ELEC4700 Assignment 4

Table of Contents

Reyad ElMahdy 101064879

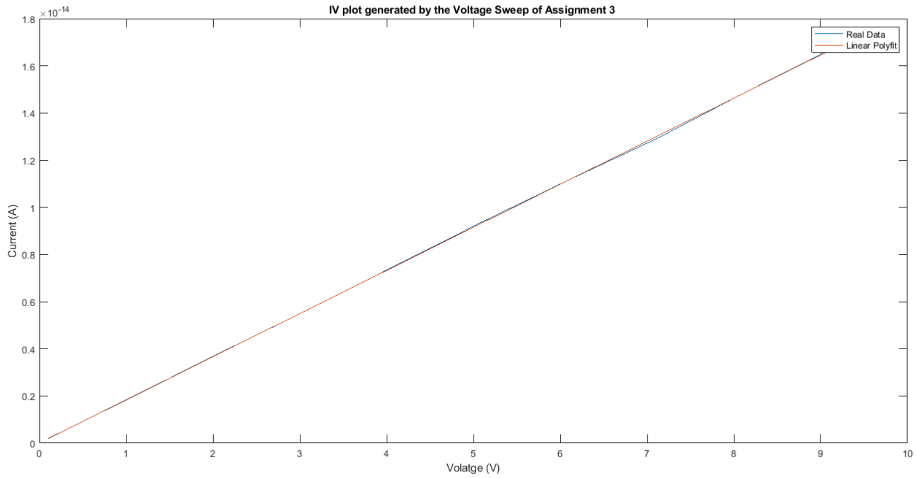
Question 1 - Voltage Sweep of Assignment 3

This section uses parts of the ode ffrom the previous assignment to perform a linear fit of the output current after doing a voltage sweep from 0.1V to 10V

```
clear; close; clc;
set(0, 'DefaultFigureWindowStyle', 'docked')
%Setting Constants
mElec = 9.11e-31; % Electron rest mass (kg)
mEff = 0.26*mElec; % Effective mass (kg)
kb = 1.381e-23; % Boltzmann's Constant (J/K)
T = 300; % Room Temperature (K)
L = 200e-9; % Length (m)
W = 100e-9; % Width (m)
q = 1.60218e-19; % Elemntary Charge (C)
V = 0.1:10;
ii = 1;
for v = 0.1:10
    Vvec = linspace(v, 0, 200);
    Vmat = zeros(100, 200);
    for k = 1:100
        Vmat(k,:) = Vvec;
    end

    [Ex,Ey] = gradient(-Vmat, 1e-9);

    Fx = q*Ex(1,1); % Force on the electrons (in Newtons)
    Fy = q*Ey(1,1); % Support for forces generated by E fields with
 vertical components

    ax = Fx/mEff; % Horizontal Acceleration
    ay = Fy/mEff; % Vertical Acceleration


    % Finding the thermal velocity
    Vth = sqrt((2*kb*T)/mEff); %Thermal Velocity in m/s
    mt = 0.2e-12; % mean time (s)
```

```
   % Modeling the electron paths

   numPar = 10; % Number of particles

   % Assigning particle positions
   posX = L.*rand(numPar,2);
   posY = W.*rand(numPar,2);
   posX(:,1) = posX(:,2);
   posY(:,1) = posY(:,2);

   spacialStep = sqrt(L^2+W^2)/1000;
   stepTime = spacialStep/Vth;

   % Assigning each particle a random velocity (acceleration will be
taken into account during the iteration)
   Vx = randn(numPar,2)*sqrt((kb*T)/mEff);
   Vy = randn(numPar,2)*sqrt((kb*T)/mEff);

   % Displacement per step of each electron
   dispX = stepTime*Vx(:,1);
   dispY = stepTime*Vy(:,1);

   colors = rand(numPar,3);

   pScatter = 1 - exp(-stepTime/mt); % Probability of an electron
scattering

   % Positions before collision
   collX = posX(:,1);
   collY = posY(:,1);

   ic = 0; % iteration counter for animated plot

   % J = n*(q/density)*mean(Vn), We can get the drift current density
from
   % this equation using the average velocity

   conc = 10^19; % concentration of electrons per m^2
   numPar2 = 10000;
   Vx = randn(numPar2,2)*sqrt((kb*T)/mEff);
   Vy = randn(numPar2,2)*sqrt((kb*T)/mEff);
   posX = L.*rand(numPar2,2);
   posY = W.*rand(numPar2,2);
   posX(:,1) = posX(:,2);
   posY(:,1) = posY(:,2);
   dispX = stepTime*Vx(:,1);
   dispY = stepTime*Vy(:,1);
   ic = 0;
   avgVx = 0;
   avgVy = 0;

   for i = 1:1000
       Vx = Vx + ax*stepTime;
       Vy = Vy + ay*stepTime;
```

```
        avgVx(i) = mean(Vx(:,2));
        avgVy(i) = mean(Vy(:,2));

        for j = 1:numPar2
            if(rand < pScatter)
                Vx(j,:) = randn(1)*sqrt((kb*T)/mEff) + ax*stepTime;
                Vy(j,:) = randn(1)*sqrt((kb*T)/mEff) + ay*stepTime;
                dispX(j,:) = stepTime*Vx(j,1);
                dispY(j,:) = stepTime*Vy(j,1);
                collX = posX(:,1);
                collY = posY(:,1);
            end
            if (posX(j,1)+dispX(j) > L)
                dispX(j,:) = stepTime*Vx(j,1);
                posX(j,2) = posX(j,1)+dispX(j)-L;
            elseif (posX(j,1)+dispX(j) < 0)
                dispX(j,:) = stepTime*Vx(j,1);
                posX(j,2) = posX(j,1)+dispX(j)+L;
            else
                dispX(j,:) = stepTime*Vx(j,1);
                posX(j,2) = posX(j,1)+dispX(j);
            end

            if ((posY(j,1)+dispY(j) > W)|| (posY(j,1)+dispY(j) < 0))
                dispY(j) = -dispY(j);
                posY(j,2) = posY(j,1)+dispY(j);
            else
                posY(j,2) = posY(j,1)+dispY(j);
            end
        end
    end
    J(ii) = mean(W*q*conc.*avgVx);
    ii = ii + 1;
end
I = J*W*L;
poly = polyfit(V, I,1);
fit = poly(1)*V + poly(2);
figure(1)
plot(V, I)
hold on;
plot(V, fit)
title('IV plot generated by the Voltage Sweep of Assignment 3 ')
xlabel('Volatge (V)')
ylabel('Current (A)')
legend('Real Data','Linear Polyfit')
hold off;
```



Question 2 - Getting R3

Normally the resistance would just be equal to the inverse of the slope of the linear fit. i.e R3 = 1/poly(1); However I am unsure of the value I'm getting from the previous assignment, the units might be off or I missed a step somewhere, so for the remainder of the assignment I'll be using R3 = 10, just so that the data for the rest of the assignment isn't skewed by R3.

Question 3 - Differential Equations and DC/AC sweeps

a) The differential equations were found by performing KCL on each of the nodes in the circuit i)

Node 1: $V1 = Vin$ I1 = (V1-V2)/R1 + C(d(V1-V2))/dt

Node 2: $$ (V1-V2)/R1 + C(d(V1-V2))/dt + V2/R2 + iL = 0

Node 3: $V3/R3 = iL = I3$ V2 - V3 = L(d(iL))/dt

Node 4: $V4 = aI3$ I4 = (V4-V5)/R4

Node 5: $$ Vo/Ro = (V4-V5)/R4

ii) The differential equations in the frequency domain Node 1: $V1 = Vin$ I1 = G1(V1-V2) + jw-C(d(V1-V2))

Node 2: $$ G1(V1-V2) + jwC(V1-V2) + V2G2 + jwL(V2-V3) = 0

Node 3: $$ jwL(V2-V3) = V3G3

Node 4: $V4 = aI3$ I4 = G4(V4-V5)

Node 5: $$ VoGo = G4(V4-V5)

```
V1,    V2,     iL,    V3,       V4,     Vo
```

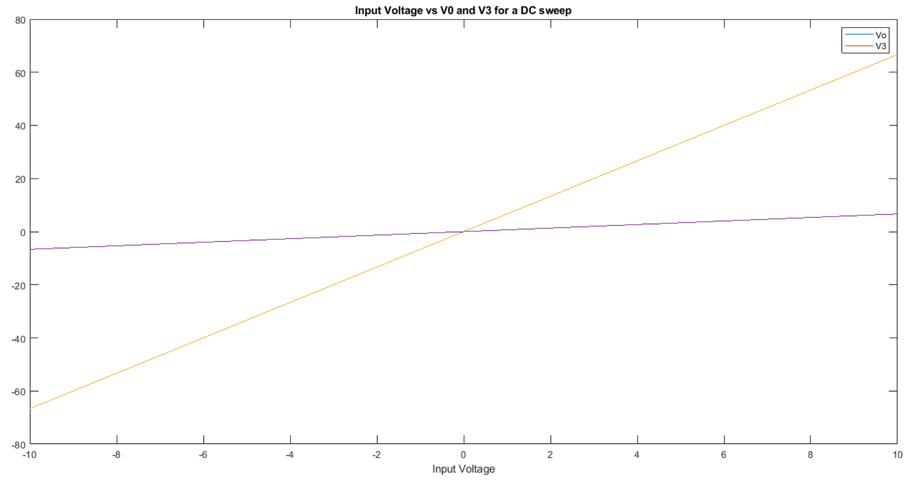
```
G:
     1,      0,       0,      0,         0,      0
 -1/R1, 1/R2+1/R1,   0,      0,         0,      0
     0,     -1,       0,      1,         0,      0
     0,      0,      -1,   1/R3,         0,      0
     0,      0,       0,  -a/R3,         1,      0
     0,      0,       0,      0,     -1/R4,  1/R4+1/Ro
```

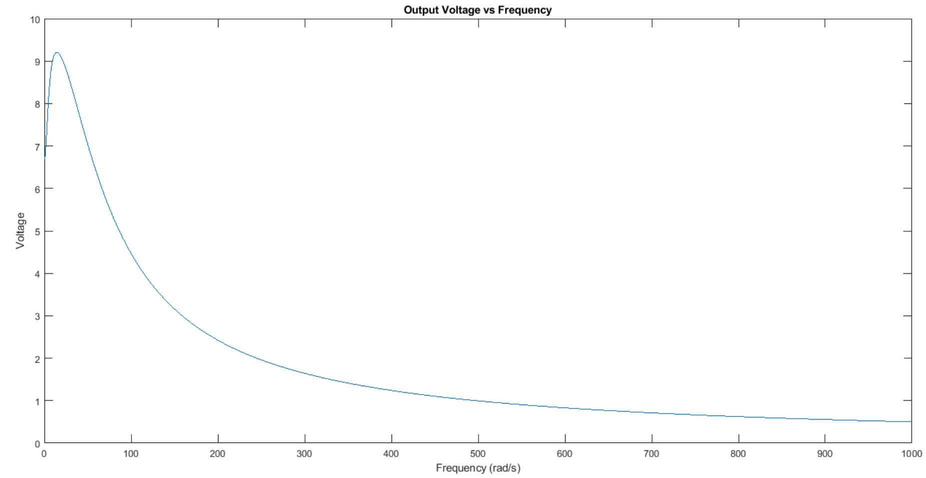
C: 0, 0, 0, 0, 0, 0 -C, C, 0, 0, 0, 0 0, 0, L, 0, 0, 0 0, 0, 0, 0, 0, 0 0, 0, 0, 0, 0, 0 0, 0, 0, 0, 0, 0

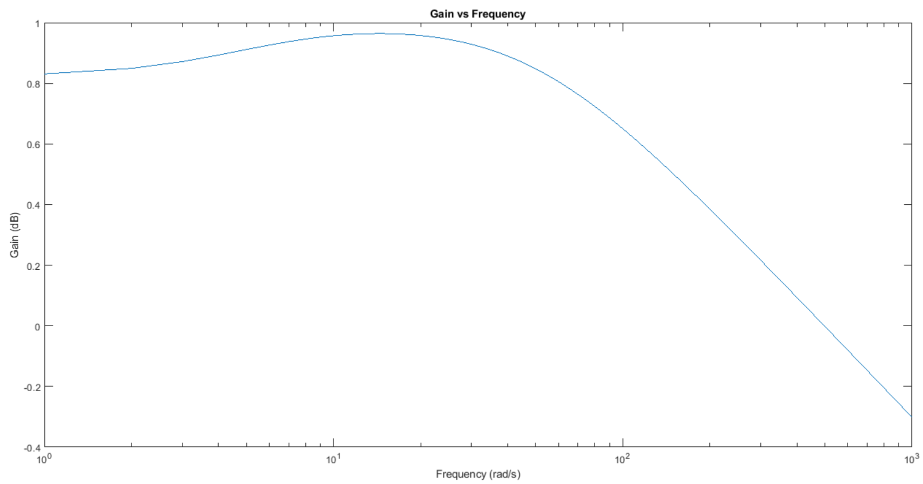
b) i) DC Sweep

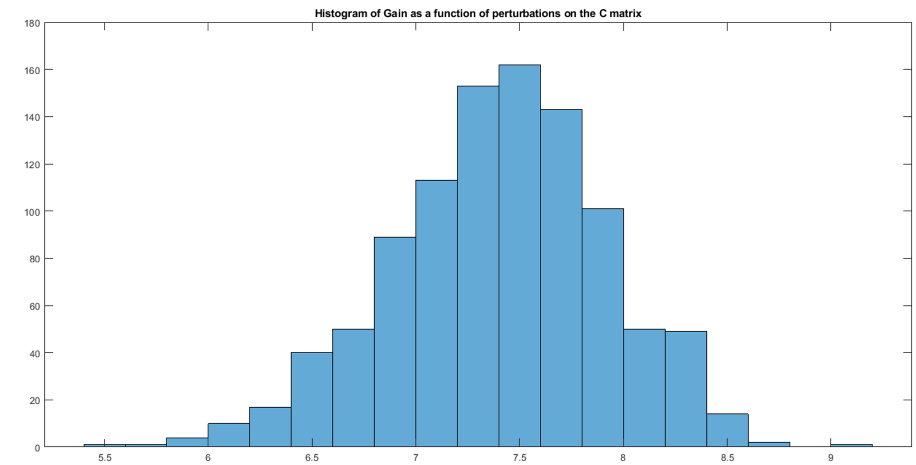
```
R1 = 1;
C = 0.25;
R2 = 2;
L = 0.2;
a = 100;
R3 = 10;
R4 = 0.1;
Ro = 1000;

G = [
    1, 0, 0, 0, 0, 0;
    -1/R1, 1/R2+1/R1, 0, 0, 0, 0;
    0, -1, 0, 1, 0, 0;
    0, 0, -1, 1/R3, 0, 0;
    0, 0, 0, -a/R3, 1, 0;
    0, 0, 0, 0, -1/R4, 1/R4+1/Ro
    ];
Cmat = zeros(6,6);
Cmat(2,1) = -C;
Cmat(2,2) = C;
Cmat(3,3) = L;

Vo = zeros(1,20);
V3 = zeros(1,20);
F = zeros(1,6);
Vin = -10:10;
for V = Vin
    F(1) = V;
    Vmat = G\F';
    Vo(V+11) = Vmat(6);
    V3(V+11) = Vmat(4);
end

figure(2)
plot(Vin,Vo)
hold on
plot(Vin,V3)
xlabel('Input Voltage')
legend('Vo', 'V3')
title ('Input Voltage vs V0 and V3 for a DC sweep')

% ii) AC case plot
```

```
F(1) = 1;
Vo = zeros(1,1000);
w = 0:1000;
j = sqrt(-1);
for i = w
    G_ac = G + j*i*Cmat;
    Vmat = G_ac\F';
    Vo(i+1) = Vmat(6);
end
figure(3)
plot(w,abs(Vo))
xlabel('Frequency (rad/s)')
ylabel('Voltage')
title ('Output Voltage vs Frequency')


figure(4)
semilogx(w,log10(abs(Vo)))
xlabel('Frequency (rad/s)')
ylabel('Gain (dB)')
title ('Gain vs Frequency')

% iii) Gain as a function of random perturbations

AV = zeros(1,1000);
for i = 1:1000
    Cmat(2, 1) = normrnd(-C, 0.05);
    Cmat(2, 2) = normrnd(C, 0.05);
    Cmat(3, 3) = normrnd(L, 0.05);
    Vmat = (G+j*pi*Cmat)\F';
    AV(i) = Vmat(6)/F(1);
end
figure(5)
histogram(real(AV))
title('Histogram of Gain as a function of perturbations on the C
 matrix')
```




Output Voltage vs Frequency
Voltage
Frequency (rad/s)


Gain vs Frequency
Gain (dB)
Frequency (rad/s)


Histogram of Gain as a function of perturbations on the C matrix

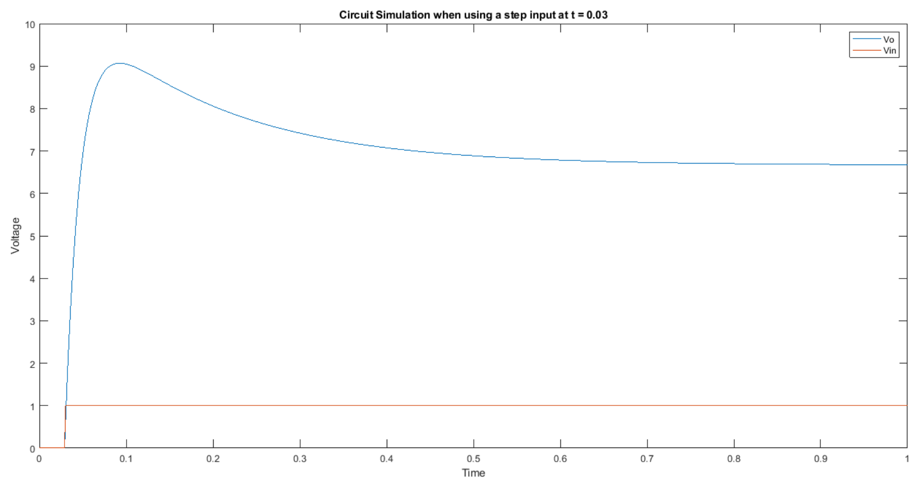
Question 4 Simulation of the Transient Circuit

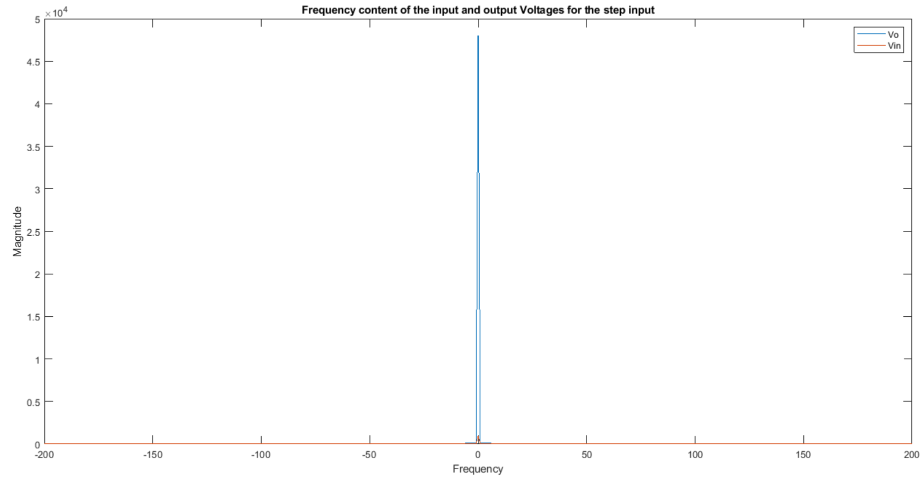
The next part of the assignment is to perform a transient circuit simulation, which requires us to solve the time domain equation of the circuit. $$ frac{C(dV)}{dt} + GV = F A numerical solution of the equation can be done using the finite difference method.

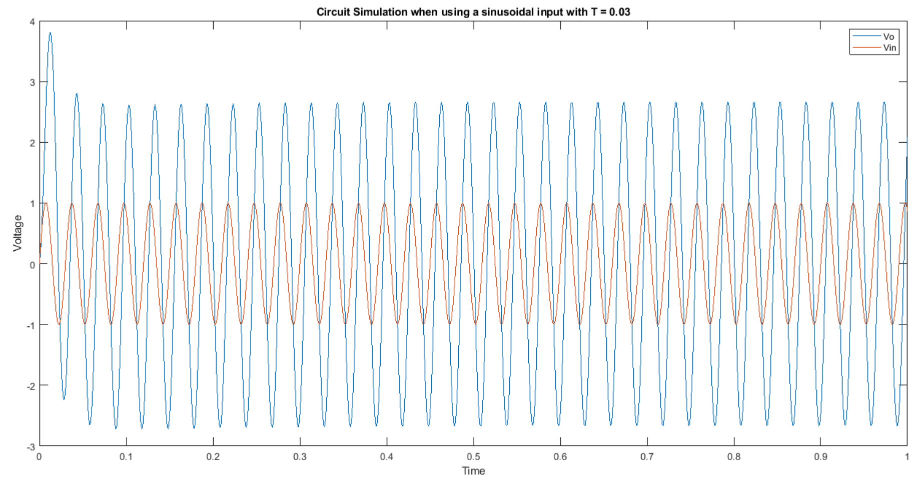
```
% a) This circuit behaves like a low pass filter
% b) The gain should be constant and unattenuated at low frequencies
 and
% then there should be a constant drop in the gain starting from the
 circuit's
% cutoff frequency
% c) Using the Finite difference method we get
% $$ (frac{C(dV)}{dt} + GV = F

% d)
% In order to find plot the response of Vin and Vout in the time and
% frequency domain we neeed to simulate the circuit for each different
 type of input we use
% First we simulate the circuit with a step input changing the input
% voltage from 0 to 1 at t = 0.03
Vin = zeros(1, 1000); Vo = Vin; V3 = Vo;
V = zeros(6,1); F = zeros(1,6);
t = linspace(0,1,1000);

for i = 1:1000
    if t(i) < 0.03
        Vin(i) = 0;
    else
        Vin(i) = 1;
    end
    F(1) = Vin(i);
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end

figure(6)
plot(t, Vo)
title('Circuit Simulation when using a step input at t = 0.03')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;

f = fft(Vo);
fx = (-1000/2:1000/2-1);
p = abs(fftshift(f)).^2/1000;
figure(7)
plot(fx,p)
xlabel('Frequency')
ylabel('Magnitude')
```

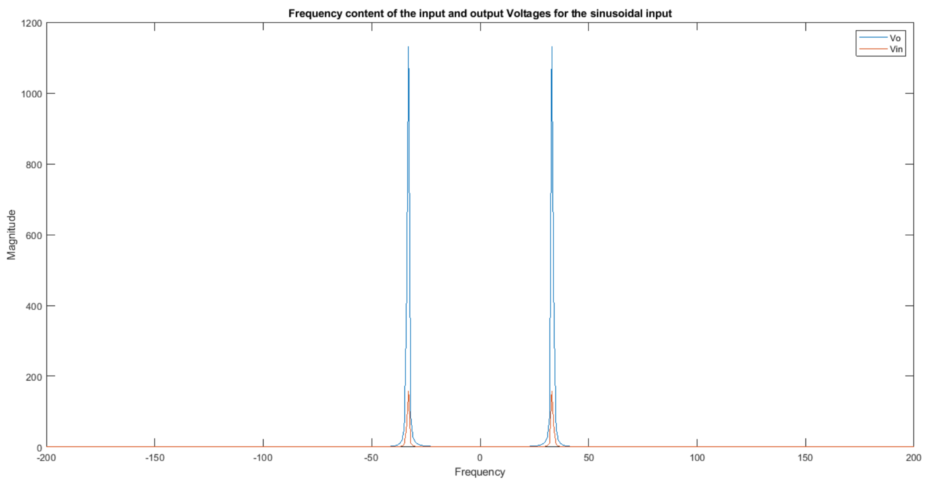
```
title('Frequency content of the input and output Voltages for the step
 input')
hold on;
p = abs(fftshift(fft(Vin))).^2/1000;
plot(fx,p)
legend('Vo', 'Vin')
xlim([-200 200])
hold off;

% Next, we repeat the simulation using a sinusoidal input with A = 1V
 and
% T = 0.03s
V = zeros(6,1); F = zeros(1,6);
for i = 1:1000
    Vin(i) = sin(2*pi*(1/.03)*t(i));
    F(1) = Vin(i);
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(8)
plot(t, Vo)
title('Circuit Simulation when using a sinusoidal input with T =
 0.03')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;

f = fft(Vo);
fx = (-1000/2:1000/2-1);
p = abs(fftshift(f)).^2/1000;
figure(9)
plot(fx,p)
xlabel('Frequency')
ylabel('Magnitude')
title('Frequency content of the input and output Voltages for the
 sinusoidal input')
hold on;
p = abs(fftshift(fft(Vin))).^2/1000;
plot(fx,p)
legend('Vo', 'Vin')
xlim([-200 200])
hold off;

% Finally, we repeat the simulation one more time except this time
 using a
% gaussian pulse of magnitude 1, 0.03s std deviation, and 0.06s delay
% Next, we repeat the simulation using a sinusoidal input with A = 1V
 and
% T = 0.03s
V = zeros(6,1); F = zeros(1,6);
for i = 1:1000
```

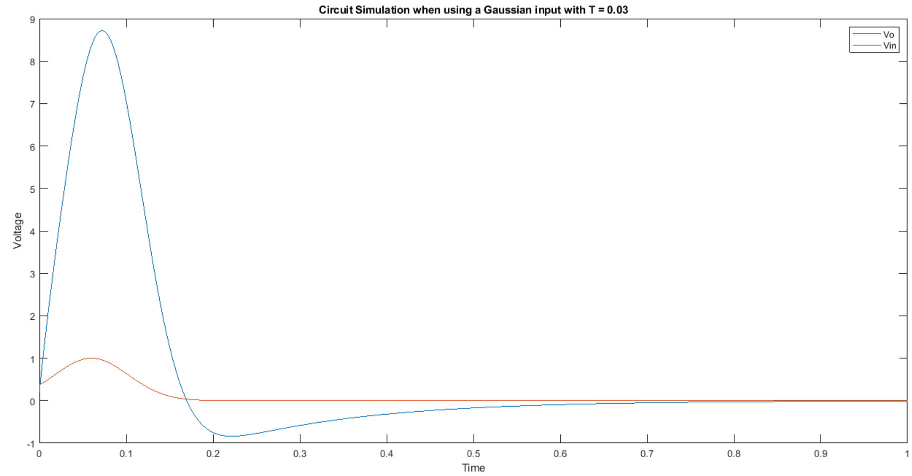
```
    Vin(i) = exp(-((t(i)-0.06)/(2*0.03))^2);
    F(1) = Vin(i);
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(10)
plot(t, Vo)
title('Circuit Simulation when using a Gaussian input with T = 0.03')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;

f = fft(Vo);
fx = (-1000/2:1000/2-1);
p = abs(fftshift(f)).^2/1000;
figure(11)
plot(fx,p)
xlabel('Frequency')
ylabel('Magnitude')
title('Frequency content of the input and output Voltages for the
 Gaussian pulse input')
hold on;
p = abs(fftshift(fft(Vin))).^2/1000;
plot(fx,p)
legend('Vo', 'Vin')
xlim([-200 200])
hold off;
```

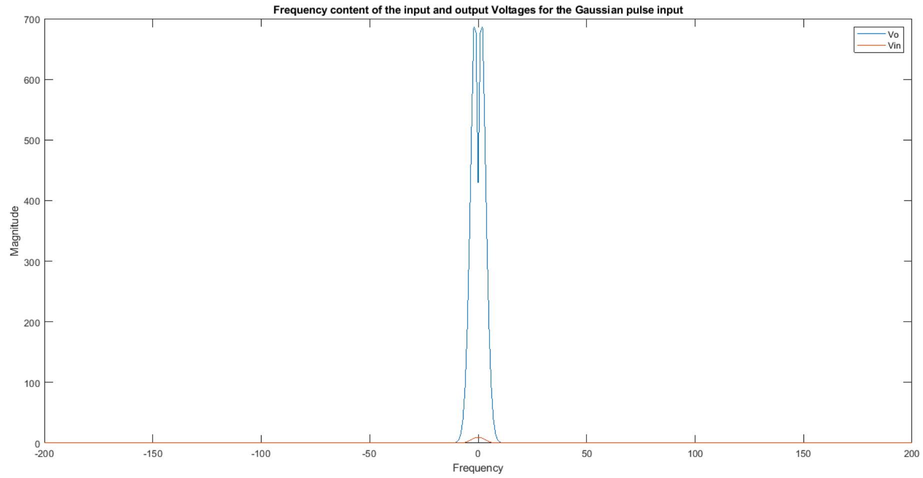



Frequency content of the input and output Voltages for the step input
× 10^4
Magnitude
Frequency
Vo
Vin
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-200
-150
-100
-50
0
50
100
150
200


Circuit Simulation when using a sinusoidal input with T = 0.03
Voltage
Time
Vo
Vin
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


Frequency content of the input and output Voltages for the sinusoidal input
Magnitude
Frequency
Vo
Vin
1200
1000
800
600
400
200
0
-200
-150
-100
-50
0
50
100
150
200





Question 5 Simulation of the Circuit With Noise

since all the previous simulations used certain values for the C and G matrices, adding another curent source and a capacitor to the circuit means we neeed another equation to be able to properly model the circuit. As a result the C and G matrices need to be reformulated to account for this.

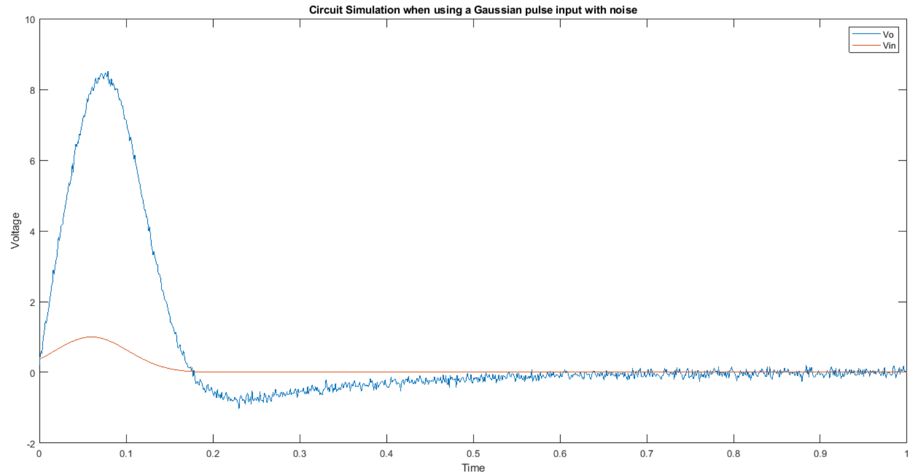
```
Cn = 0.00001;

G = zeros(7); Cmat = G;
G = [
    1, 0, 0, 0, 0, 0, 0;
    -1/R1, 1/R2+1/R1, 0, 0, 0, 0, 0;
    0, -1, 0, 1, 0, 0, 0;
    0, 0, -1, 1/R3, 0, 0, 1;
    0, 0, 0, -a/R3, 1, 0, 0;
    0, 0, 0, 0, -1/R4, 1/R4+1/Ro, 0;
    0, 0, 0, 0, 0, 0, 1
    ];
```

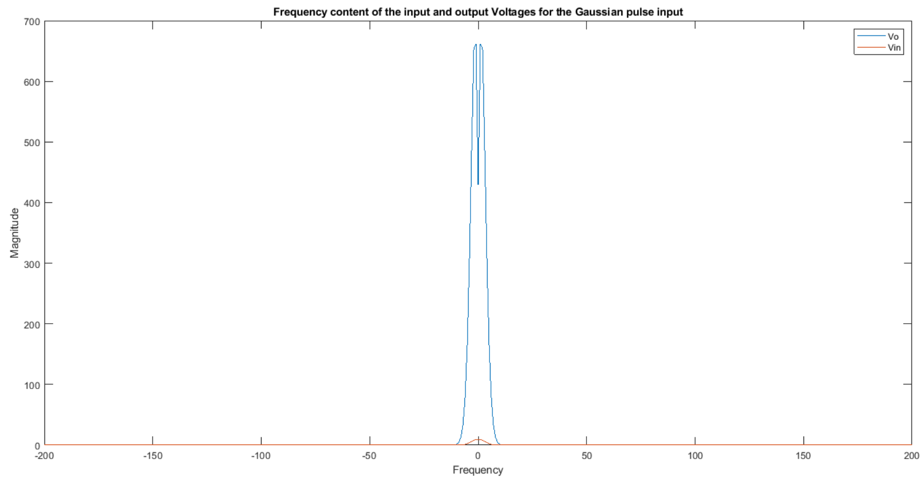
```
Cmat(2,1) = -C;
Cmat(2,2) = C;
Cmat(3,3) = L;
Cmat(4,4) = Cn;

V = zeros(7,1); F = zeros(1,7);
for i = 1:1000
    Vin(i) = exp(-((t(i)-0.06)/(2*0.03))^2);
    F(1) = Vin(i);
    F(7) = randn*0.001;
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(12)
plot(t, Vo)
title('Circuit Simulation when using a Gaussian pulse input with
 noise')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;

f = fft(Vo);
fx = (-1000/2:1000/2-1);
p = abs(fftshift(f)).^2/1000;
figure(13)
plot(fx,p)
xlabel('Frequency')
ylabel('Magnitude')
title('Frequency content of the input and output Voltages for the
 Gaussian pulse input')
hold on;
p = abs(fftshift(fft(Vin))).^2/1000;
plot(fx,p)
legend('Vo', 'Vin')
xlim([-200 200])
hold off;

% Next the Capacitance will gradually increase to see the effect of
% increased Cn on the response
Cmat(4,4) = 0.0001;
V = zeros(7,1); F = zeros(1,7);
for i = 1:1000
    Vin(i) = exp(-((t(i)-0.06)/(2*0.03))^2);
    F(1) = Vin(i);
    F(7) = randn*0.001;
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(14)
plot(t, Vo)
```

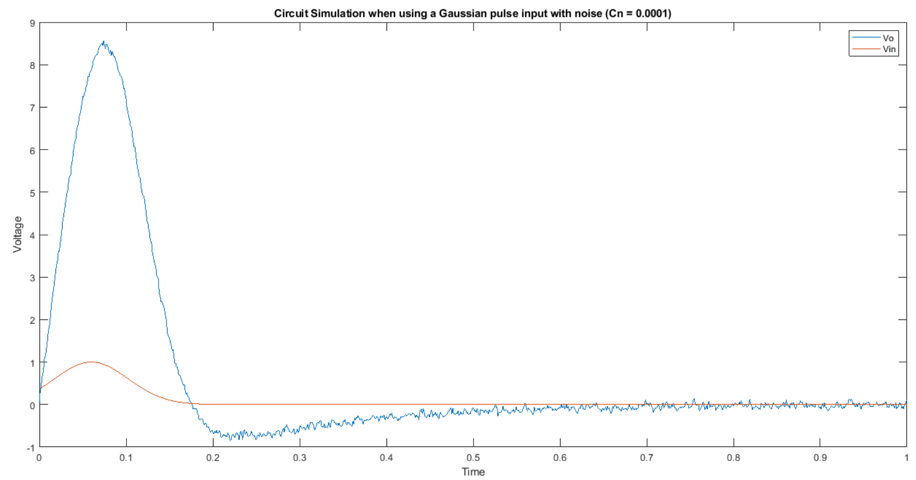
```
title('Circuit Simulation when using a Gaussian pulse input with noise
 (Cn = 0.0001)')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;

f = fft(Vo);
fx = (-1000/2:1000/2-1);
p = abs(fftshift(f)).^2/1000;
figure(15)
plot(fx,p)
xlabel('Frequency')
ylabel('Magnitude')
title('Frequency content of the input and output Voltages for the
 Gaussian pulse input (Cn = 0.0001)')
hold on;
p = abs(fftshift(fft(Vin))).^2/1000;
plot(fx,p)
legend('Vo', 'Vin')
xlim([-200 200])
hold off;

Cmat(4,4) = 0.001;
V = zeros(7,1); F = zeros(1,7);
for i = 1:1000
    Vin(i) = exp(-((t(i)-0.06)/(2*0.03))^2);
    F(1) = Vin(i);
    F(7) = randn*0.001;
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(16)
plot(t, Vo)
title('Circuit Simulation when using a Gaussian pulse input with noise
 (Cn = 0.001)')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;

f = fft(Vo);
fx = (-1000/2:1000/2-1);
p = abs(fftshift(f)).^2/1000;
figure(17)
plot(fx,p)
xlabel('Frequency')
ylabel('Magnitude')
title('Frequency content of the input and output Voltages for the
 Gaussian pulse input (Cn = 0.001)')
```

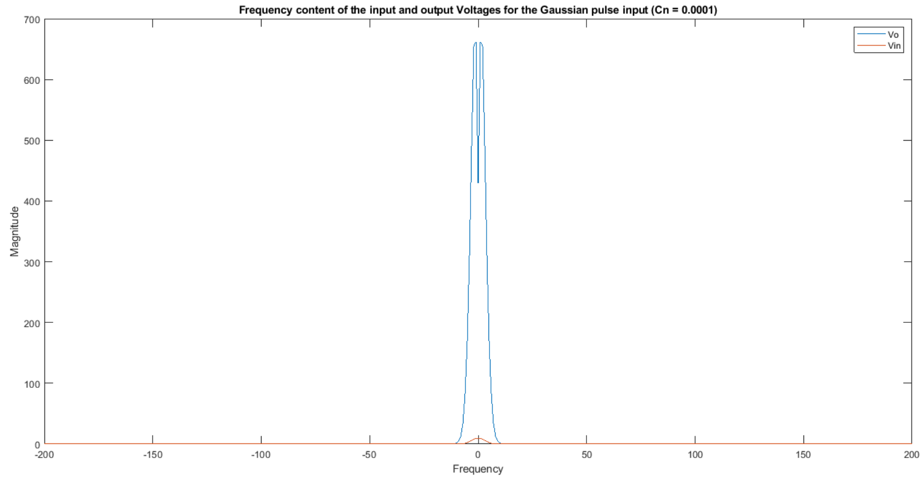
```
hold on;
p = abs(fftshift(fft(Vin))).^2/1000;
plot(fx,p)
legend('Vo', 'Vin')
xlim([-200 200])
hold off;

Cmat(4,4) = 0.01;
V = zeros(7,1); F = zeros(1,7);
for i = 1:1000
    Vin(i) = exp(-((t(i)-0.06)/(2*0.03))^2);
    F(1) = Vin(i);
    F(7) = randn*0.001;
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(18)
plot(t, Vo)
title('Circuit Simulation when using a Gaussian pulse input with noise
 (Cn = 0.01)')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;

f = fft(Vo);
fx = (-1000/2:1000/2-1);
p = abs(fftshift(f)).^2/1000;
figure(19)
plot(fx,p)
xlabel('Frequency')
ylabel('Magnitude')
title('Frequency content of the input and output Voltages for the
 Gaussian pulse input (Cn = 0.01)')
hold on;
p = abs(fftshift(fft(Vin))).^2/1000;
plot(fx,p)
legend('Vo', 'Vin')
xlim([-200 200])
hold off;

% As shown in the above plots, the bandwidth of the plot in the
 frequency domain and the frequency
% in the time domain change more drastically with a greater noise
% capacitance. The frequency of the plots in the time domain decreases
 with
% increased capacitance and distance between the peaks in the
 frequency
% domain increases with increased noise capacitance

% The next step was to examine the how changing the number of
 timesteps
```

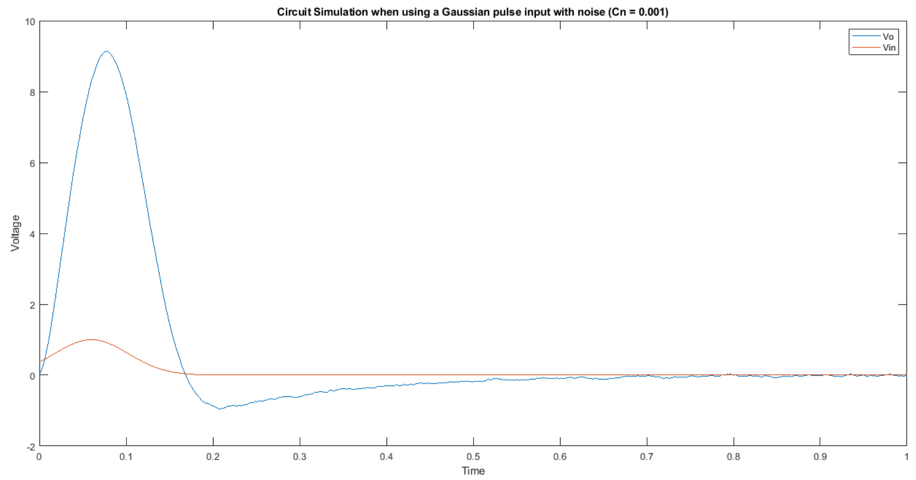
```
% affected the simulation
t = linspace(0,1,100);
Vin = zeros(1,100); Vo = Vin;
Cmat(4,4) = Cn;
V = zeros(7,1); F = zeros(1,7);
for i = 1:100
    Vin(i) = exp(-((t(i)-0.06)/(2*0.03))^2);
    F(1) = Vin(i);
    F(7) = randn*0.001;
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(20)
plot(t, Vo)
title('Circuit Simulation when using a Gaussian pulse input with noise
 (100 steps)')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off

t = linspace(0,1,10000);
Vin = zeros(1,10000); Vo = Vin;
V = zeros(7,1); F = zeros(1,7);
for i = 1:10000
    Vin(i) = exp(-((t(i)-0.06)/(2*0.03))^2);
    F(1) = Vin(i);
    F(7) = randn*0.001;
    V = (Cmat/t(2) + G)\((Cmat*V/t(2)) + F');
    Vo(i) = V(6);
end
figure(21)
plot(t, Vo)
title('Circuit Simulation when using a Gaussian pulse input with noise
 (10000 steps)')
xlabel('Time')
ylabel('Voltage')
hold on
plot(t,Vin)
legend('Vo', 'Vin')
hold off;
```

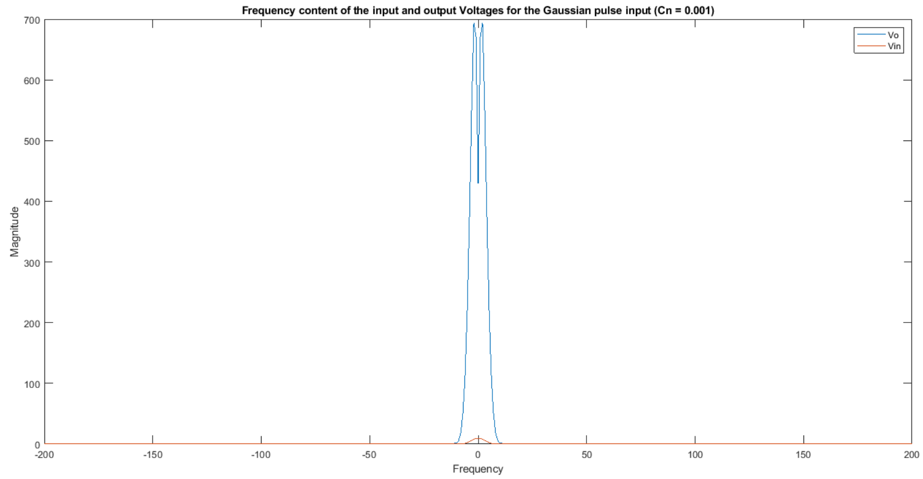

Circuit Simulation when using a Gaussian pulse input with noise
Vo
Vin
Voltage
Time
10
8
6
4
2
0
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

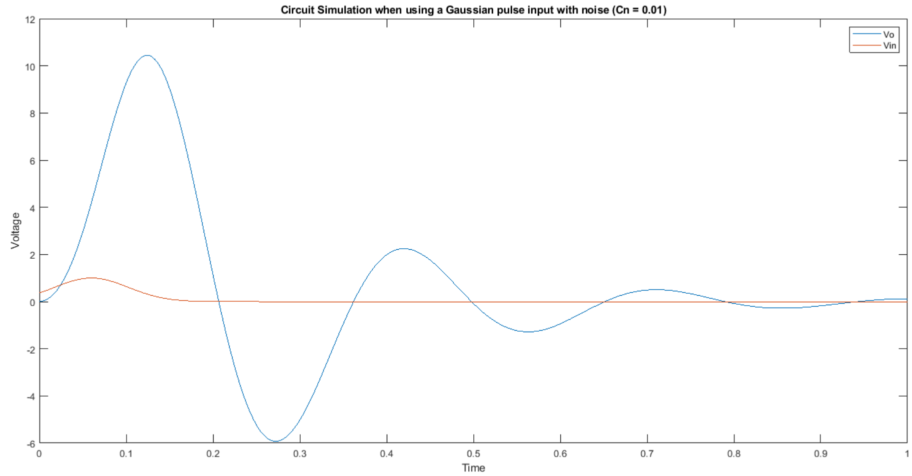

Frequency content of the input and output Voltages for the Gaussian pulse input
Vo
Vin
Magnitude
Frequency
700
600
500
400
300
200
100
0
-200
-150
-100
-50
0
50
100
150
200

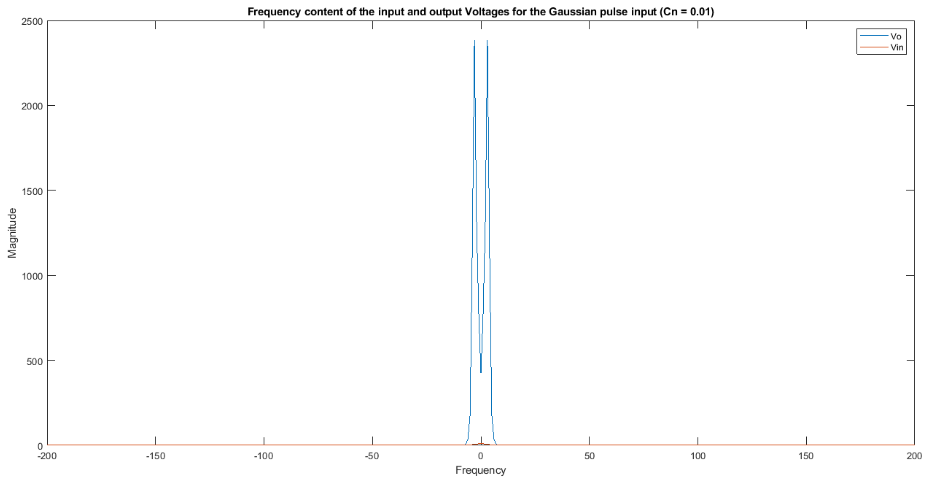

Circuit Simulation when using a Gaussian pulse input with noise (Cn = 0.0001)
Vo
Vin
Voltage
Time
9
8
7
6
5
4
3
2
1
0
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

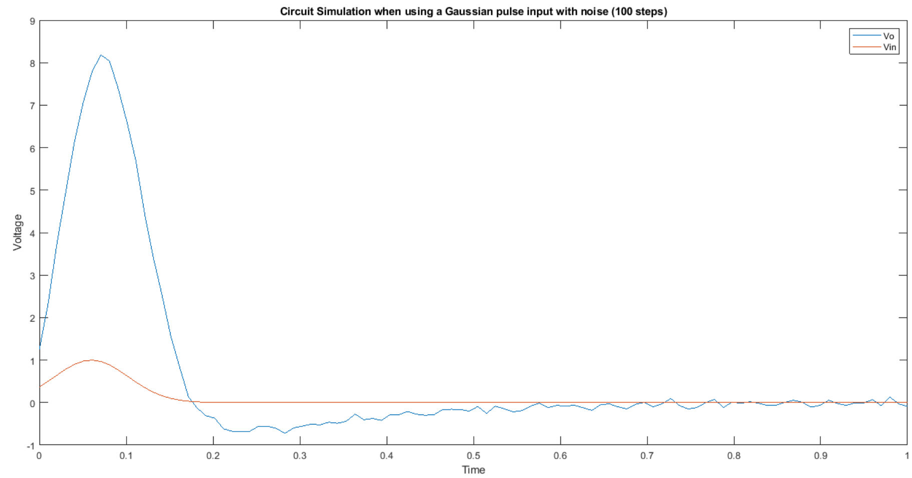

Frequency content of the input and output Voltages for the Gaussian pulse input (Cn = 0.0001)
Vo
Vin
Magnitude
Frequency
700
600
500
400
300
200
100
0
-200
-150
-100
-50
0
50
100
150
200

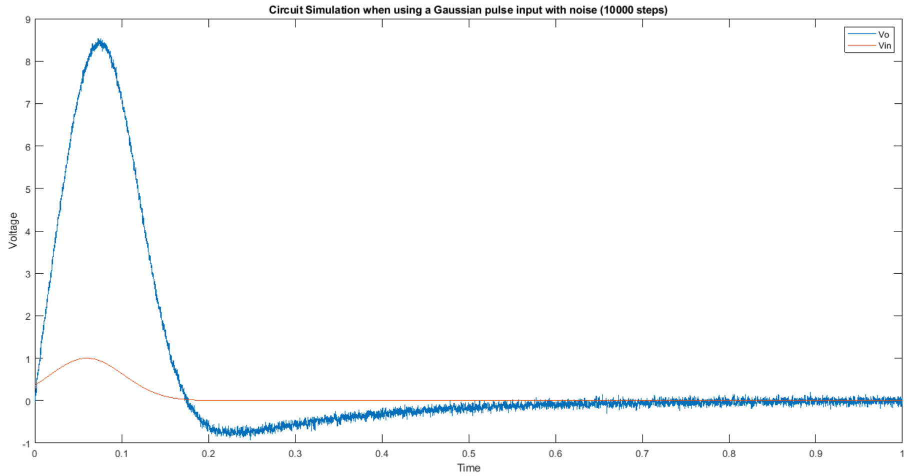

Circuit Simulation when using a Gaussian pulse input with noise (Cn = 0.001)
Vo
Vin
Voltage
Time
10
8
6
4
2
0
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency content of the input and output Voltages for the Gaussian pulse input (Cn = 0.001)
Vo
Vin
Magnitude
Frequency
700
600
500
400
300
200
100
0
-200
-150
-100
-50
0
50
100
150
200


Circuit Simulation when using a Gaussian pulse input with noise (Cn = 0.01)
Vo
Vin
Voltage
Time
12
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 the input and output Voltages for the Gaussian pulse input (Cn = 0.01)
Vo
Vin
Magnitude
Frequency
2500
2000
1500
1000
500
0
-200
-150
-100
-50
0
50
100
150
200


Circuit Simulation when using a Gaussian pulse input with noise (100 steps)
Vo
Vin
Voltage
Time
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



Question 5 - Non Linearity

If the voltage were changed the equations used to formulate the G matrix would have to be changed to account for this change, as a result the G matrix wold also have to change to reflect this. Due to the nonlinearity of this voltage, the matrix equation would also have to be altered by adding a B vector to the left hand side to account for this nonlinearity. The algorithm used to solve this equation would also have to change as gaussian elimination wouldn't work for a nonlinear system. It might be better to use another interative numerical method like the Newton Raphson method find estimates of the roots.

Published with MATLAB® R2020b