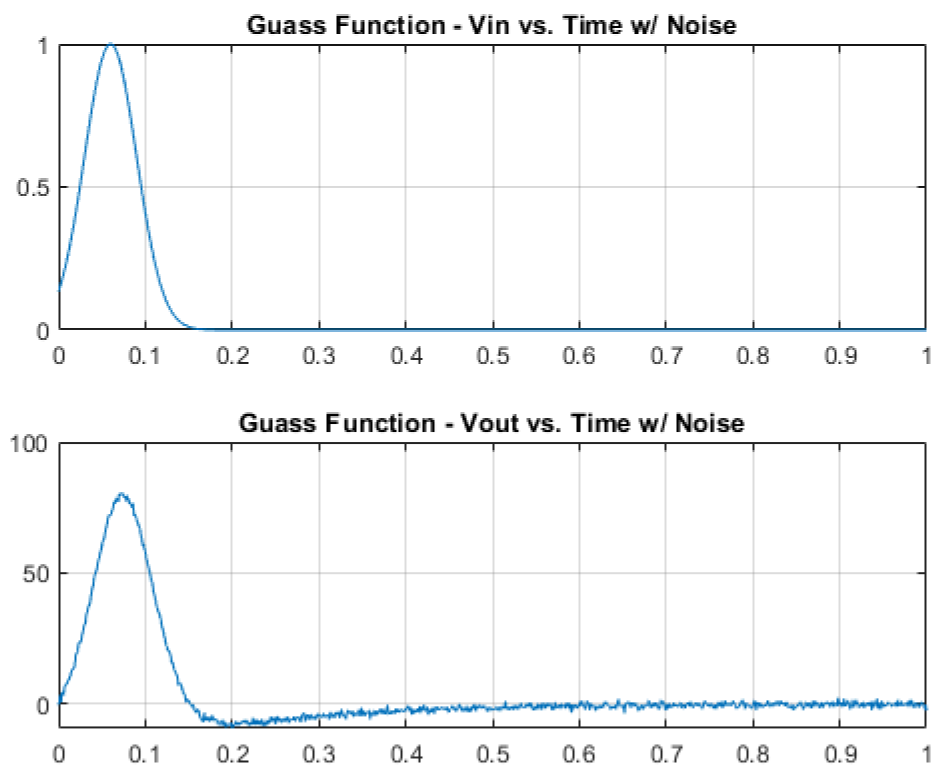
```
subplot(2,1,1)
plot(time,Vin)
title('Guass Function - Vin vs. Time w/ Noise - Cn = 0.1')
grid on

freq = 1000;
freqOut = fft(Vout);
x = length(Vout);
y = fftshift(freqOut);
freqShift = (-x/2:x/2-1)*(freq/x);
shift = abs(y).^2/x;

figure(8)
semilogy(freqShift,shift)
title('Guass Function - Freq domain w/ Noise - Cn = 0.1')


% %Change Cap value 2
Cn = 0.0000001;
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;


A = C/(time_step) + G;
Vsolp = zeros(9,1);
Vt = @(t) exp(-(1/2)*((t-0.06)/(0.03))^2);
Vout = zeros(500,1);
Vin = zeros(500,1);
i = 1;
time = zeros(500,1);

for t=0:time_step:0.5
    In = randn(1)*0.001;
    time(i)=t;
    F(1,7) = Vt(t);
    F(1,9) = In;
    Vsol = inv(A)*(C*Vsolp/time_step + F');
    Vout(i) = Vsol(6);
    Vin(i) = Vsol(1);
    Vsolp = Vsol;
    i = i+1;
end

figure(5)
subplot(2,1,2)
plot(time,Vout)
title('Guass Function - Vout vs. Time w/ Noise - Cn = 0.0000001')
grid on
```

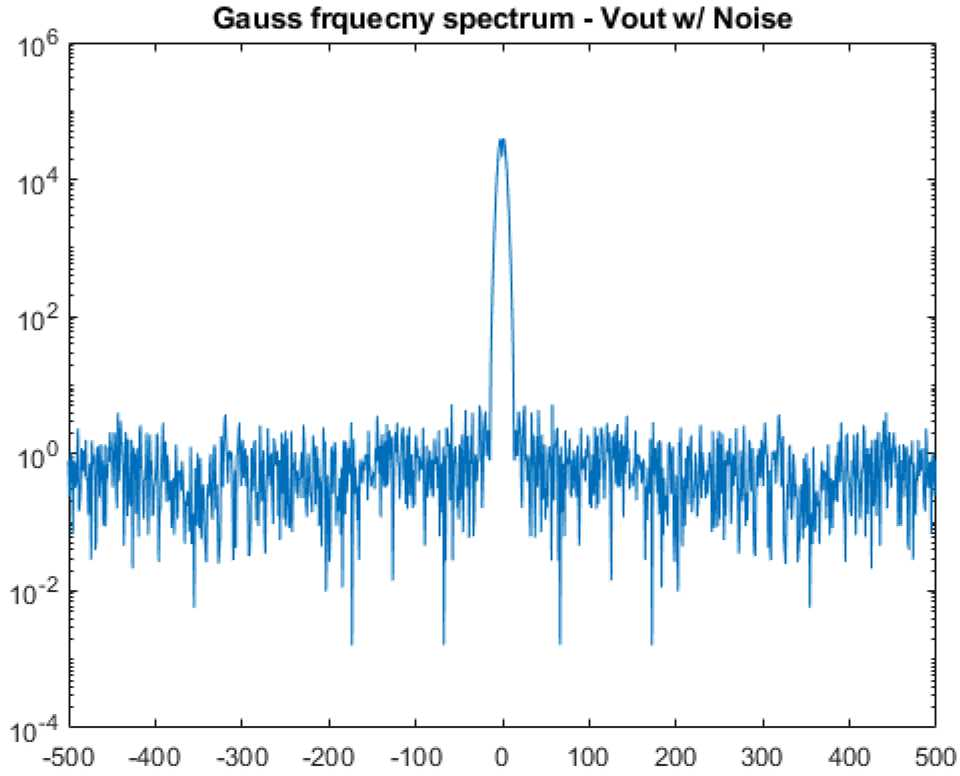
```
subplot(2,1,1)
plot(time,Vin)
title('Guass Function - Vin vs. Time w/ Noise - Cn = 0.0000001')
grid on

freq = 1000;
freqOut = fft(Vout);
x = length(Vout);
y = fftshift(freqOut);
freqShift = (-x/2:x/2-1)*(freq/x);
shift = abs(y).^2/x;

figure(11)
semilogy(freqShift,shift)
title('Guass Function - Freq domain w/ Noise - Cn = 0.0000001')

%Change Cap value 3
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;


A = C/(time_step) + G;
Vsolp = zeros(9,1);
Vt=@(t) exp(-(1/2)*((t-0.06)/(0.03))^2);
Vout=zeros(500,1);
Vin=zeros(500,1);
i = 1;
time=zeros(500,1);

for t=0:time_step:0.5
    In = randn(1)*0.001;
    time(i)=t;
    F(1,7) = Vt(t);
    F(1,9) = In;
    Vsol = inv(A)*(C*Vsolp/time_step + F');
    Vout(i) = Vsol(6);
    Vin(i) = Vsol(1);
    Vsolp = Vsol;
    i = i+1;
end

figure(9)
subplot(2,1,2)
plot(time,Vout)
title('Guass Function - Vout vs. Time w/ Noise - Cn = 0.001')
grid on
```

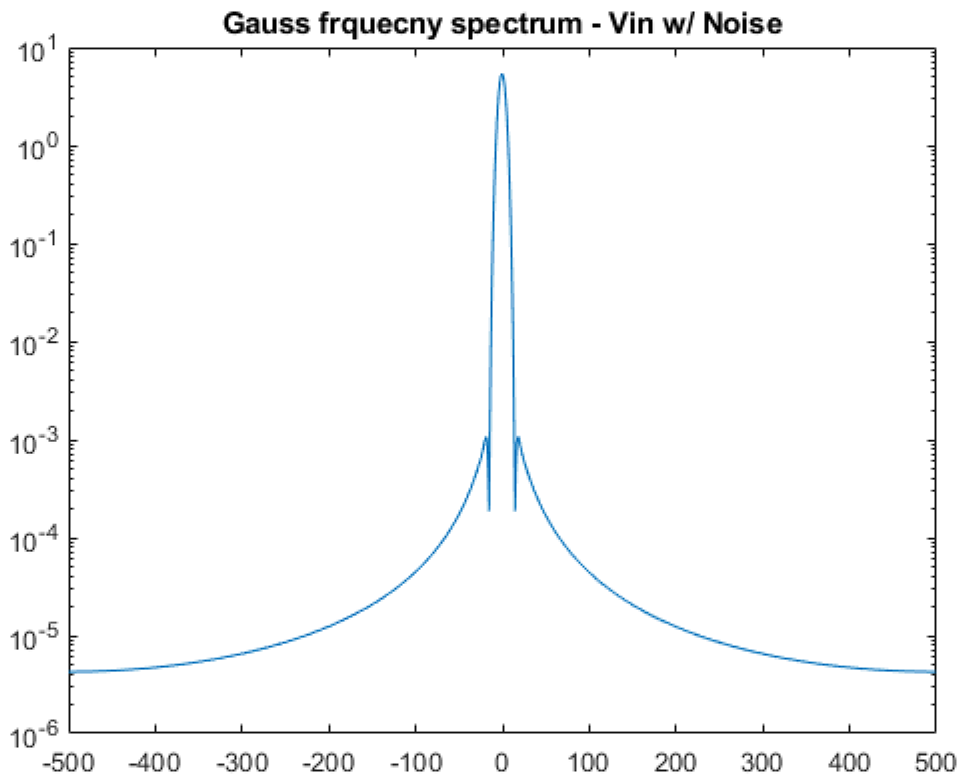
```
subplot(2,1,1)
plot(time,Vin)
title('Guass Function - Vin vs. Time w/ Noise - Cn = 0.001')
grid on

freq = 1000;
freqOut = fft(Vout);
x = length(Vout);
y = fftshift(freqOut);
freqShift = (-x/2:x/2-1)*(freq/x);
shift = abs(y).^2/x;

figure(10)
semilogy(freqShift,shift)
title('Guass Function - Freq domain w/ Noise - Cn = 0.001')

freq = 1000;



% Vary the Time Step

% Time Step = 0.0001

time_step=0.0001;

Vsolp= zeros(9,1);

Cn = 0.0000001;
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;

A = C/(time_step) + G;

Vt=@(t) exp(-(1/2)*((t-0.06)/(0.03))^2);
Vout = zeros(10000,1);
Vin = zeros(10000,1);

i = 1;

time = zeros(10000,1);
for t=0:time_step:1

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

    time(i) = t;
```

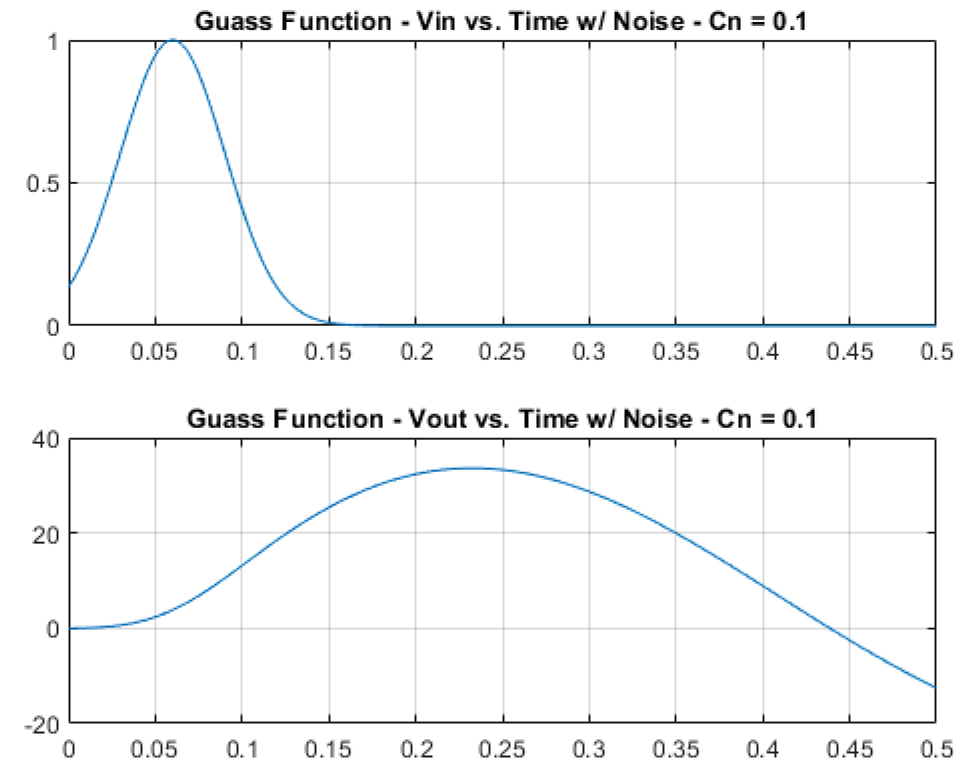
```
    F(1,7) = Vt(t);
    F(1,9) = In;
    Vsol = inv(A)*(C*Vsolp/time_step + F');
    Vout(i) = Vsol(6);
    Vin(i) = Vsol(1);
    Vsolp = Vsol;
    i = i+1;

end

figure(6)
subplot(2,1,2)
plot(time,Vout)
title('Guass Function - Vout vs. Time w/ Noise - Time step = 0.0001')
grid on

subplot(2,1,1)
plot(time,Vin)
title('Guass Function - Vout vs. Time w/ Noise - Time step = 0.0001')
grid on

% Part (d

time_step=0.01;

Vsolp= zeros(9,1);

Cn = 0.0000001;
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;

A = C/(time_step) + G;

Vt=@(t) exp(-(1/2)*((t-0.06)/(0.03))^2);
Vout = zeros(10000,1);
Vin = zeros(10000,1);

i = 1;

time = zeros(10000,1);
for t=0:time_step:1

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

    time(i) = t;
    F(1,7) = Vt(t);
    F(1,9) = In;
```

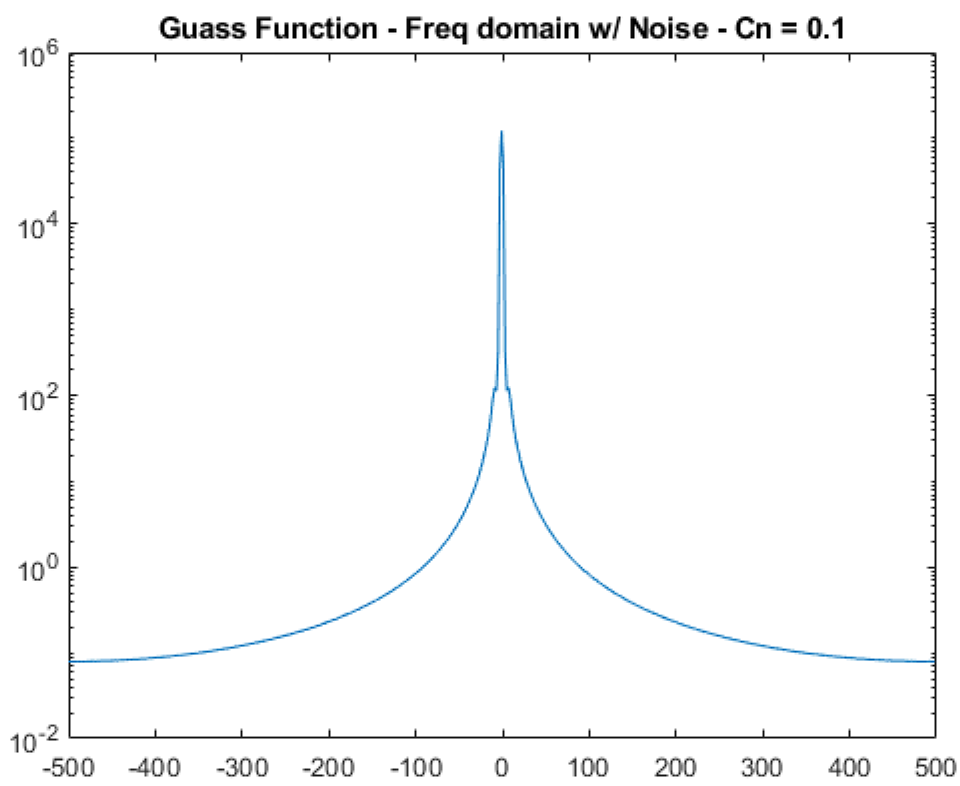
```
    Vsol = inv(A)*(C*Vsolp/time_step + F');
    Vout(i) = Vsol(6);
    Vin(i) = Vsol(1);
    Vsolp = Vsol;
    i = i+1;

end

figure(7)
subplot(2,1,2)
plot(time,Vout)
title('Guass Function - Vout vs. Time w/ Noise - Time step = 0.01')
grid on

subplot(2,1,1)
plot(time,Vin)
title('Guass Function - Vin vs. Time w/ Noise - Time step = 0.01')
grid on
```

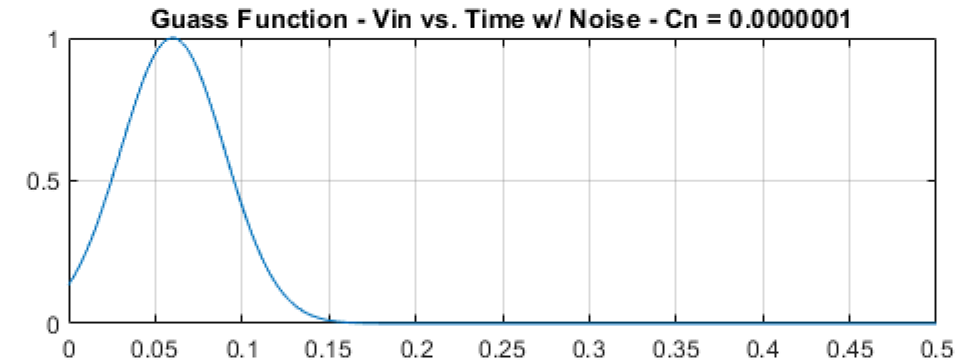



Gauss frquecny spectrum - Vout w/ Noise
10^6
10^4
10^2
10^0
10^-2
10^-4
-500
-400
-300
-200
-100
0
100
200
300
400
500

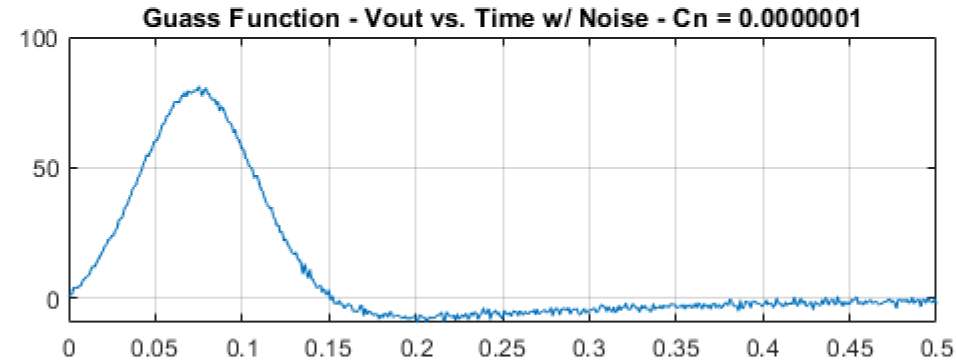

Gauss frquecny spectrum - Vin w/ Noise
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
-500
-400
-300
-200
-100
0
100
200
300
400
500


Guass Function - Vin vs. Time w/ Noise - Cn = 0.1
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
Guass Function - Vout vs. Time w/ Noise - Cn = 0.1
40
20
0
-20

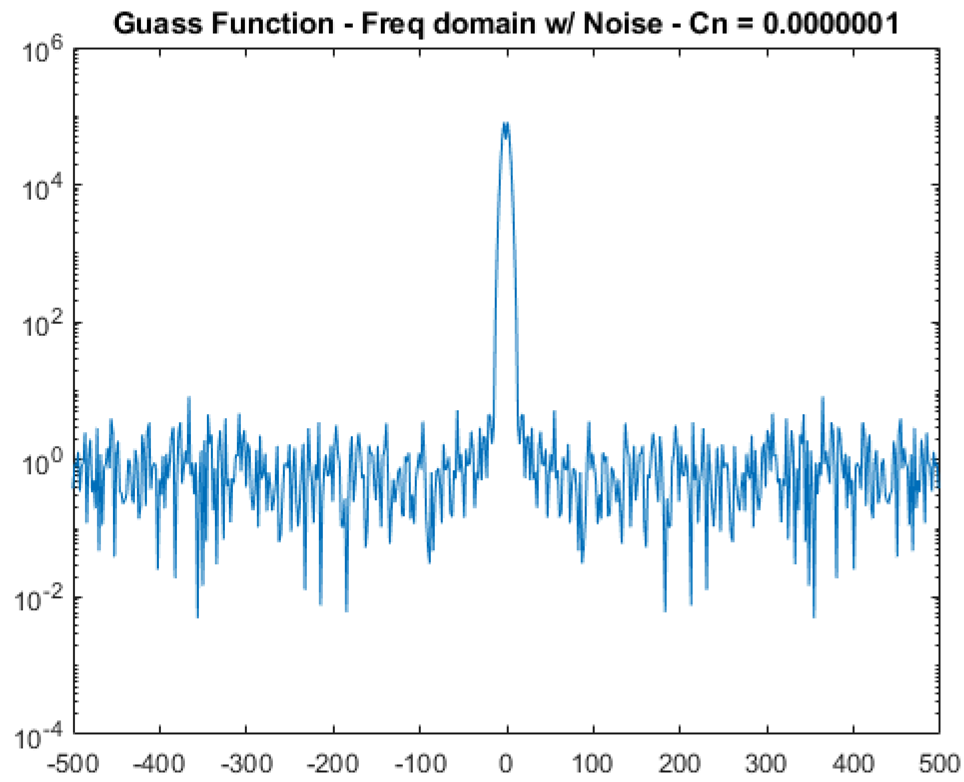

Guass Function - Freq domain w/ Noise - Cn = 0.1
10^6
10^4
10^2
10^0
10^{-2}
-500
-400
-300
-200
-100
0
100
200
300
400
500



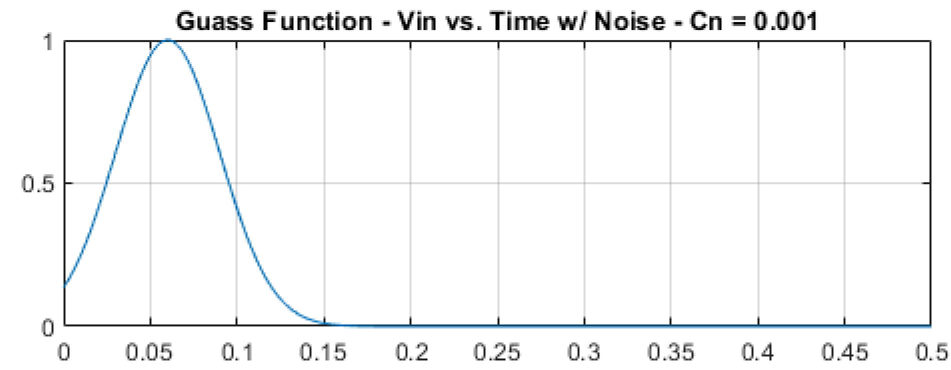
Guass Function - Vin vs. Time w/ Noise - Cn = 0.0000001
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

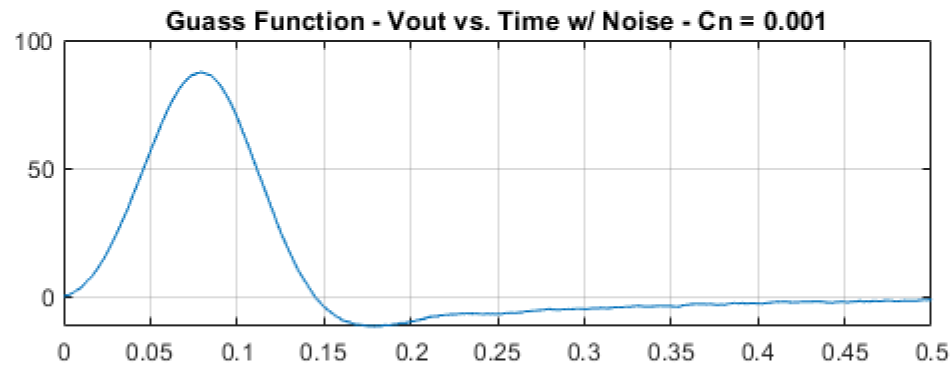


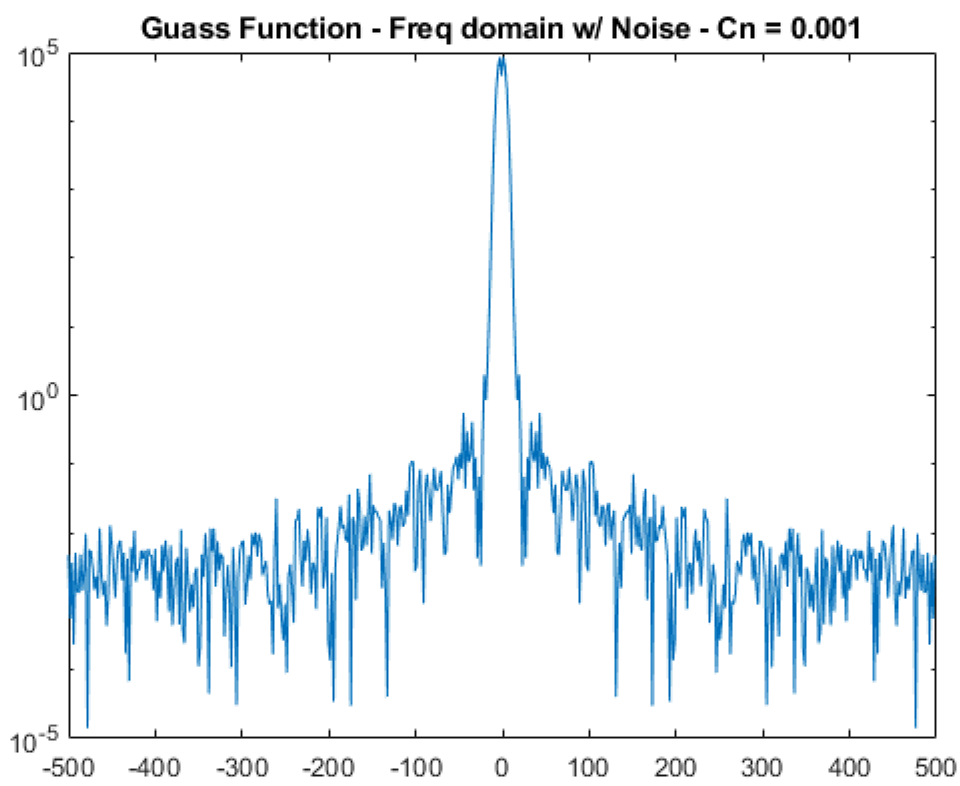
Guass Function - Vout vs. Time w/ Noise - Cn = 0.0000001
100
50
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

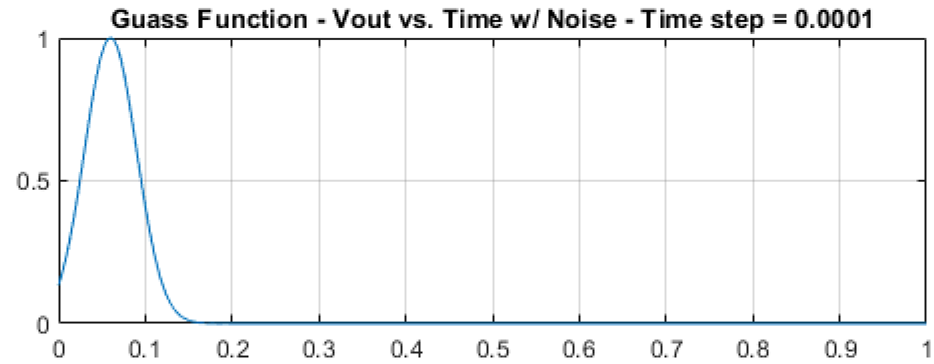


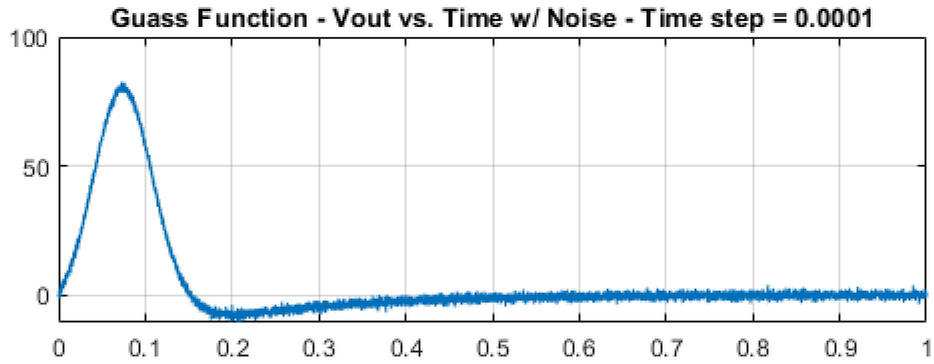
Guass Function - Freq domain w/ Noise - Cn = 0.0000001
10^6
10^4
10^2
10^0
10^{-2}
10^{-4}
-500
-400
-300
-200
-100
0
100
200
300
400
500

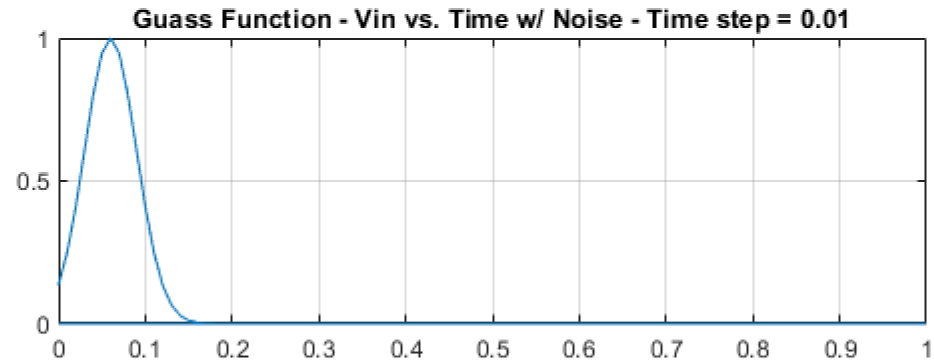

Guass Function - Vin vs. Time w/ Noise - Cn = 0.001
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

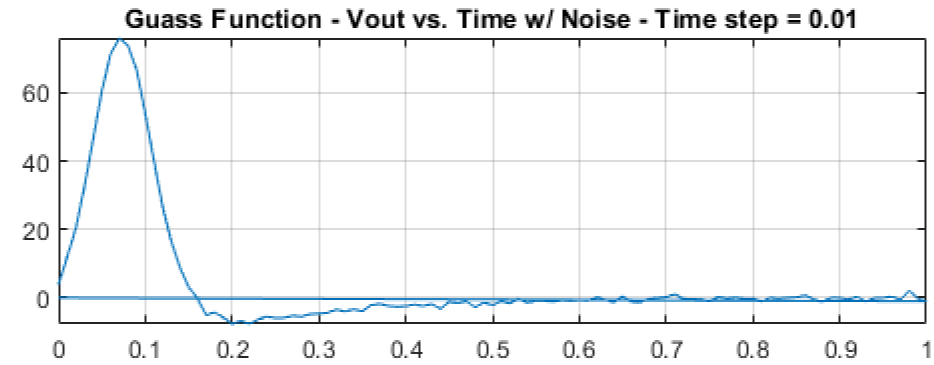

Guass Function - Vout vs. Time w/ Noise - Cn = 0.001
100
50
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5


Guass Function - Freq domain w/ Noise - Cn = 0.001
10^5
10^0
10^{-5}
-500
-400
-300
-200
-100
0
100
200
300
400
500


Guass Function - Vout vs. Time w/ Noise - Time step = 0.0001


Guass Function - Vout vs. Time w/ Noise - Time step = 0.0001


Guass Function - Vin vs. Time w/ Noise - Time step = 0.01


Guass Function - Vout vs. Time w/ Noise - Time step = 0.01

Incresing the value of Cn appears to decrease the bandwidth of the signal over the output.

Assignment 4 Part 4 - Andrew Paul

The fourth section of this assignment dicusses the implimentation of non-linearity in the ciruict.

Through research on the modelling of non-linearity in circuits it was found that the following method could be used to impliment the given changes in the circuit. First, a Jacobian matrix could be created and solved to give the solution to the non-linear voltage gneration which would be controlled by the current. The Newton Ralphson method could then be used to solve for the finite change in the value of the voltage generator. Similiarly to previous procedures, a matrix defining the output voltage would then be solved for to give a good approximation of the final output values.

Published with MATLAB® R2018b