ELEC 4700 Assignment 4

Table of Contents

Tom Palmer - 101045113

1 - Current-Voltage Charicteristics

By solving for the electric field for a varying voltage, it is possible to assess the average current within the device using the equations given below.

$$J = \sigma * E$$

$$I = J/A$$

Since the bottle neck is not being changed, and thus the effective resistance of the device is not changing, we expect the current and the voltage of the device to be linearly related based on the equation...

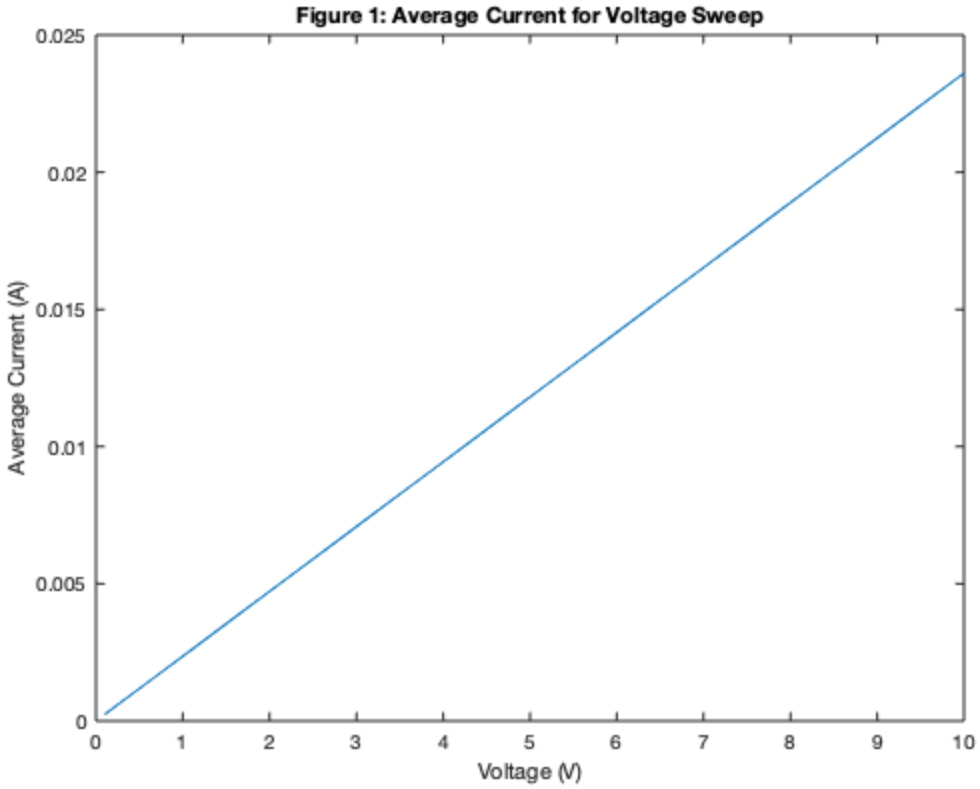
$$V = I * R$$

...and the code below demonstraits this.

```
% Setup
clear
clc

% Define Matrix
nx = 200;
ny = 100;
G = sparse(nx*ny,nx*ny);
B = zeros(nx*ny,1);
%S = ones(nx,ny).*(1/(6.2e2)); % Accounts for conductivity of Si
S = ones(nx,ny);
A = zeros(1,50);
V0 = linspace(0.1,10,50);

for i = 1:nx
    for j = 1:ny
        if i >= 80 && i <= 120
            if j <= 40 || j >= 60
                S(i,j) = 1e-10;
            end
        end
    end
end

for count = 1:50
```

```
for i = 1:nx
    for j = 1:ny
        n = j + (i-1)*ny;

        if i == 1
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = V0(count);
        elseif i == nx
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = 0;


        elseif j == 1
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nyp = j+1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            ryp = (S(i,j) + S(i,j+1))/2;

            G(n,n) = -(rxm+rxp+ryp);
            G(n,nxm) = rxm;
            G(n,nxp) = rxp;
            G(n,nyp) = ryp;

        elseif j == ny
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            rym = (S(i,j) + S(i,j-1))/2;

            G(n,n) = -(rxm+rxp+rym);
            G(n,nxm) = rxm;
            G(n,nxp) = rxp;
            G(n,nym) = rym;

        else
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;
            nyp = j+1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            rym = (S(i,j) + S(i,j-1))/2;
            ryp = (S(i,j) + S(i,j+1))/2;

            G(n,n) = -(rxm+rxp+rym+ryp);
```

```
                G(n,nxm) = rxm;
                G(n,nxp) = rxp;
                G(n,nym) = rym;
                G(n,nyp) = ryp;
            end
        end
    end

        solve = G\B;
        data_z = zeros(nx,ny);

        for i = 1:nx
            for j = 1:ny
                position = j + (i-1)*ny;
                data_z(i,j) = solve(position,1);
            end
        end

        [Ex,Ey] = gradient(-1.*data_z);
        Jx = S.*Ex;
        Jy = S.*Ey;

        % For current we will take the average of the current density
 across the
        % area of the material.

        J = sqrt(Jx.^2 + Jy.^2);
        A(1,count) = mean(mean(J));
end

figure(1)
plot(V0,A)
xlabel('Voltage (V)')
ylabel('Average Current (A)')
title('Figure 1: Average Current for Voltage Sweep')
```



2 - Resistance Value

A linear fit can now be applied to the data obtained in Part 1. Since the slope of this fit will be voltage divided by current, this will return the value of the resistane of the device. This is given by the code below.

```
R = polyfit(V0,A,1);
Resistance = R(1)
```

```
Resistance =

    0.0024
```

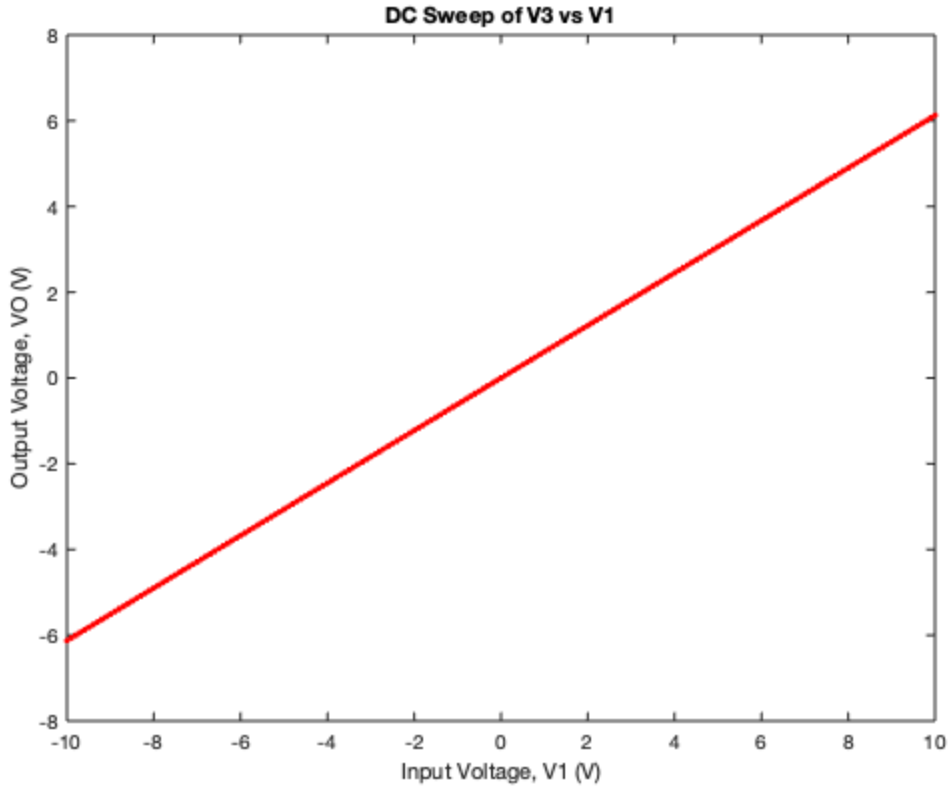
3 - MNPA Work

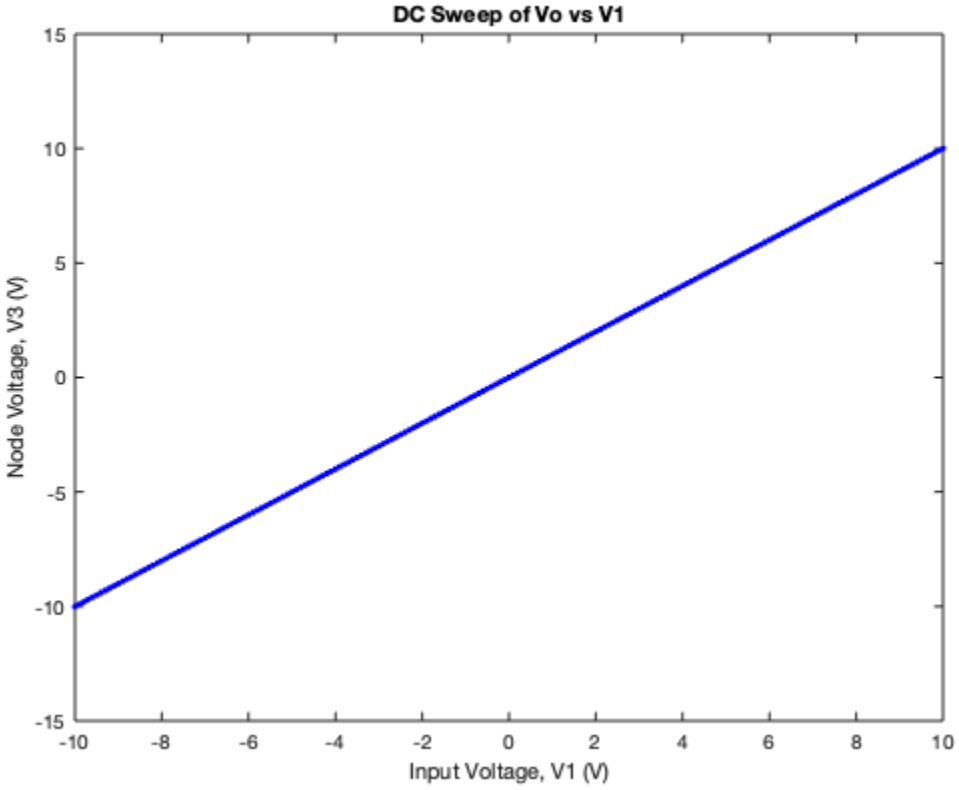
```
% Define Variables
w = 200;
s = 1i*w;
G1 = 1/1;
G2 = 1/2;
G3 = 1/10;
G4 = 1/0.1;
G0 = 1/1000;
c = 0.25;
L = 0.2;
```

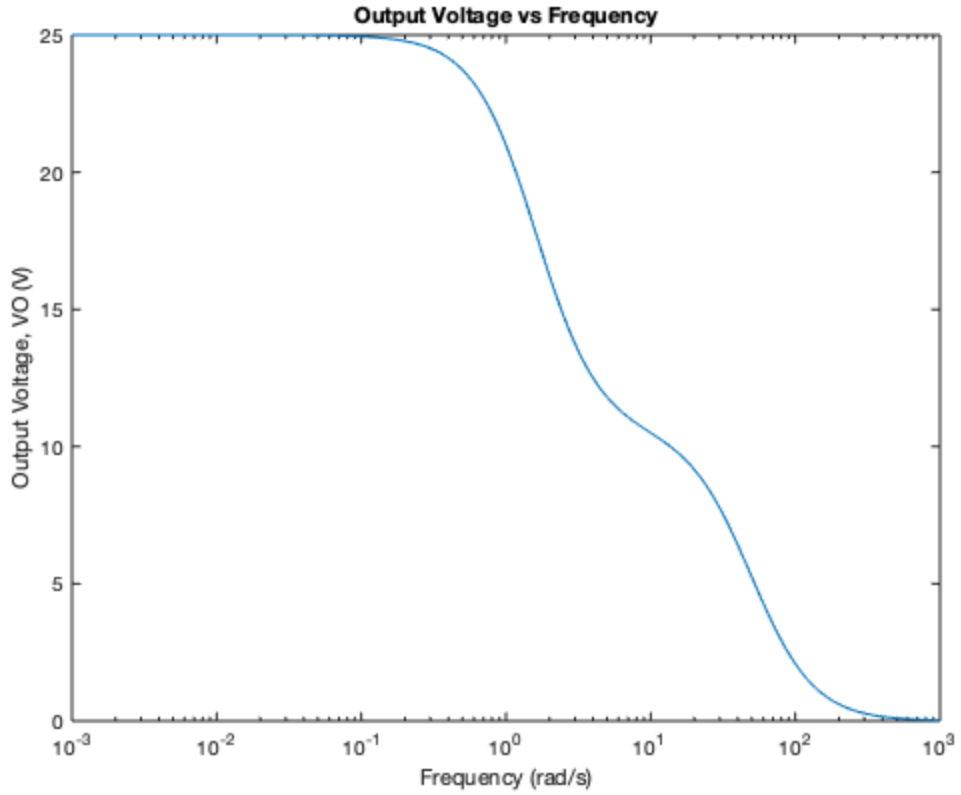
```
a = 100;

% Define Matrix

G = [0,  1,     0,               0,            0,    0,         0;
     1,  G1,    -G1,             0,            0,    0,         0;
     0,  -G1,   G1-G2+1/(s*L),   -1/(s*L),     0,    0,         0;
     0,  0,     -1/(s*L),        1/(s*L)-G3,   0,    0,         0;
     0,  0,     0,               0,            -1,   0,         a;
     0,  0,     0,               0,            G4,   -G4-G0,    0;
     0,  0,     0,               G3,           0,    0,         -1];

C = [0,  0,   0,   0,  0,  0,  0;
     0,  c,   -c,  0,  0,  0,  0;
     0,  -c,  c,   0,  0,  0,  0;
     0,  0,   0,   0,  0,  0,  0;
     0,  0,   0,   0,  0,  0,  0;
     0,  0,   0,   0,  0,  0,  0;
     0,  0,   0,   0,  0,  0,  0];

%DC sweep Vo
for Vin = -10:0.1:10

    F = [Vin;
       0;
       0;
       0;
       0;
       0;
       0];

    V = (G+(s.*C))\F;

    figure(2)
    plot(Vin, real(V(6)), 'r.')
    xlabel('Input Voltage, V1 (V)')
    ylabel('Output Voltage, VO (V)')
    title('DC Sweep of V3 vs V1')
    hold on
end

%DC sweep V3
for Vin = -10:0.1:10

    F = [Vin;
       0;
       0;
       0;
       0;
       0;
       0];

    V = (G+(s.*C))\F;
```

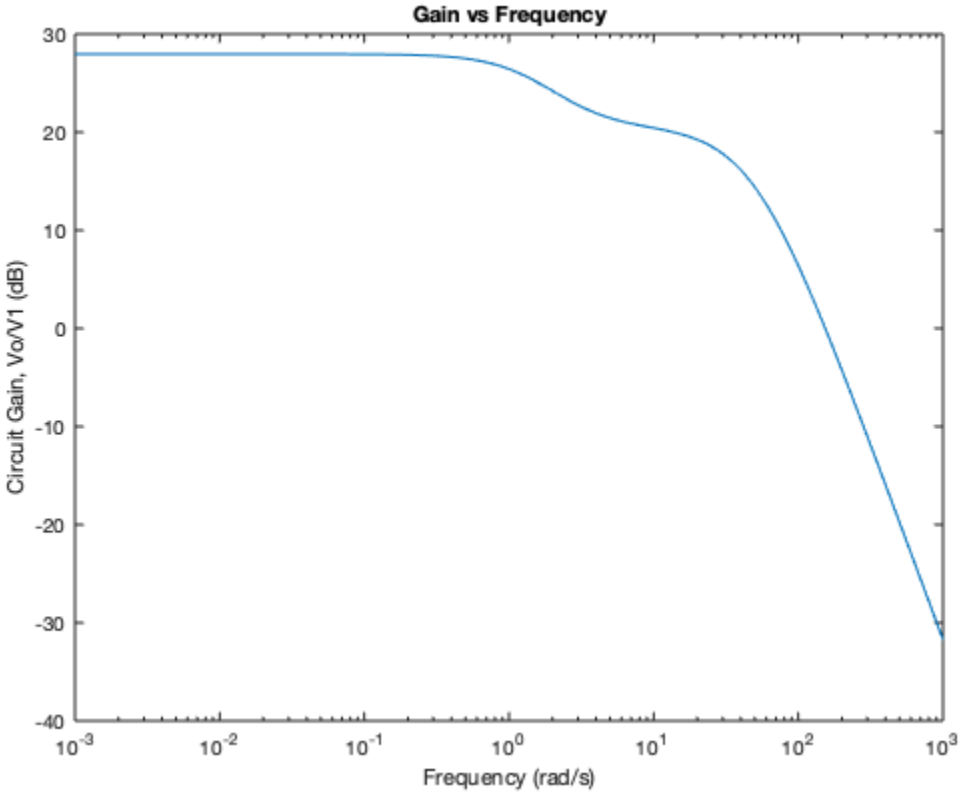
```
    figure(3)
    plot(Vin, real(V(3)), 'b.')
    xlabel('Input Voltage, V1 (V)')
    ylabel('Node Voltage, V3 (V)')
    title('DC Sweep of Vo vs V1')
    hold on
end



%AC voltage vs frequency
Vin = 1;
F = [Vin;
     0;
     0;
     0;
     0;
     0;
     0];
Vfreq = zeros(1,1000);
W = logspace(-3,3,1000);
for n = 1:1000

    s = 1i*W(n);

    G = [0,  1,     0,              0,           0,    0,        0;
         1,  G1,    -G1,            0,           0,    0,        0;
         0,  -G1,   G1-G2+1/(s*L),  -1/(s*L),    0,    0,        0;
         0,  0,     -1/(s*L),       1/(s*L)-G3,  0,    0,        0;
         0,  0,     0,              0,           -1,   0,        a;
         0,  0,     0,              0,           G4,   -G4-G0,   0;
         0,  0,     0,              G3,          0,    0,        -1];

    V = (G+(s.*C))\F;

    Vfreq(1,n) = real(V(6));

end

figure(4)
    semilogx(W,Vfreq)
    title('Output Voltage vs Frequency')
    xlabel('Frequency (rad/s)')
    ylabel('Output Voltage, VO (V)')

figure(5)
    semilogx(W,20*log10(Vfreq./Vin))
    title('Gain vs Frequency')
    xlabel('Frequency (rad/s)')
    ylabel('Circuit Gain, Vo/V1 (dB)')
```

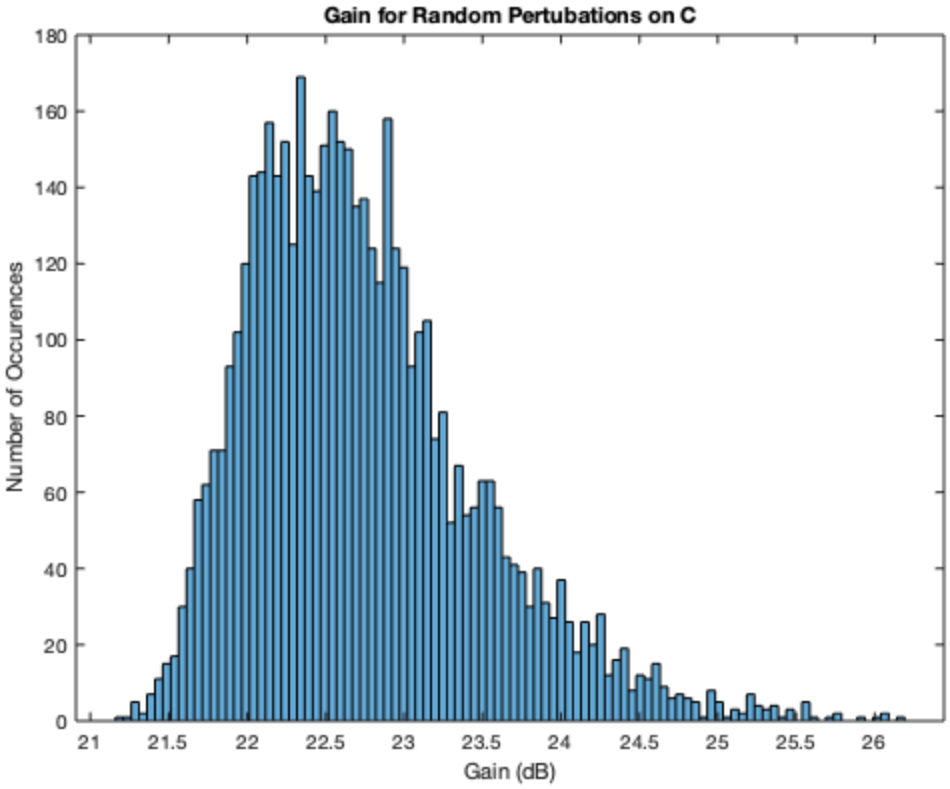
```
%AC Random Perturbations on C
c = 0.05.*randn(1,5000) + 0.25;
Vpert = zeros(1,5000);
s = 1i*pi;
G = [0,  1,     0,              0,           0,    0,        0;
     1,  G1,    -G1,            0,           0,    0,        0;
     0,  -G1,   G1-G2+1/(s*L),  -1/(s*L),    0,    0,        0;
     0,  0,     -1/(s*L),       1/(s*L)-G3,  0,    0,        0;
     0,  0,     0,              0,           -1,   0,        a;
     0,  0,     0,              0,           G4,   -G4-G0,   0;
     0,  0,     0,              G3,          0,    0,        -1];

for n = 1:5000

C = [0,  0,        0,        0,  0,  0,  0;
     0,  c(1,n),   -c(1,n),  0,  0,  0,  0;
     0,  -c(1,n),  c(1,n),   0,  0,  0,  0;
     0,  0,        0,        0,  0,  0,  0;
     0,  0,        0,        0,  0,  0,  0;
     0,  0,        0,        0,  0,  0,  0;
     0,  0,        0,        0,  0,  0,  0];

 V = (G+(s.*C))\F;
 Vpert(1,n) = 20*log10(real(V(6))/Vin);
end

figure(6)
histogram(Vpert,100)
title('Gain for Random Pertubations on C')
xlabel('Gain (dB)')
ylabel('Number of Occurences')
```


DC Sweep of V3 vs V1
Output Voltage, VO (V)
Input Voltage, V1 (V)
8
6
4
2
0
-2
-4
-6
-8
-10
-8
-6
-4
-2
0
2
4
6
8
10


DC Sweep of Vo vs V1
Node Voltage, V3 (V)
Input Voltage, V1 (V)
15
10
5
0
-5
-10
-15
-10
-8
-6
-4
-2
0
2
4
6
8
10


Output Voltage vs Frequency
Output Voltage, VO (V)
Frequency (rad/s)
25
20
15
10
5
0
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}


Gain vs Frequency
Circuit Gain, Vo/V1 (dB)
Frequency (rad/s)
30
20
10
0
-10
-20
-30
-40
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}



4 - Transient Circuit Simulation

a) This is a linear, time dependant, circuit with memory.

b) As the frequency of this circuit increases, we would expect the output voltage to decrease.

c) Starting with the reperesentation of this circuit in the time domine, we can derive the finite differance formula as shown below:

$$CdV/dt + GV = F$$

$$C(V(t) - V(t - \Delta t))/\Delta t + GV(t) = F(t)$$

$$V(t)C/\Delta t + V(t)G = CV(t - \Delta t)/\Delta t + F$$

Let $A = C/\Delta t + G$ ...

$$V(t)A = CV(t - \Delta t)/\Delta t + F$$

$$V(t) = A^{-1}[CV(t - \Delta t)/\Delta t + F]$$

d) This formula can now be implemented in order to simulate this cirucit as shown by the code below. Simulations for three different types of input wave can be seen. In each case, the input is reperesented by the blue curve, and the output by the red.

The frequency content of the input and output signals was then analysed. It could be seen that the frequency content in the input signals matched the frequency content on their corresponding output signals.

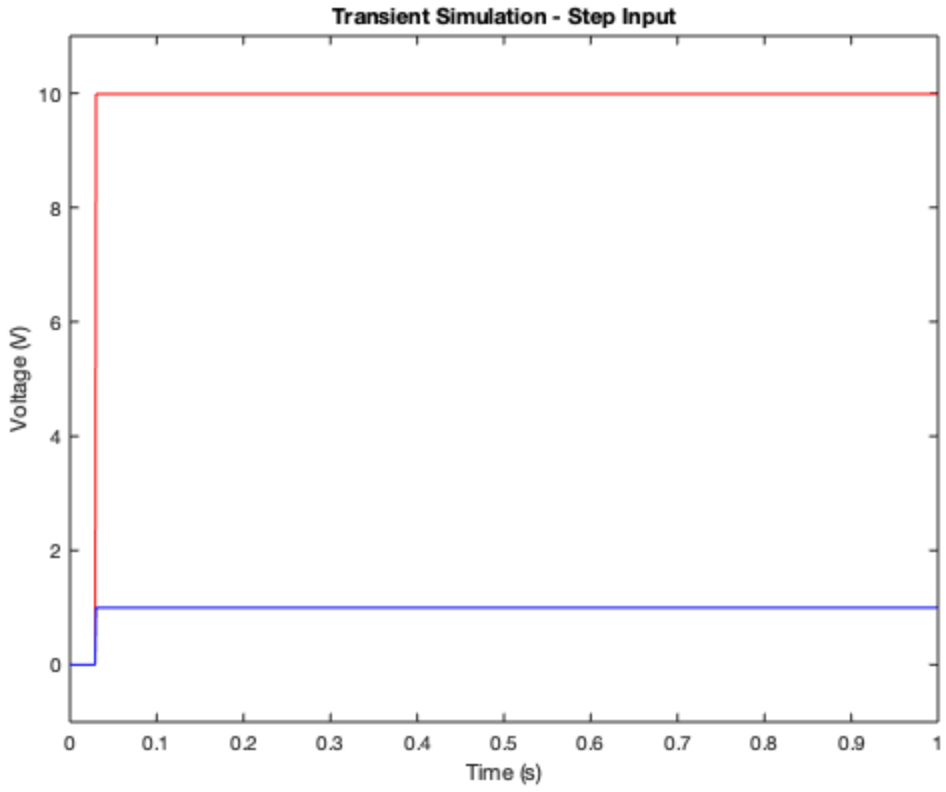
The process was then repeate with a time step increased to 100 steps per second. It was seen that the frequency content of the new time step seemed to be of a similar shape to before, but with the waveform spread out over a larger amount of time.

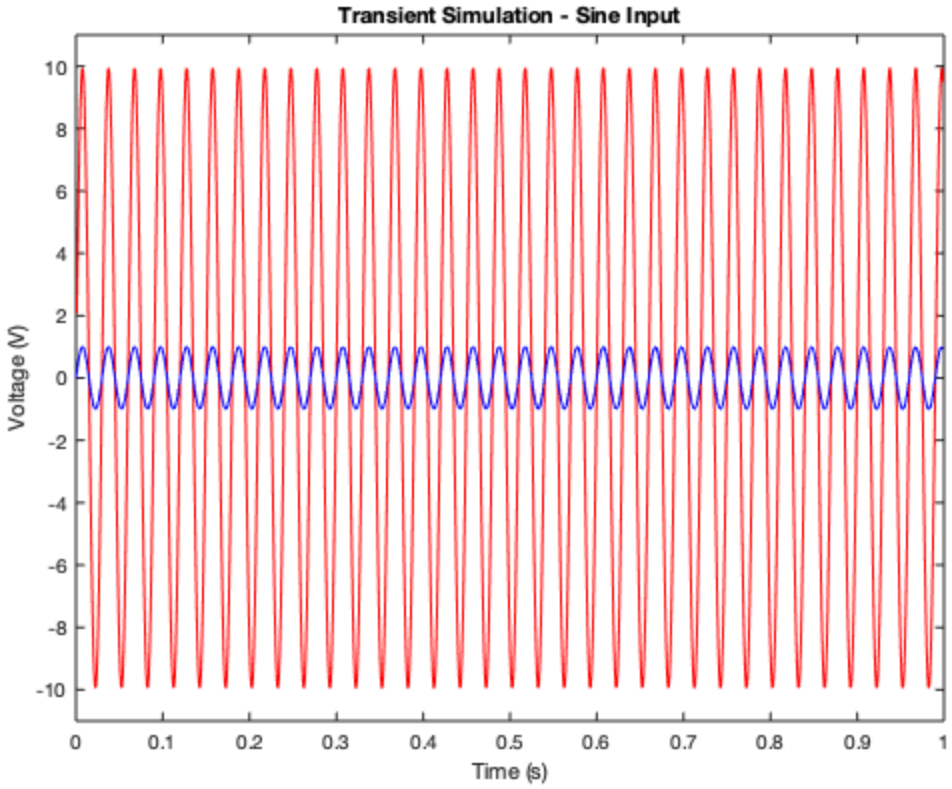
```
dt = 1e-3; %s
tmax = 1;  %s
nsteps = tmax/dt;
time = zeros(nsteps);

for j = 1:nsteps
    time(j)=(j*dt)-dt;
end

in_1 = zeros(nsteps);
for j = 1:nsteps
    if time(j)>=0.03
        in_1(j)=1;
    end
end

in_2 = zeros(nsteps);
f = 1/0.03;
for j = 1:nsteps
    in_2(j)=sin(2*pi*f*time(j));
end

in_3 = gaussmf(time,[0.03,0.06]);

c = 0.25;
C = [0,  0,        0,     0,  0,  0,  0;
     0,  c,        -c,    0,  0,  0,  0;
     0,  -c,       c,     0,  0,  0,  0;
     0,  0,        0,     0,  0,  0,  0;
     0,  0,        0,     0,  0,  0,  0;
     0,  0,        0,     0,  0,  0,  0;
     0,  0,        0,     0,  0,  0,  0];

A = C/dt + G;

F = [0;0;0;0;0;0;0];

V = 0;


%Simulation 1
out_1 = zeros(nsteps);
for j = 2:nsteps

    F(1) = in_1(j);
    V = A\(F+(C*out_1(j-1)/dt));
    out_1(j) = real(V(6));
```

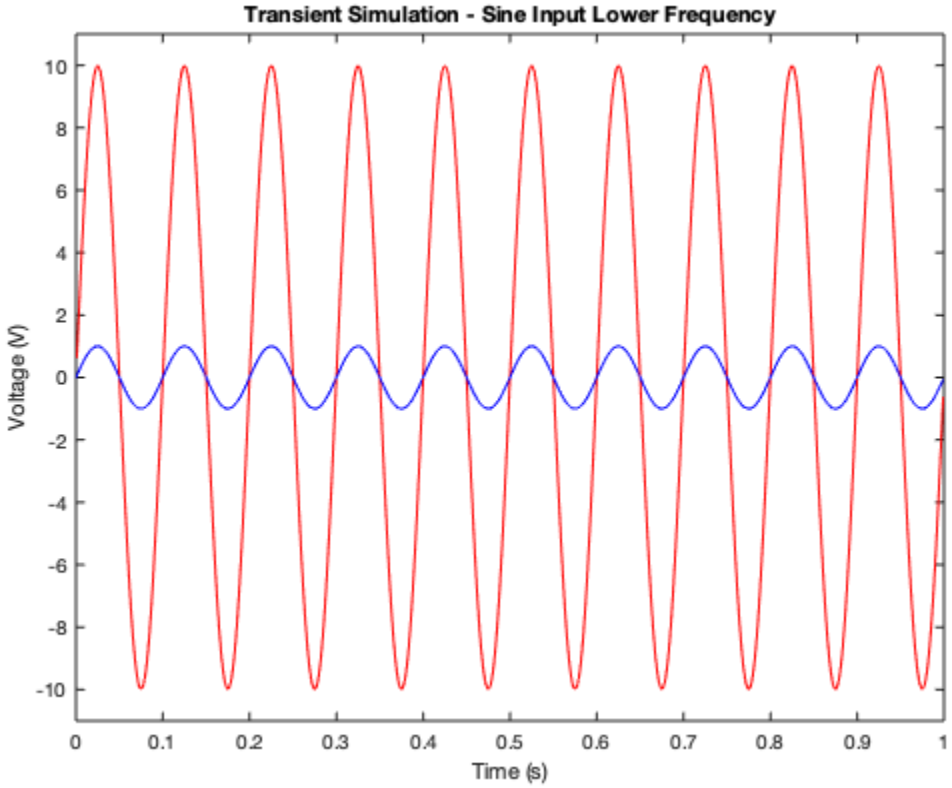
```
    figure(7)
    plot([time(j-1) time(j)],[out_1(j-1) out_1(j)],'r')
    plot([time(j-1) time(j)],[in_1(j-1) in_1(j)],'b')
    xlim([0 tmax])
    ylim([-1 11])
    title('Transient Simulation - Step Input')
    xlabel('Time (s)')
    ylabel('Voltage (V)')
    hold on

end


%Simulation 2
out_2 = zeros(nsteps);
for j = 2:nsteps

    F(1) = in_2(j);
    V = A\(F+(C*out_2(j-1)/dt));
    out_2(j) = real(V(6));

    figure(8)
    plot([time(j-1) time(j)],[out_2(j-1) out_2(j)],'r')
    plot([time(j-1) time(j)],[in_2(j-1) in_2(j)],'b')
    xlim([0 tmax])
    ylim([-11 11])
    title('Transient Simulation - Sine Input')
    xlabel('Time (s)')
    ylabel('Voltage (V)')
    hold on

end

%Simulation 2.B
in_2 = zeros(nsteps);
f = 1/0.1;
for j = 1:nsteps
    in_2(j)=sin(2*pi*f*time(j));
end
out_2 = zeros(nsteps);
for j = 2:nsteps

    F(1) = in_2(j);
    V = A\(F+(C*out_2(j-1)/dt));
    out_2(j) = real(V(6));

    figure(9)
    plot([time(j-1) time(j)],[out_2(j-1) out_2(j)],'r')
    plot([time(j-1) time(j)],[in_2(j-1) in_2(j)],'b')
    xlim([0 tmax])
    ylim([-11 11])
    title('Transient Simulation - Sine Input Lower Frequency')
    xlabel('Time (s)')
```

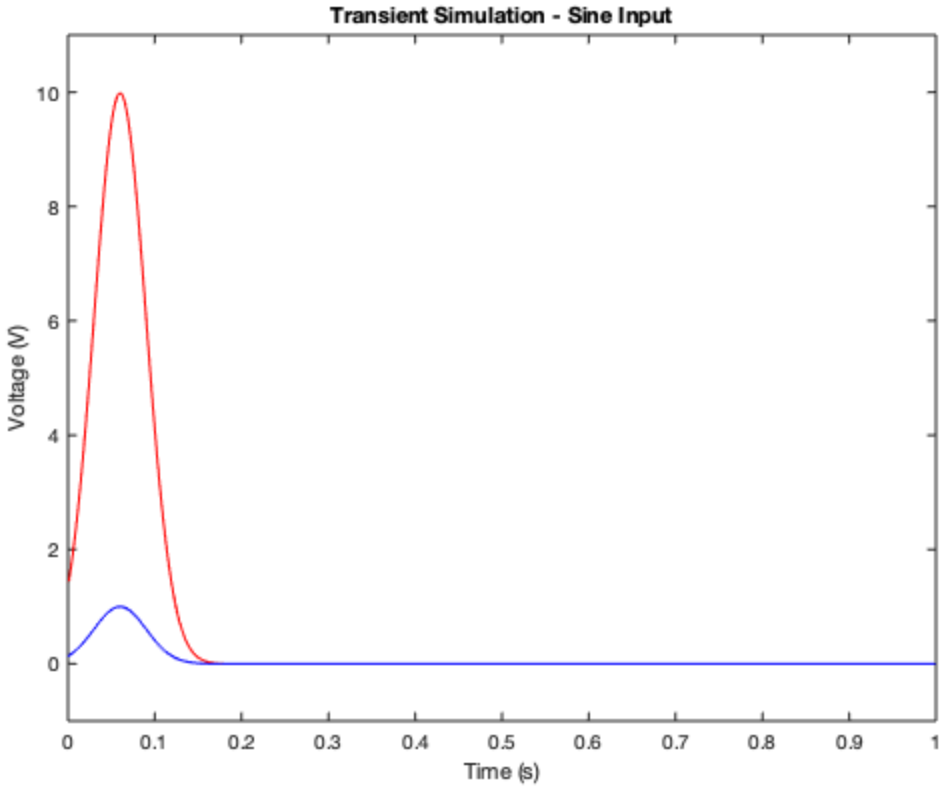
```
    ylabel('Voltage (V)')
    hold on

end

%Simulation 3
out_3 = zeros(nsteps);
for j = 2:nsteps

    F(1) = in_3(j);
    V = A\(F+(C*out_3(j-1)/dt));
    out_3(j) = real(V(6));

    figure(10)
    plot([time(j-1) time(j)],[out_3(j-1) out_3(j)],'r')
    plot([time(j-1) time(j)],[in_3(j-1) in_3(j)],'b')
    xlim([0 tmax])
    ylim([-1 11])
    title('Transient Simulation - Sine Input')
    xlabel('Time (s)')
    ylabel('Voltage (V)')
    hold on

end


fft_in_1 = fft(in_1(:,1));
fft_in_1 = fftshift(fft_in_1);
fft_out_1 = fft(out_1(:,1));
fft_out_1 = fftshift(fft_out_1);

figure(11)
plot(time,real(fft_in_1))
title('Frequency Content for Step Input')

figure(12)
plot(time,real(fft_out_1))
title('Frequency Content for Step Output')

fft_in_2 = fft(in_2(:,1));
fft_in_2 = fftshift(fft_in_2);
fft_out_2 = fft(out_2(:,1));
fft_out_2 = fftshift(fft_out_2);

figure(13)
plot(time,real(fft_in_2))
title('Frequency Content for Sine Input')

figure(14)
plot(time,real(fft_out_2))
title('Frequency Content for Sine Output')


fft_in_3 = fft(in_3(:,1));
```

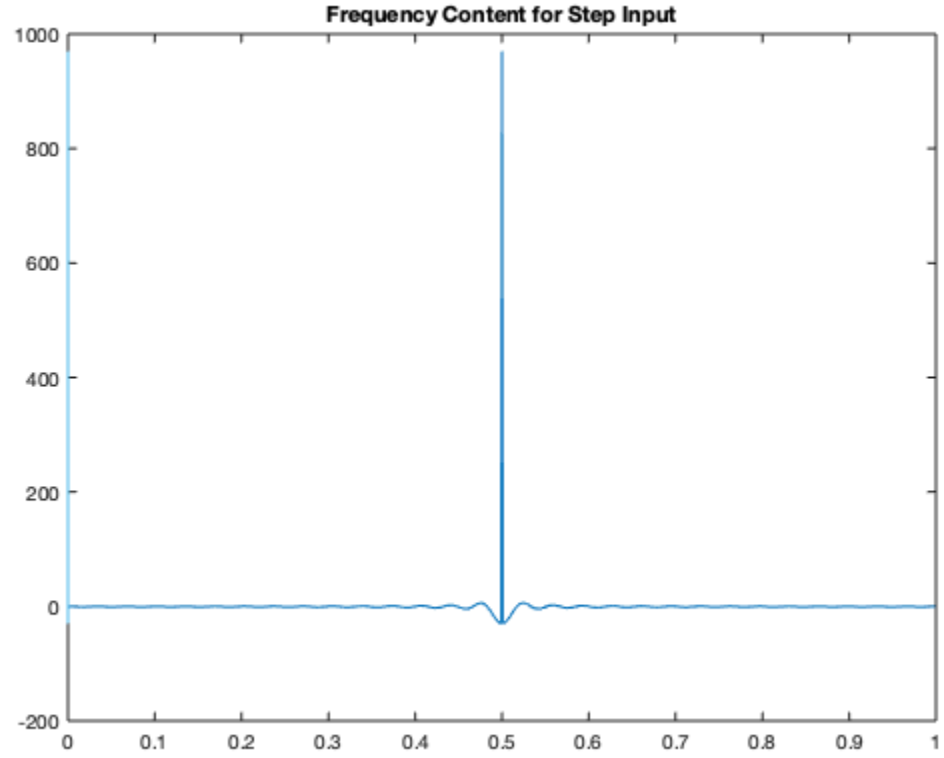
```
fft_in_3 = fftshift(fft_in_3);
fft_out_3 = fft(out_3(:,1));
fft_out_3 = fftshift(fft_out_3);

figure(15)
plot(time,real(fft_in_3))
title('Frequency Content for Gaussian Input')

figure(16)
plot(time,real(fft_out_3))
title('Frequency Content for Gaussian Output')
```

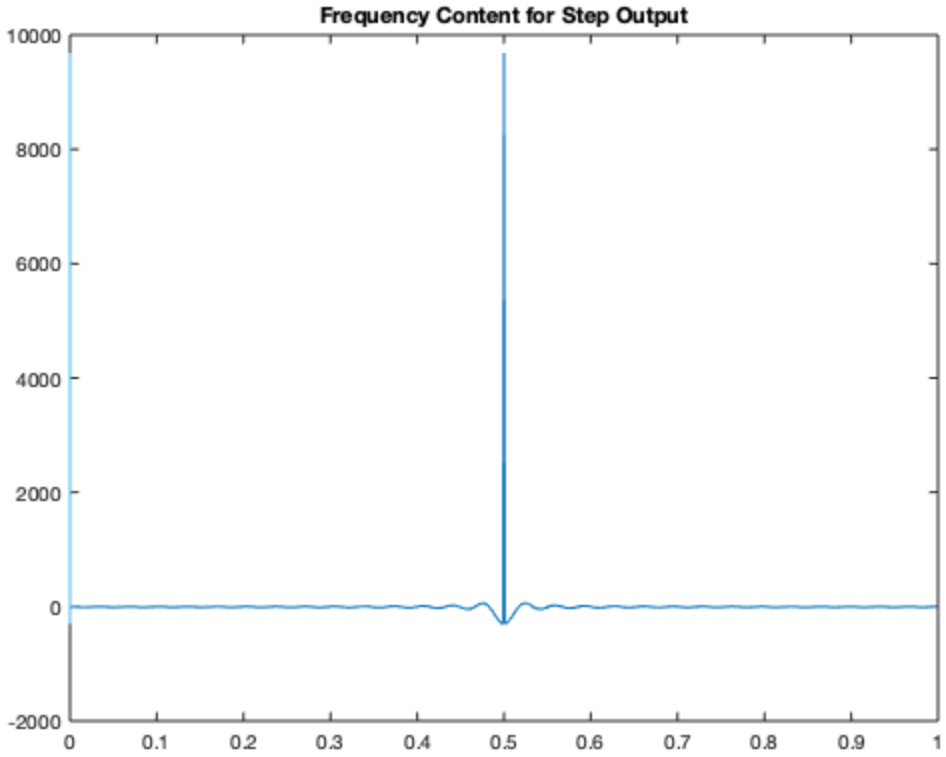


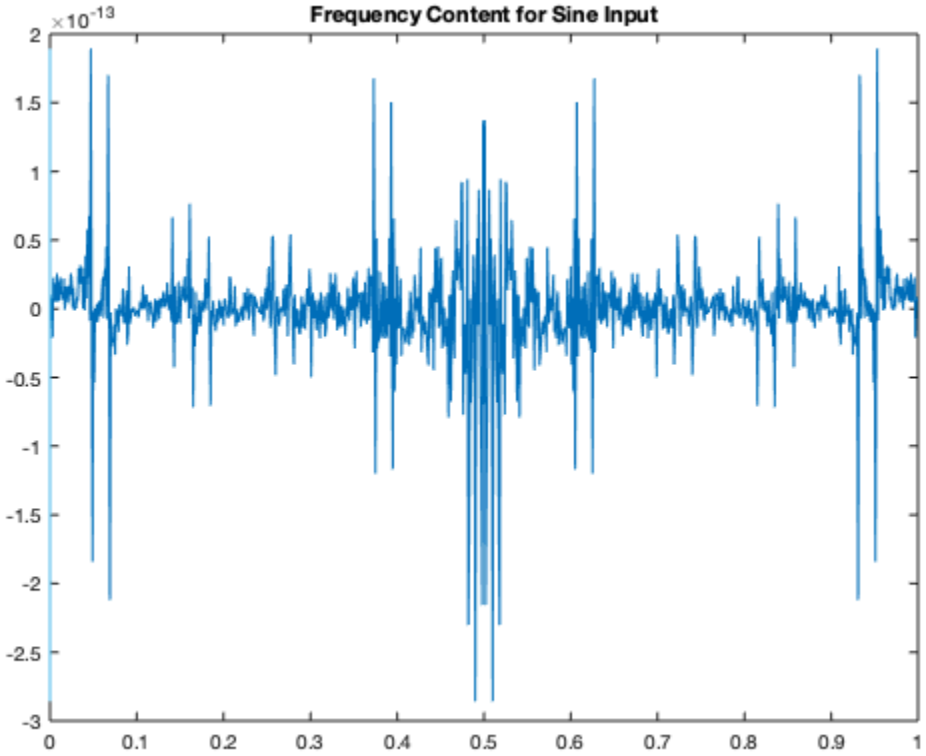

Transient Simulation - Sine Input
Voltage (V)
Time (s)
10
8
6
4
2
0
-2
-4
-6
-8
-10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

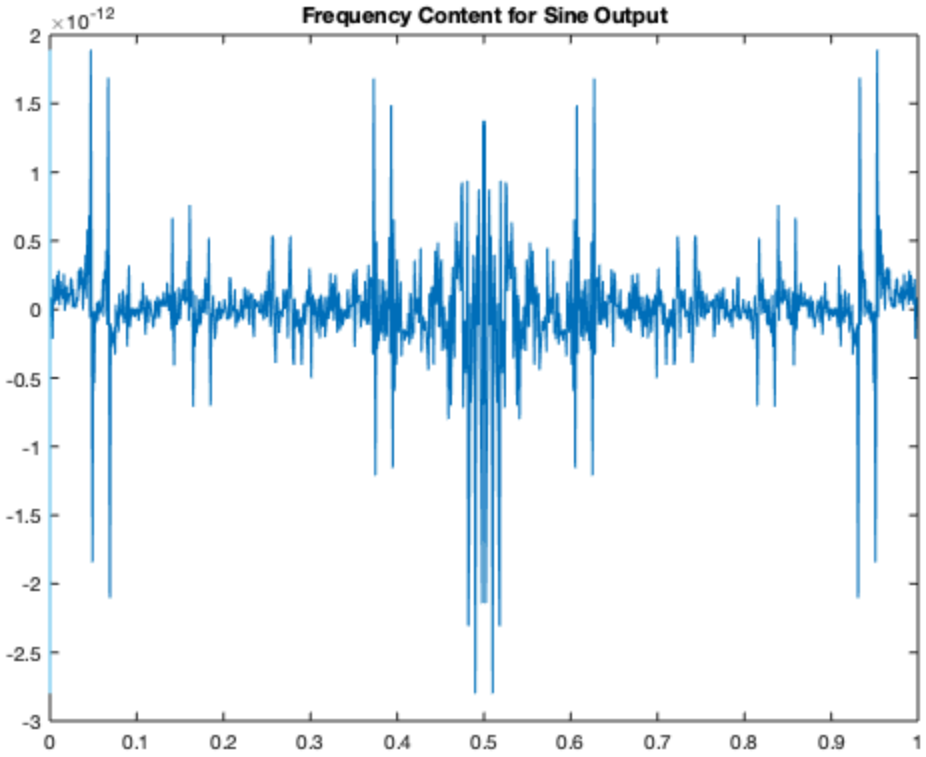

Transient Simulation - Sine Input Lower Frequency
Voltage (V)
Time (s)
10
8
6
4
2
0
-2
-4
-6
-8
-10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

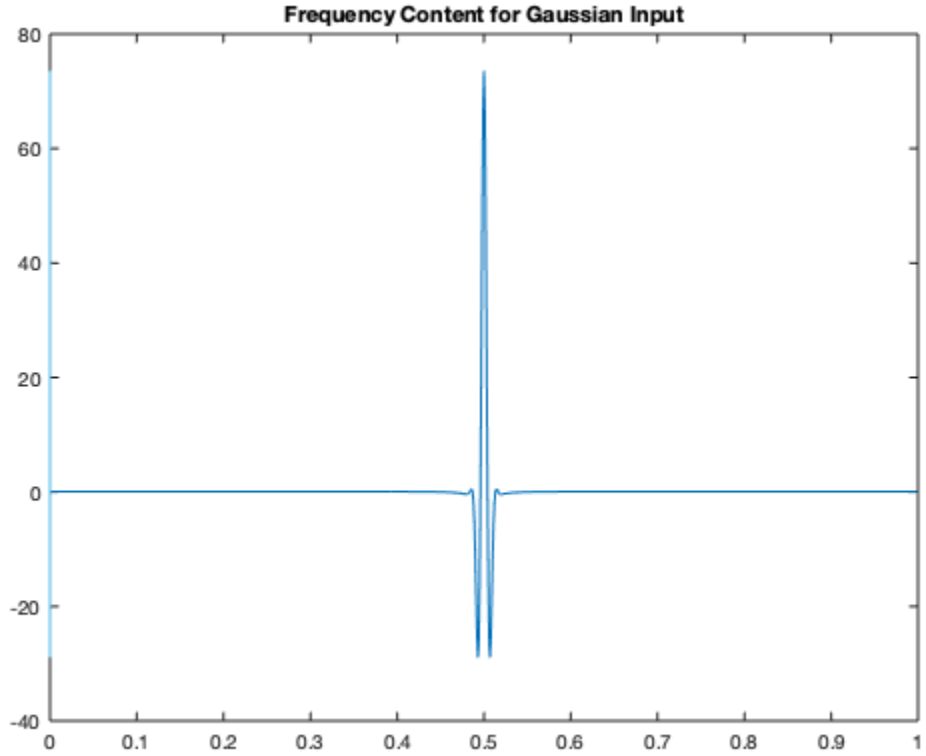

Transient Simulation - Sine Input
Voltage (V)
Time (s)
10
8
6
4
2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

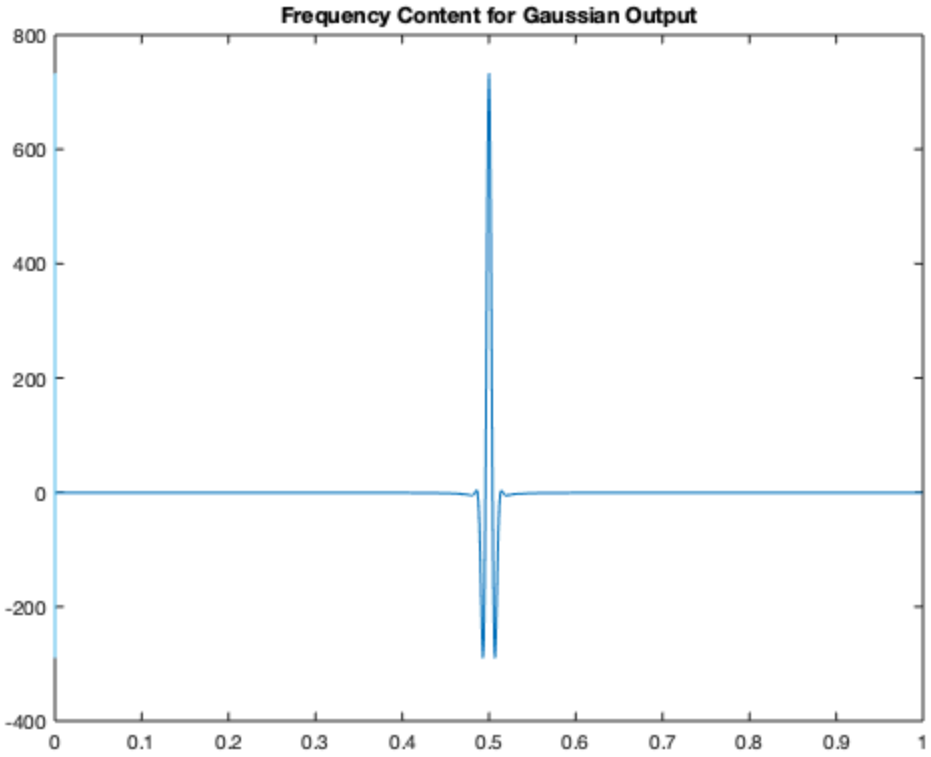

Frequency Content for Step Input
1000
800
600
400
200
0
-200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Content for Step Output
10000
8000
6000
4000
2000
0
-2000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Content for Sine Input
×10^-13
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Content for Sine Output
×10^-12
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Content for Gaussian Input
80
60
40
20
0
-20
-40
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



5 - Circuit with Noise

We could now introduce noise into the system in the form of a current source and a bandwidth limiting capacitor. The simulation of this new system can be seen below.

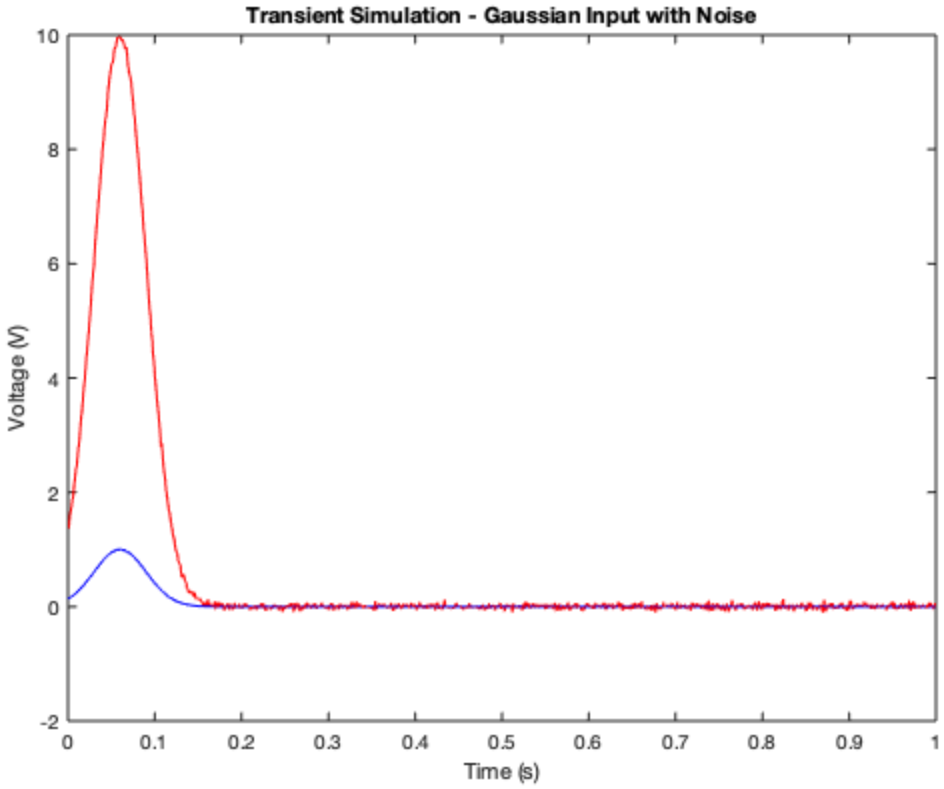
By compairing the frequency responce for the system woth a capacitor Cn of values 0.00001, and 0.0001, it can be seen that increasing the value of the capacitance also increases the effect of noise on the circuits frequency response.

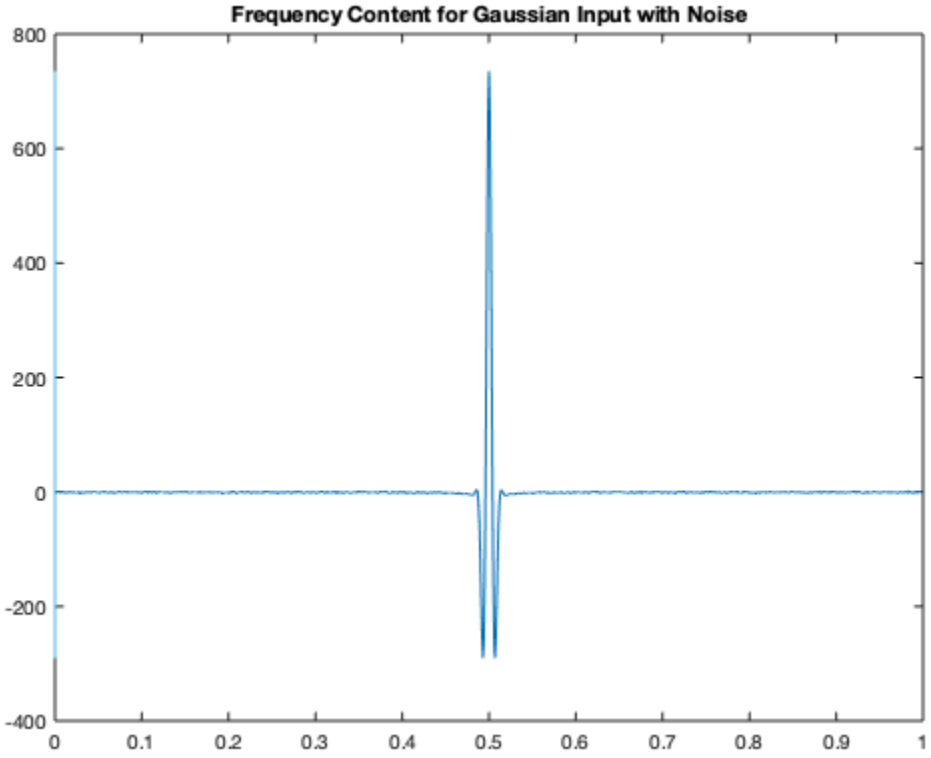
As before, increasing the time step caused the frequency responce curve of the system to become more spread out over a larger range of frequencies.

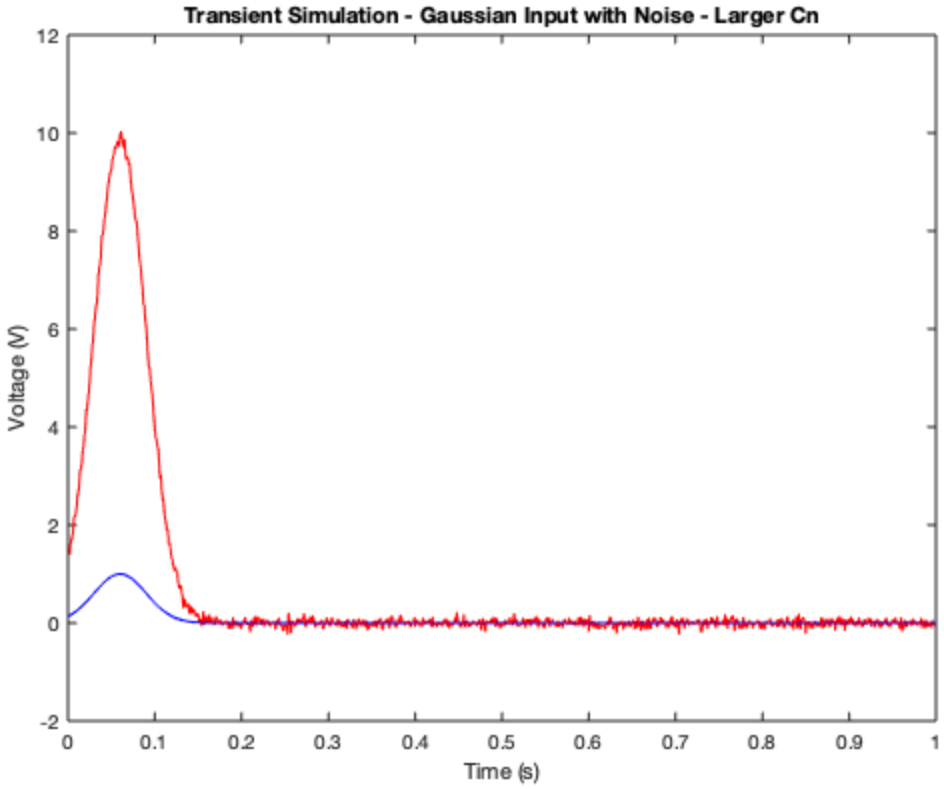
```
% Define new C matrix
cn = 0.00001
C = [0,  0,   0,   0,    0,  0,  0;
     0,  c,   -c,  0,    0,  0,  0;
     0,  -c,  c,   0,    0,  0,  0;
     0,  0,   0,   -cn,  0,  0,  0;
     0,  0,   0,   0,    0,  0,  0;
     0,  0,   0,   0,    0,  0,  0;
     0,  0,   0,   0,    0,  0,  0];


in_N = gaussmf(time,[0.03,0.06]);

A = C/dt + G;
```

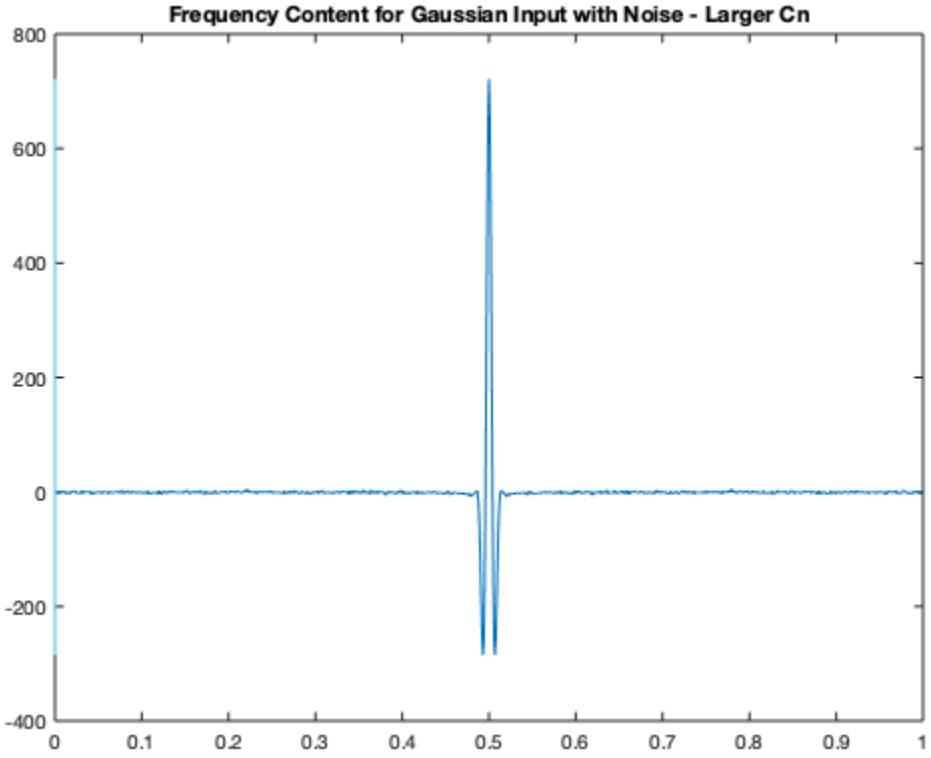
```
In = 0.1*randn(nsteps,1);

F = [0;0;0;0;0;0;0];

V = 0;

out_N = zeros(nsteps);
for j = 2:nsteps

    F(1) = in_N(j);
    F(4) = In(j);
    V = A\(F+(C*out_N(j-1)/dt));
    out_N(j) = real(V(6));

    figure(17)
    plot([time(j-1) time(j)],[out_N(j-1) out_N(j)],'r')
    plot([time(j-1) time(j)],[in_N(j-1) in_N(j)],'b')
    xlim([0 tmax])
    %ylim([-1 11])
    title('Transient Simulation - Gaussian Input with Noise')
    xlabel('Time (s)')
    ylabel('Voltage (V)')
    hold on

end


fft_out_N = fft(out_N(:,1));
fft_out_N = fftshift(fft_out_N);

figure(18)
plot(time,real(fft_out_N))
title('Frequency Content for Gaussian Input with Noise')


%Repeat for higher value of Cn
cn = 0.0001
C = [0,  0,   0,   0,    0,  0,  0;
     0,  c,   -c,  0,    0,  0,  0;
     0,  -c,  c,   0,    0,  0,  0;
     0,  0,   0,   -cn,  0,  0,  0;
     0,  0,   0,   0,    0,  0,  0;
     0,  0,   0,   0,    0,  0,  0;
     0,  0,   0,   0,    0,  0,  0];


in_N2 = gaussmf(time,[0.03,0.06]);

A = C/dt + G;

In = 0.1.*randn(nsteps,1);

F = [0;0;0;0;0;0;0];
```

```
V = 0;

out_N2 = zeros(nsteps);
for j = 2:nsteps

    F(1) = in_N2(j);
    F(4) = In(j);
    V = A\(F+(C*out_N2(j-1)/dt));
    out_N2(j) = real(V(6));

    figure(19)
    plot([time(j-1) time(j)],[out_N2(j-1) out_N2(j)],'r')
    plot([time(j-1) time(j)],[in_N2(j-1) in_N2(j)],'b')
    xlim([0 tmax])
    %ylim([-1 11])
    title('Transient Simulation - Gaussian Input with Noise - Larger
 Cn')
    xlabel('Time (s)')
    ylabel('Voltage (V)')
    hold on

end


fft_out_N2 = fft(out_N2(:,1));
fft_out_N2 = fftshift(fft_out_N2);

figure(20)
plot(time,real(fft_out_N2))
title('Frequency Content for Gaussian Input with Noise - Larger Cn')


cn =

   1.0000e-05


cn =

   1.0000e-04
```


Transient Simulation - Gaussian Input with Noise
Voltage (V)
Time (s)
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 for Gaussian Input with Noise
800
600
400
200
0
-200
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Transient Simulation - Gaussian Input with Noise - Larger Cn
Voltage (V)
Time (s)
12
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 for Gaussian Input with Noise - Larger Cn
800
600
400
200
0
-200
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Published with MATLAB® R2019b