Carleton University

Modelling of Integrated Devices

ELEC 4700

Assignment 4 - Circuit Modeling

Author: Maharshi Gurjar
Student number: 101094330
Date submitted: April 11, 2021

Contents

List of Figures

List of Source Codes

1 Noiseless Circuit Modeling

1.1 DC Sweep

Using the Modified Nodal Analysis stamps from ELEC 4506, this portion of the assignment expands on the nodal analysis PA (MNPA) to solve,

$$\mathbf{C}\frac{d\mathbf{V}}{dt} + \mathbf{G}\mathbf{V} = F \tag{1}$$

$$(\mathbf{G} + j\omega\mathbf{C})V = F(\omega) \tag{2}$$

Where the $\mathbf{G}, \mathbf{C}$, and $\mathbf{F}$ matrix that were created by the stamps are,

$$\mathbf{G} = \begin{bmatrix} 1.0000 & -1.0000 & 0 & 0 & 0 & 1.0000 & 0 & 0 \\ -1.0000 & 1.5000 & 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0.0054 & 0 & 0 & 0 & -1.0000 & 0 \\ 0 & 0 & 0 & 10.0000 & -10.0000 & 0 & 0 & 1.0000 \\ 0 & 0 & 0 & -10.0000 & 10.0010 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & -1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5376 & 1.0000 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0.2500 & -0.2500 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2500 & 0.2500 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.2000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Using these MNA equations matrix a DC Sweep simulation was conducted. By obtaining the voltage as,

$$V = G \backslash F \tag{3}$$

Where the '\' is the MATLAB operator to do Matrix left division operation. Sweeping the input voltage from (-10 to 10)V, and obtaining the outputs at Nodes 3 and 5, where Node 5 is the output voltage node the are obtained as shown in Figure 1 on the following page.

1.2 AC Sweep

Next the same circuit was simulated again to obtain the frequency response of the circuit. Using the same nodal stamps and equations, the circuit was swept from 1Hz to 100kHz. The results of this simulation show the response at Node 5, both in Voltage and dB as shown in Figure 2 on the following page. The results of the simulation show a 'pass-band' like filter response from this circuit, where outside of the pass band range (approx 10hz to 10kHz) the signal is attenuated.

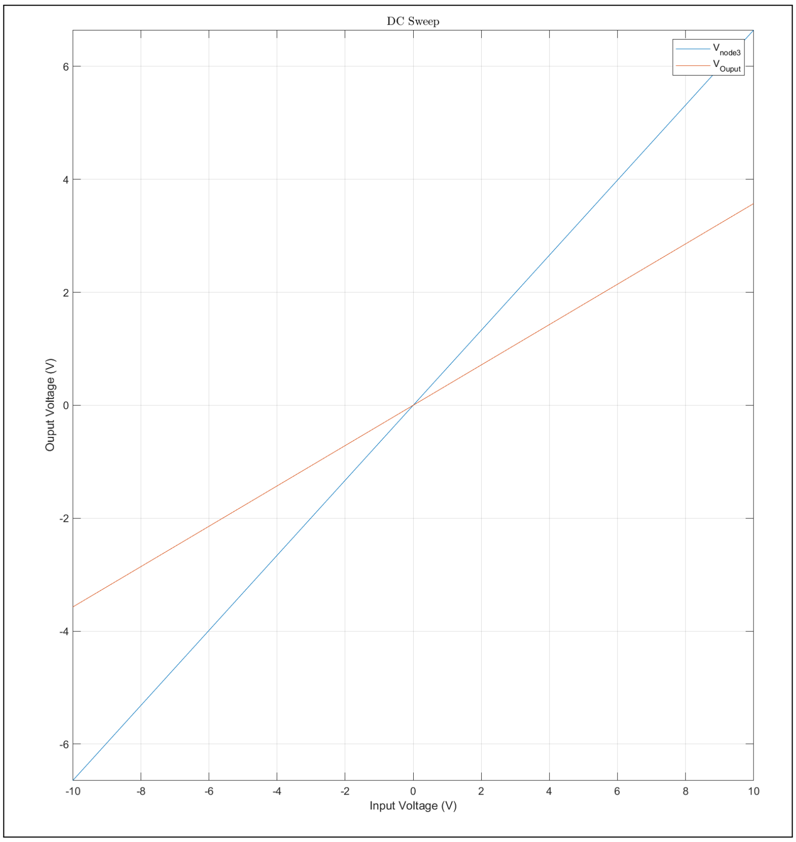

Figure 1: DC Sweep Simulation Results

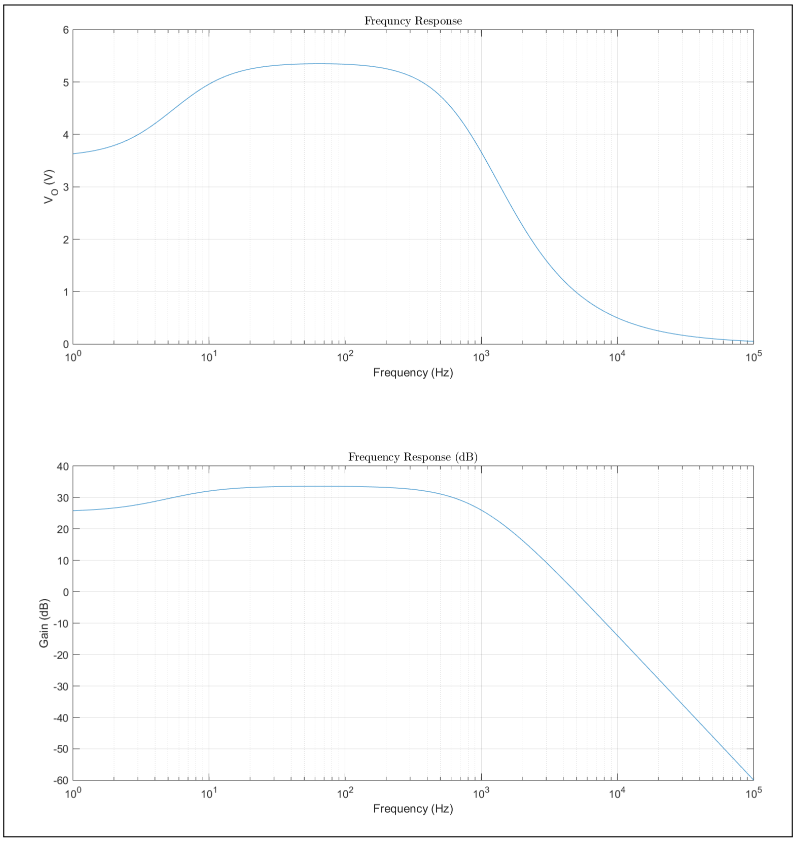

Figure 2: Frequency Sweep Simulation Results

1.3 Random Perturbations of Capacitance

This section of the simulations is centred around exploring the effects of random variances in the manufacturing process can effect the circuit response. This simulation is done to show the effects that a variance in the capacitance value can have on the overall gain of a circuit. A normal distribution ($\sigma = 5$) was created to vary the capacitance value, and the gain was re-simulated under those conditions. The results of this simulation are as shown in Figure 3 below.

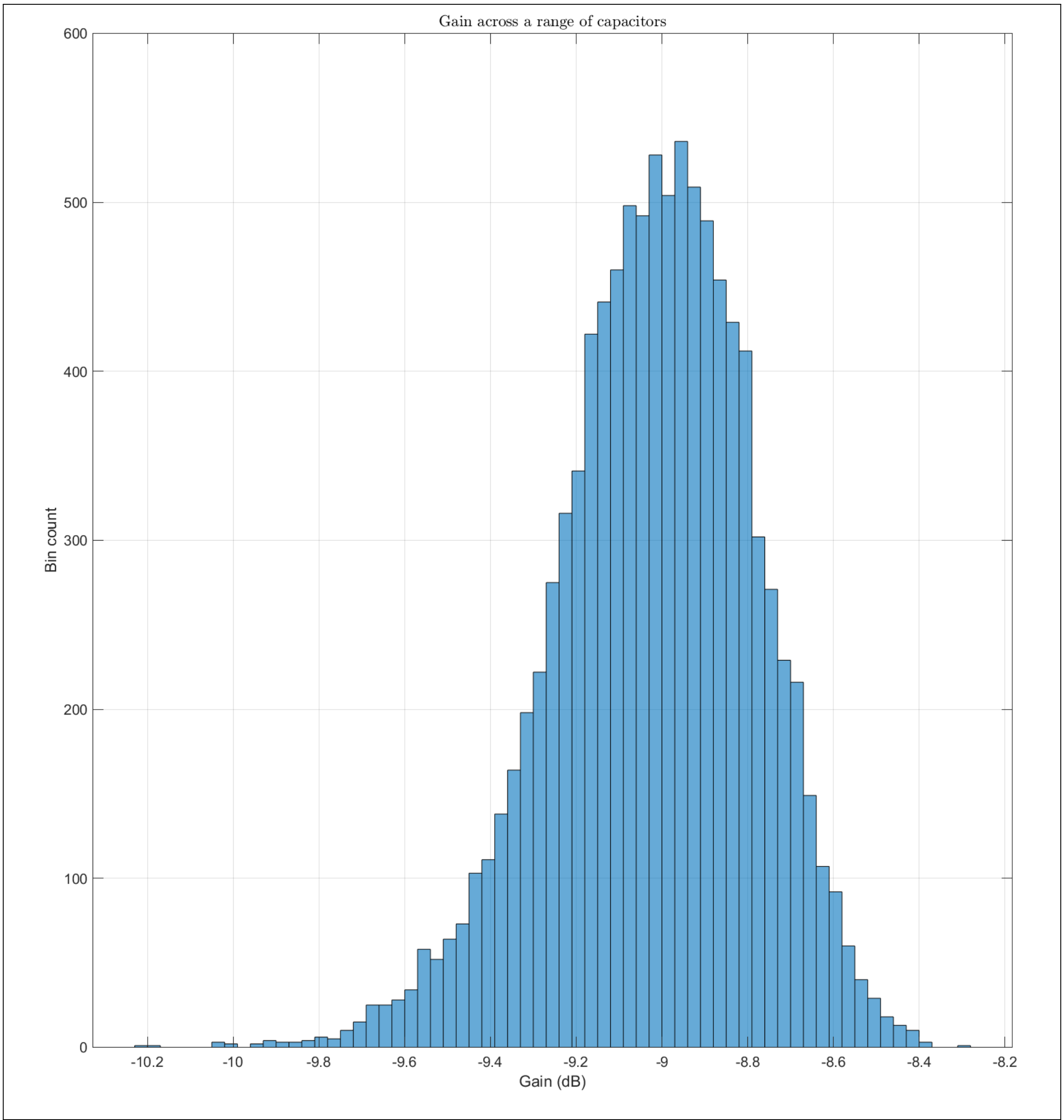

Figure 3: Random C Perturbations Simulation Results

1.4 Transient and Frequency Spectrum Simulation

This portion of the assignment simulates the transient circuit response using three different types of inputs; step, sin and pulse.

Step Input Figure 4 below shows the results of the transient simulation, as well as the frequency spectrum result of a step input signal.

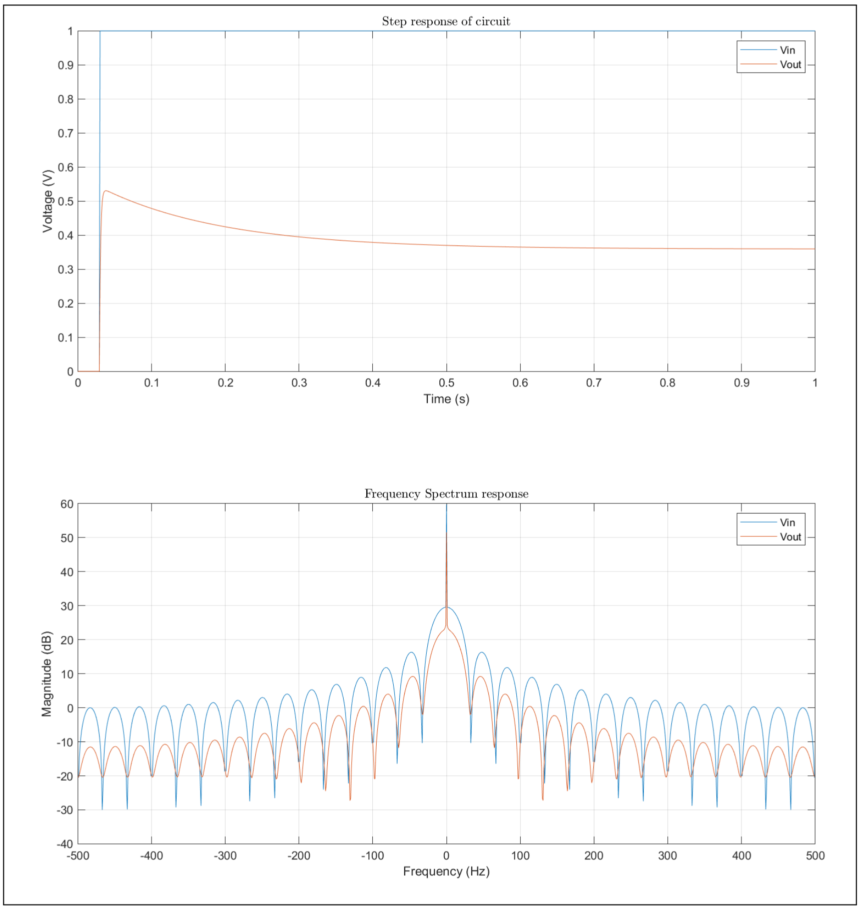

Figure 4: Step Input Simulation Results

Sin Input Figure 5 below shows the results of the transient simulation, as well as the frequency spectrum result of a Sinusoidal input signal.

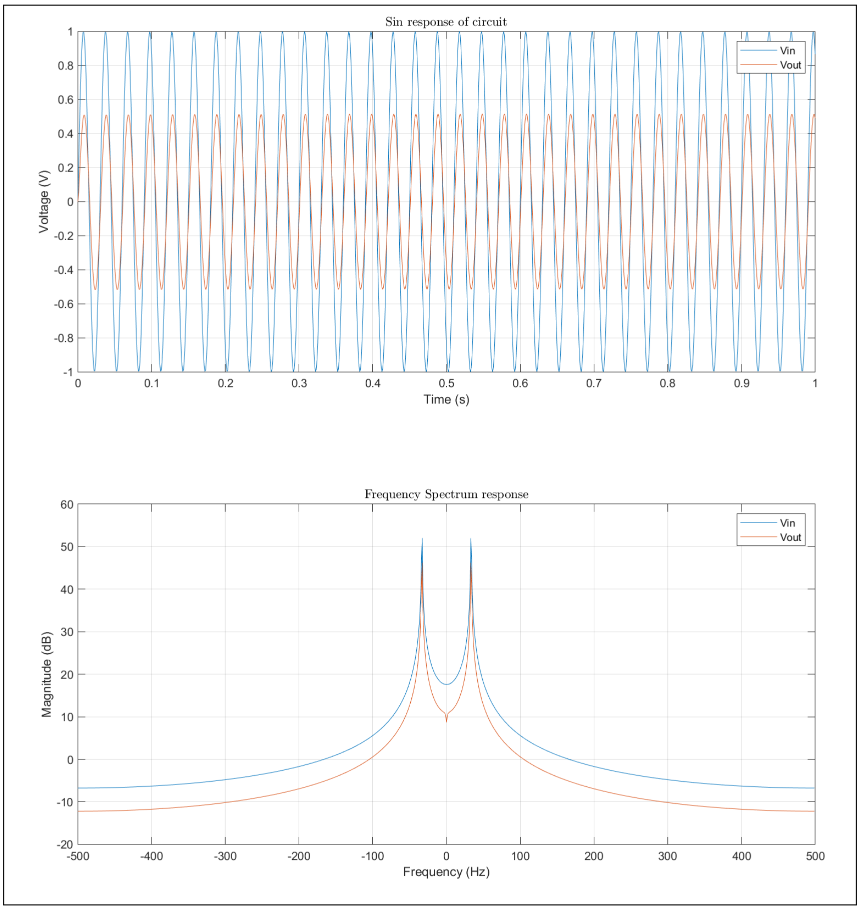

Figure 5: Sin Input Simulation Results

Sin Input Figure 6 below shows the results of the transient simulation, as well as the frequency spectrum result of a Sinusoidal input signal at a lower frequency than of 5.

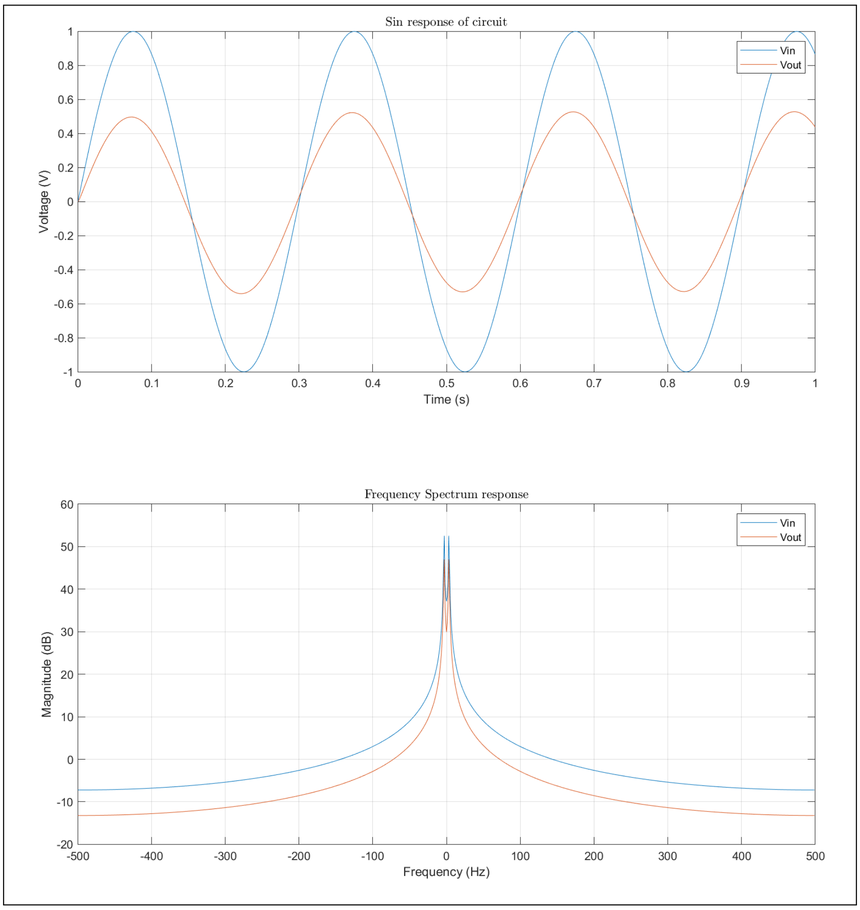

Figure 6: Sin Input (Different Frequency) Simulation Results

Step Input Figure 7 below shows the results of the transient simulation, as well as the frequency spectrum result of a impulse input signal.

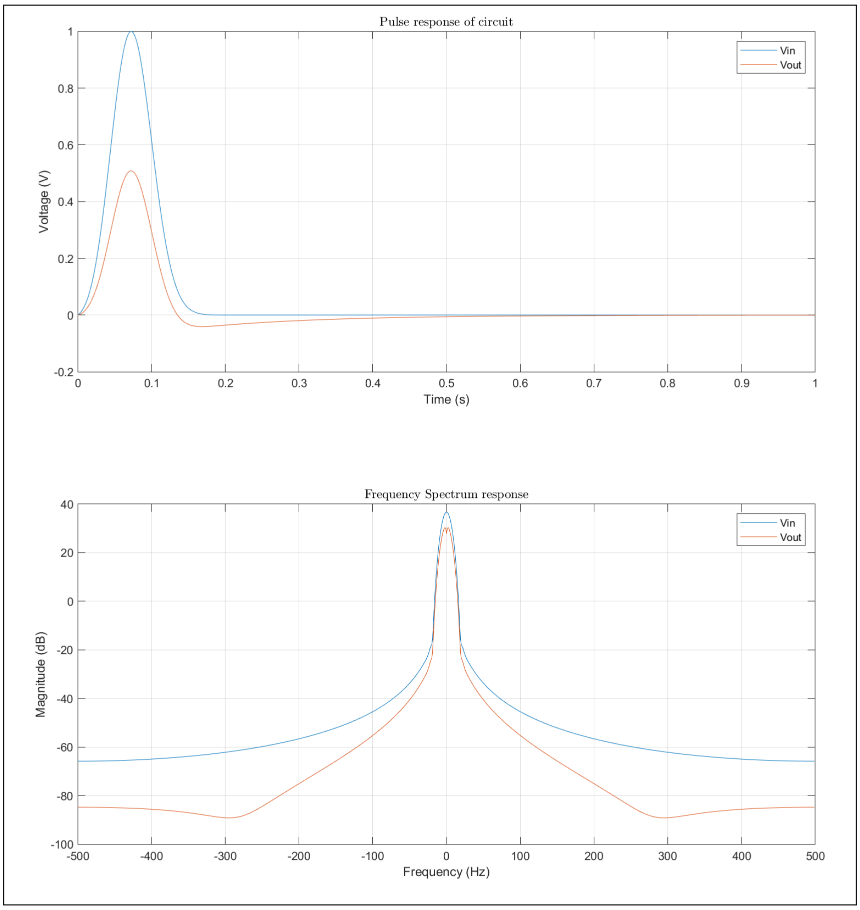

Figure 7: Impulse Input Simulation Results

2 Noisy Circuit Modelling

This portion of the assignment aims to model the effects of noise in electrical circuits, and explore the effects it has on the output signal. This section also explores the effects of carrying the capacitance and time steps.

2.1 Noisy Signal

Noise is simulated using a current source to in parallel with the resistance to model random thermal noise, where $I_n = 0.0001$ to scale the normally distributed random numbers. Each of the following simulations were conducted using the Impulse signal, as shown by Figure 8 below. The frequency spectrum results shows that the output is only slightly above the noise floor, and the overall effect of noise with respect to signal degeneration.

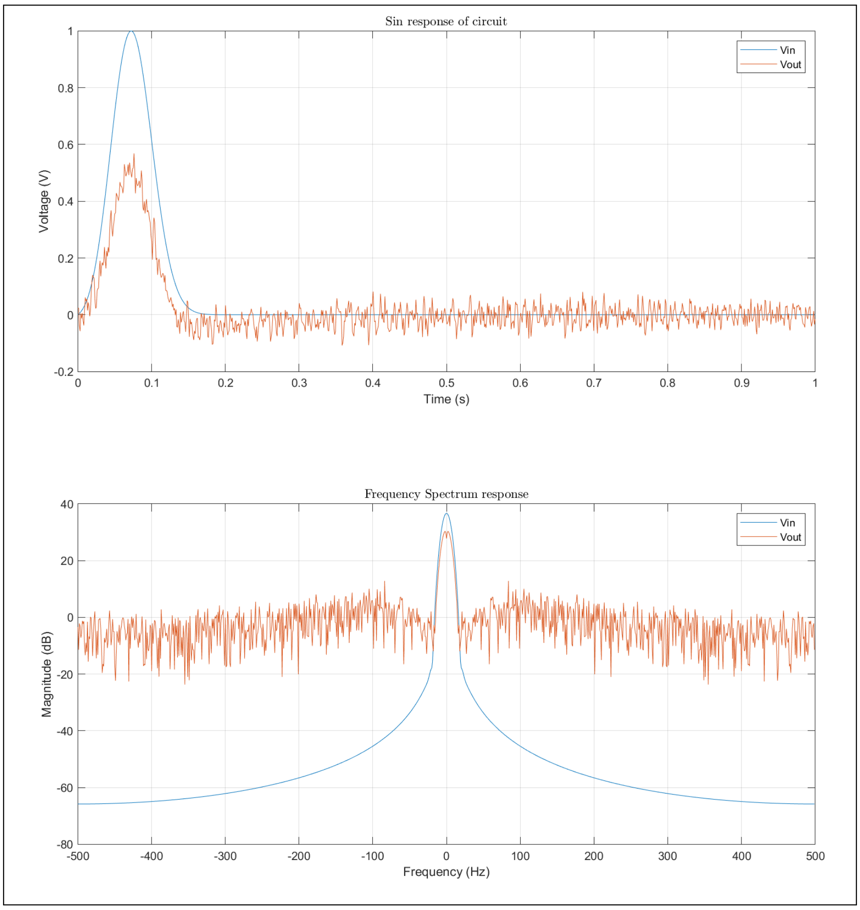

Figure 8: Impulse Input With Noise Transient Simulation Results

2.2 Varying C_n

The results of simulating the circuit over a range of C_n shows that the capacitance limits the bandwidth (wherein the higher frequency noise is attenuated). Each of the C_n cases provides some insight to the effects, as discussed below. The final take away from these simulations is that as C_n increases the bandwidth decreases.

- $C_n = 0.00001$ - Due to the small capacitance value, the noisy signal is passed through, hence the noise is easily seen in both the time and frequency domain.
- $C_n = 0.00010$ - Here compared to the case above, less of the noise is passed through, while some amount is still exists.
- $C_n = 0.00100$ - At this capacitance value the the higher frequencies pass through, allowing the Gaussian of the input signal to be seen at the output.
- $C_n = 0.01000$ - At this capacitance value the frequencies are passed through, but not fully attenuated as it still blocks some of the higher frequencies.
- $C_n = 0.10000$ - This case has the largest capacitance value, and shows that the effects of this is that the higher frequencies are blocked, as shown by the sign signal passed at the output, but attenuated.

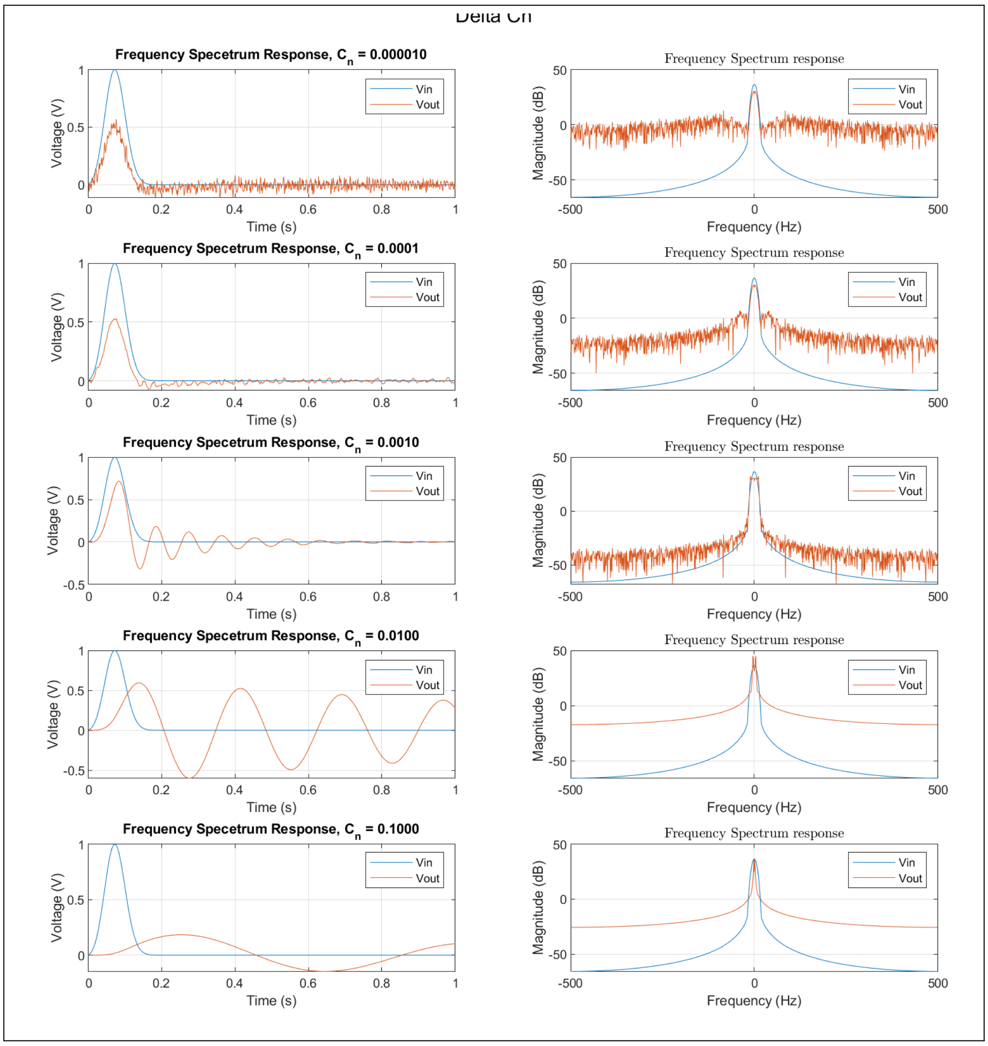

Figure 9: Varrying C_n With Noise Simulation Results

2.3 Varying Δt

This portion of the assignment addresses the effects of varying the time steps. This section is fundamentally the same as the previous (in terms of Matlab Code). The main take away from this simulation is the effects the time step size has on the resolution of the simulations. As expected and explored in previous assignments (analogous to the mesh size concept explored in Assignment 2), more time steps allow for a higher degree of accuracy, at the trade off of simulation time.

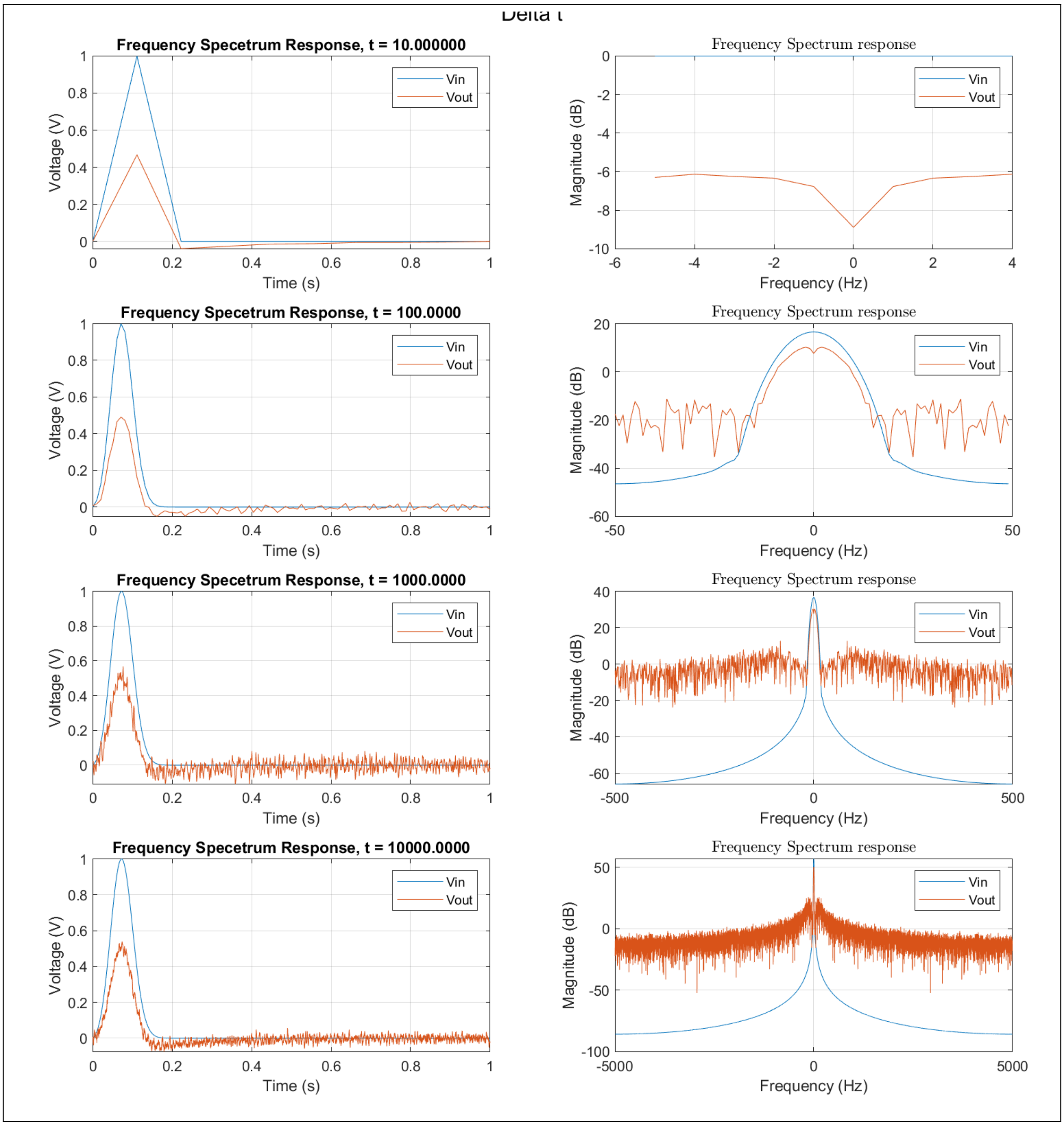

Figure 10: Varrying Time Step With Noise Simulation Results

3 Non-Linearity

The effects of non-linear components within the circuit can be integrated simply by replacing $V = \alpha I_3$ with $V = \alpha I_3 + \beta I_3^2 + \gamma I_3^3$, and implementing this non-linear element inside of a B matrix,

$$C\frac{dV}{dt} + G * V + B = F \tag{4}$$

Where to solve this system to find the voltage,

$$V_i = (\frac{C}{dt} + G)\backslash(F + \frac{C}{dt}V_{i-1} - B) \tag{5}$$

One other possible solution is to implement the Voltage Controlled Voltage Source and input the non-linear coefficients (see included VCVS code in appendix).

4 Appendix

4.1 Main Code

```
%{
Author : Maharshi Gurjar
Elec 4700 Assignment 4 -  Circuit Modeling
Part 1 - No noise simulation
%}
clc; close all; clear;
set(0, 'DefaultFigureWindowStyle', 'docked')
%% Find value for R3 using FDM from assignment 2
%{
Author : Maharshi Gurjar
Elec 4700 Assignment 2 -  Finite Difference Method
Part 2
%}
%% Define G,C,b matrix and fill in
global G C b
NrNodes = 5;  % The total number of nodes in the circuit
% Define G, C, b, for a circuit (do not include additional variables).
G = zeros(NrNodes,NrNodes);
C = zeros(NrNodes,NrNodes);
b = zeros(NrNodes,1);
%set up component values
vol(1,0,1);
res(1,2,1)
cap(1,2,0.25)
res(2,0,2)
ind(2,3,0.2)
res(3,0,186)
vcvs(4,0,3,0,(100/186))
res(4,5,0.1)
res(5,0,1000)
%% DC Sweep of Circuit
VNode3 =[];
Vout =[];
for n= -10:1:10
    b(6) = n; %Mapping sweep for the input
    Voltage = G\b; % Solve for Voltage
    VNode3 = [VNode3 Voltage(3)]; %Voltage at node 3
    Vout = [Vout Voltage(5)]; %Voltage at output node
    %gain = Vout(n)/Vin(n); % Vout/Vin gain
end
figure('name','DC Sweep');
plot(-10:1:10,VNode3, -10:1:10, Vout);
title('DC Sweep','interpreter','latex')
xlabel('Input Voltage (V)')
ylabel('Ouput Voltage (V)')
legend(['V_{node3}';'V_{Ouput}'])
axis tight;
grid on;
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪  Results','[Part1]DC Sweep.png'),'png')
%% Frequency Response of circuit
Freq = logspace(0, 5, 5000); %Define the frequency of the simulation
for n=1:length(Freq)
    w = 2*pi*Freq(n); %define omega at frequency point
    s = 1i*Freq(n); %define s at frequency point
    A = G + (s.*C); %define full G C matrix
    Voltage = A\b; %solve for voltage at frequency
    Vout(n) = abs(Voltage(5));%output at frequency
    gain(n) = 20*log(abs(Voltage(5)));%gain at frequency
```

```
end
figure('name','reseponse');
subplot(2,1,1)
semilogx(Freq, Vout);
xlabel('Frequency (Hz)');
ylabel('V_O (V)');
title('Frequncy Response','interpreter','latex');
axis auto;
grid on;
subplot(2,1,2)
semilogx(Freq, gain);
xlabel('Frequency (Hz)');
ylabel('Gain (dB)');
title('Frequency Response (dB)','interpreter','latex');
axis auto;
grid on;
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪  Results','[Part1]FrequencySweep.png'),'png')
%% Create histogram with capacitance pertubations
std_dev = 0.05;
range = 10e3;
Rand_Cap = 0.25*std_dev.*randn(range,1);
for n=1:range
    C(1,1) = Rand_Cap(n);
    C(2,2) = Rand_Cap(n);
    C(1,2) = -Rand_Cap(n);
    C(2,1) = -Rand_Cap(n);
    s = 2*pi;
    A = G + (s.*C);
    V = A\b;
    gain(n) = 20*log10(abs(abs(V(5)))/abs(V(1)));
end
figure('name','Histogram of gain across a range of capacitors')
histogram(gain)
xlabel('Gain (dB)')
ylabel('Bin count')
title('Gain across a range of capacitors','interpreter','latex')
grid on;
axis auto;
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪  Results','[Part1]Histogram.png'),'png')
%% Transient Analysis
clc; clear;
% Define G,C,b matrix and fill in
global G C b
NrNodes = 5;  % The total number of nodes in the circuit
% Define and create simulation parameters
Sim_Time = 1;
Steps = 1000;
delta = Sim_Time/Steps;
Time = linspace(0,1,Steps);
% Step the frequency of the input
Frequency = 1/0.03;
Frequency2 = 1/0.3;
% Define the three input signals
Input_Step = zeros(1,Steps);
Input_Sin = zeros(1,Steps);
Input_Sin2 = zeros(1,Steps);
Input_Guassian = zeros(1,Steps);

% To make the step input an actual step input
Transition_Time = 0.03;
Input_Step = double(Time>=Transition_Time);

```

```
% Fill in sin input with sin values
Input_Sin = sin(2*pi*Frequency*Time);
Input_Sin2 = sin(2*pi*Frequency2*Time);
% Create Guassian signal values
GaussianDist = makedist('Normal', 'mu', 0.06, 'sigma', 0.03);
GuassianPulse = pdf(GaussianDist, Time);
Input_Pulse = (GuassianPulse.*sin(pi*Time))/max(GuassianPulse.*sin(pi*Time));

% Do trans. sim for step input
Vin = Input_Step;
for i = 1:Steps
    % Define G, C, b, for a circuit (do not include additional variables).
    G = zeros(NrNodes,NrNodes);
    C = zeros(NrNodes,NrNodes);
    b = zeros(NrNodes,1);
    %set up component values

    res(1,2,1)
    cap(1,2,0.25)
    res(2,0,2)
    ind(2,3,0.2)
    res(3,0,185)
    vcvs(4,0,3,0,(100/185))
    res(4,5,0.1)
    res(5,0,1000)
    vol(1,0,Vin(i));
    if i == 1
        NewVoltage = (G + (C./delta)) \ b;
    else
        NewVoltage = (G + (C./delta)) \ (b  + (C./delta)*OldVoltage);
    end
    Voutput(i) = NewVoltage(5);
    OldVoltage = NewVoltage;
end
Xaxis = (-length(Vin)/2:length(Vin)/2-1);
figure('Name','Trans Sim Step')
subplot(2,1,1)
plot(Time,Vin,Time,Voutput)
grid on;
axis auto;
legend('Vin','Vout')
xlabel('Time (s)')
ylabel('Voltage (V)')
title('Step response of circuit','interpreter','latex')
subplot(2,1,2)
plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
title('Frequency Spectrum response','interpreter','latex')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
grid on;
axis auto;
legend('Vin','Vout')
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪ Results','[Part1]Step input response.png'),'png')

% Do trans. sim for Sin input
Vin = Input_Sin;
for i = 1:Steps
    % Define G, C, b, for a circuit (do not include additional variables).
    G = zeros(NrNodes,NrNodes);
    C = zeros(NrNodes,NrNodes);
    b = zeros(NrNodes,1);
    %set up component values
    vol(1,0,Vin(i));
```

```
    res(1,2,1)
    cap(1,2,0.25)
    res(2,0,2)
    ind(2,3,0.2)
    res(3,0,185)
    vcvs(4,0,3,0,(100/185))
    res(4,5,0.1)
    res(5,0,1000)

    if i == 1
        NewVoltage = (G + (C./delta)) \ b;
    else
        NewVoltage = (G + (C./delta)) \ (b  + (C./delta)*OldVoltage);
    end
    Voutput(i) = NewVoltage(5);
    OldVoltage = NewVoltage;
end
Xaxis = (-length(Vin)/2:length(Vin)/2-1);
figure('Name','Trans Sim Sin')
subplot(2,1,1)
plot(Time,Vin,Time,Voutput)
grid on;
axis auto;
legend('Vin','Vout')
xlabel('Time (s)')
ylabel('Voltage (V)')
title('Sin response of circuit','interpreter','latex')
subplot(2,1,2)
plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
title('Frequency Spectrum response','interpreter','latex')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
grid on;
axis auto;
legend('Vin','Vout')
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪  Results','[Part1]Sin1.png'),'png')

%Impulse response
Vin = Input_Pulse;
for i = 1:Steps
    % Define G, C, b, for a circuit (do not include additional variables).
    G = zeros(NrNodes,NrNodes);
    C = zeros(NrNodes,NrNodes);
    b = zeros(NrNodes,1);
    %set up component values

    res(1,2,1)
    cap(1,2,0.25)
    res(2,0,2)
    ind(2,3,0.2)
    res(3,0,185)
    vcvs(4,0,3,0,(100/185))
    res(4,5,0.1)
    res(5,0,1000)
    vol(1,0,Vin(i));
    if i == 1
        NewVoltage = (G + (C./delta)) \ b;
    else
        NewVoltage = (G + (C./delta)) \ (b  + (C./delta)*OldVoltage);
    end
    Voutput(i) = NewVoltage(5);
    OldVoltage = NewVoltage;
end
```

```
Xaxis = (-length(Vin)/2:length(Vin)/2-1);
figure('Name','Trans Sim Impulse')
subplot(2,1,1)
plot(Time,Vin,Time,Voutput)
grid on;
axis auto;
legend('Vin','Vout')
xlabel('Time (s)')
ylabel('Voltage (V)')
title('Pulse response of circuit','interpreter','latex')
subplot(2,1,2)
plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
title('Frequency Spectrum response','interpreter','latex')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
grid on;
axis auto;
legend('Vin','Vout')
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation Results','[Part1]Impluse.png'),'png')

% Do trans. sim for Sin input`
Vin = Input_Sin2;
for i = 1:Steps
    % Define G, C, b, for a circuit (do not include additional variables).
    G = zeros(NrNodes,NrNodes);
    C = zeros(NrNodes,NrNodes);
    b = zeros(NrNodes,1);
    %set up component values
    res(1,2,1)
    cap(1,2,0.25)
    res(2,0,2)
    ind(2,3,0.2)
    res(3,0,185)
    vcvs(4,0,3,0,(100/185))
    res(4,5,0.1)
    res(5,0,1000)
    vol(1,0,Vin(i));
    if i == 1
        NewVoltage = (G + (C./delta)) \ b;
    else
        NewVoltage = (G + (C./delta)) \ (b  + (C./delta)*OldVoltage);
    end
    Voutput(i) = NewVoltage(5);
    OldVoltage = NewVoltage;
end
Xaxis = (-length(Vin)/2:length(Vin)/2-1);
figure('Name','Trans Sim Sin diff Freq')
subplot(2,1,1)
plot(Time,Vin,Time,Voutput)
grid on;
axis auto;
legend('Vin','Vout')
xlabel('Time (s)')
ylabel('Voltage (V)')
title('Sin response of circuit','interpreter','latex')
subplot(2,1,2)
plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
title('Frequency Spectrum response','interpreter','latex')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
grid on;
axis auto;
legend('Vin','Vout')
```

```
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation Results','[Part1]Sin2.png'),'png')
%% Circuit with noise
clc; clear;
% Define G,C,b matrix and fill in
global G C b
NrNodes = 5;  % The total number of nodes in the circuit
% Define and create simulation parameters
Sim_Time = 1;
Steps = 1000;
delta = Sim_Time/Steps;
Time = linspace(0,1,Steps);
% Step the frequency of the input
Frequency = 1/0.03;

Input_Guassian = zeros(1,Steps);

GaussianDist = makedist('Normal', 'mu', 0.06, 'sigma', 0.03);
GuassianPulse = pdf(GaussianDist, Time);
Input_Pulse = (GuassianPulse.*sin(pi*Time))/max(GuassianPulse.*sin(pi*Time));

Vin = Input_Pulse;
Vout = zeros(1,Steps);
Cn = 0.00001;
In = 0.001;
h = 1/10e3;
W = 0:h:1;
CurrNoise = In*randn(1,numel(W));
for i = 1:Steps
    % Define G, C, b, for a circuit (do not include additional variables).
    G = zeros(NrNodes,NrNodes);
    C = zeros(NrNodes,NrNodes);
    b = zeros(NrNodes,1);
    %set up component values
    vol(1,0,Vin(i));
    cur(3,0,CurrNoise(i));
    res(1,2,1)
    cap(1,2,0.25)
    cap(3,0,Cn)
    res(2,0,2)
    ind(2,3,0.2)
    res(3,0,185)
    res(4,5,0.1)
    res(5,0,1000)
    vcvs(4,0,3,0,100/185);
    if i == 1
        NewVoltage = (G + (C./delta)) \ b;
    else
        NewVoltage = (G + (C./delta)) \ (b  + (C./delta)*OldVoltage);
    end
    Voutput(i) = NewVoltage(5);
    OldVoltage = NewVoltage;
end

C

Xaxis = (-length(Vin)/2:length(Vin)/2-1);
figure('Name','Trans Sim Noise')
subplot(2,1,1)
plot(Time,Vin,Time,Voutput)
grid on;
axis auto;
legend('Vin','Vout')
xlabel('Time (s)')
```

```
ylabel('Voltage (V)')
title('Sin response of circuit','interpreter','latex')
subplot(2,1,2)
plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
title('Frequency Spectrum response','interpreter','latex')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
grid on;
axis auto;
legend('Vin','Vout')
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪ Results','[Part2]NoisySignal.png'),'png')
% Varrying Cn

Cn = [0.00001 0.0001 0.001 0.01 0.1];
for i = 1:numel(Cn)
    for j = 1:Steps
        % Define G, C, b, for a circuit (do not include additional variables).
        G = zeros(NrNodes,NrNodes);
        C = zeros(NrNodes,NrNodes);
        b = zeros(NrNodes,1);
        %set up component values
        vol(1,0,Vin(j));
        cur(3,0,CurrNoise(j));
        res(1,2,1)
        cap(1,2,0.25)
        cap(3,0,Cn(i))
        res(2,0,2)
        ind(2,3,0.2)
        res(3,0,185)
        res(4,5,0.1)
        res(5,0,1000)
        vcvs(4,0,3,0,100/185);
        if j == 1
            NewVoltage = (G + (C./delta)) \ b;
        else
            NewVoltage = (G + (C./delta)) \ (b  + (C./delta)*OldVoltage);
        end
        Voutput(j) = NewVoltage(5);
        OldVoltage = NewVoltage;
    end
    Xaxis = (-length(Vin)/2:length(Vin)/2-1);
    sgtitle('Delta Cn')
    if i == 1
        figure(21)
        subplot(5,2,i)
        plot(Time,Vin,Time,Voutput)
        grid on;
        axis auto;
        legend('Vin','Vout')
        xlabel('Time (s)')
        ylabel('Voltage (V)')
        title(sprintf('Frequency Specetrum Response, C_n = %f',Cn(i)))
        hold on;
        subplot(5,2,i+1)
        plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
        title('Frequency Spectrum response','interpreter','latex')
        xlabel('Frequency (Hz)')
        ylabel('Magnitude (dB)')
        grid on;
        axis auto;
        legend('Vin','Vout')
        hold on;
    else
```

```
        figure(21)
        subplot(5,2,(i*2-1))
        plot(Time,Vin,Time,Voutput)
        grid on;
        axis auto;
        legend('Vin','Vout')
        xlabel('Time (s)')
        ylabel('Voltage (V)')
        title(sprintf('Frequency Specetrum Response, C_n = %.4f',Cn(i)))
        hold on;
        subplot(5,2,i*2)
        plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
        title('Frequency Spectrum response','interpreter','latex')
        xlabel('Frequency (Hz)')
        ylabel('Magnitude (dB)')
        grid on;
        axis auto;
        legend('Vin','Vout')
        hold on;
    end
end
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation Results','[Part2]VaryingCn.png'),'png')

%Varrying Time

Steps = [10 100 1e3 1e4];

Cn = 0.00001;
for i = 1: numel(Steps)
    delta = 1/Steps(i);
    Time = linspace(0,1,Steps(i));
    GaussianDist = makedist('Normal', 'mu', 0.06, 'sigma', 0.03);
    GuassianPulse = pdf(GaussianDist, Time);
    Input_Pulse = (GuassianPulse.*sin(pi*Time))/max(GuassianPulse.*sin(pi*Time));
    Vin = Input_Pulse;
    Voutput=zeros(1,Steps(i));
    for j = 1:Steps(i)
        % Define G, C, b, for a circuit (do not include additional variables).
        G = zeros(NrNodes,NrNodes);
        C = zeros(NrNodes,NrNodes);
        b = zeros(NrNodes,1);
        %set up component values
        vol(1,0,Vin(j));
        cur(3,0,CurrNoise(j));
        res(1,2,1)
        cap(1,2,0.25)
        cap(3,0,Cn)
        res(2,0,2)
        ind(2,3,0.2)
        res(3,0,185)
        res(4,5,0.1)
        res(5,0,1000)
        vcvs(4,0,3,0,100/185);
        if j == 1
            NewVoltage = (G + (C./delta)) \ b;
        else
            NewVoltage = (G + (C./delta)) \ (b  + (C./delta)*OldVoltage);
        end
        Voutput(j) = NewVoltage(5);
        OldVoltage = NewVoltage;
    end
    Xaxis = (-length(Vin)/2:length(Vin)/2-1);
    sgtitle('Delta t')
```

```
    if i == 1
        figure(22)
        subplot(4,2,i)
        plot(Time,Vin,Time,Voutput)
        grid on;
        axis auto;
        legend('Vin','Vout')
        xlabel('Time (s)')
        ylabel('Voltage (V)')
        title(sprintf('Frequency Specetrum Response, t = %f',Steps(i)))
        hold on;
        subplot(4,2,i+1)
        plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
        title('Frequency Spectrum response','interpreter','latex')
        xlabel('Frequency (Hz)')
        ylabel('Magnitude (dB)')
        grid on;
        axis auto;
        legend('Vin','Vout')
        hold on;
    else
        figure(22)
        subplot(4,2,(i*2-1))
        plot(Time,Vin,Time,Voutput)
        grid on;
        axis auto;
        legend('Vin','Vout')
        xlabel('Time (s)')
        ylabel('Voltage (V)')
        title(sprintf('Frequency Specetrum Response, t = %.4f',Steps(i)))
        hold on;
        subplot(4,2,i*2)
        plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
        title('Frequency Spectrum response','interpreter','latex')
        xlabel('Frequency (Hz)')
        ylabel('Magnitude (dB)')
        grid on;
        axis auto;
        legend('Vin','Vout')
        hold on;
    end
end
%saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪    Results','[Part2]VarryingTime.png'),'png')
```

Source Code 1: Full Assignment 4 Code

4.2 Stamps

4.2.1 Resistor Stamp

```
function res(n1,n2,val)
% Adds the stamp of a resistor with a value of "val" (Ohms)
% connected between nodes n1 and n2 to the G matrix in
% circuit representation.
%
%                   val
%       n1 O-----/\/\/\/\---O n2     where R=val (ohms)
%---------------------------------------------------------
global G   %define global variable

if (n1 ~= 0)
    G(n1,n1) = G(n1,n1) + 1/val;
end

if (n2 ~= 0)
    G(n2,n2) = G(n2,n2) + 1/val;
end

if (n1 ~= 0) && (n2 ~= 0)
    G(n1,n2) = G(n1,n2) - 1/val;
    G(n2,n1) = G(n2,n1) - 1/val;
end

end %func
```

Source Code 2: MNA Stamps - Resistor

4.2.2 Capacitor Stamp

```
function cap(n1,n2,val)
% Adds the stamp of a capacitor with a value of "val"
% (Farads) connected between nodes n1 and n2 to the
% C matrix in circuit representation.
%
%              val
%       n1 o---||---o n2     where C= val (Farads)
%----------------------------------------------------------
global C   %define global variable

if (n1 ~= 0)
    C(n1,n1) = C(n1,n1) + val;
end

if (n2 ~= 0)
    C(n2,n2) = C(n2,n2) + val;

if (n1 ~= 0) && (n2 ~= 0)
    C(n1,n2) = C(n1,n2) - val;
    C(n2,n1) = C(n2,n1) - val;
end

end %func
```

Source Code 3: MNA Stamps - Capcitor

4.2.3 Inductor Stamp

```
function ind(n1, n2, val)
global G C b   %define global variables
d = size(G,1); % current size of the MNA
xr = d+1;      % new (extera)row/column

% Using an index bigger than the current size,  Matlab automatically
... increases the size of the matrix:

G(xr,xr) = 0; % add new row/column
C(xr,xr) = 0;
b(xr) = 0;    % add new row

if (n1 ~= 0)
    G(n1,xr) = 1;
    G(xr,n1) = 1;
end
if (n2 ~= 0)
    G(n2,xr) = -1;
    G(xr,n2) = -1;
end

    C(xr,xr) = - val;

end
```

Source Code 4: MNA Stamps - Inductor

4.2.4 Voltage Source Stamp

```
function vol(n1,n2,val)
% Adds the stamp of an independent voltage source with a value
% of "val" (Volts) connected between nodes n1 and n2 to the
% matrices in circuit representation.
%
%                   val
%                  /  \
%      n1 O-------(+  -)--------O n2    where Vsrc= val (volts)
%                  \  /
%             Isrc ---->
%-----------------------------------------------------------------

global G b C   % define global variables

d = size(G,1); % current size of the MNA
xr = d+1;      % new (extera)row/column

% Using an index bigger than the current size, Matlab
...automaticallyincreases the size of the matrix:

G(xr,xr) = 0; % add new row/column
C(xr,xr) = 0;

if (n1 ~= 0)
    G(n1,xr) = 1;
    G(xr,n1) = 1;
end

if (n2 ~= 0)
```

```
    G(n2,xr) = -1;
    G(xr,n2) = -1;
end
b(xr) = val;

end %func
```

Source Code 5: MNA Stamps - Voltage Source

4.2.5 Current Source Stamp

```
function cur(n1,n2,val)
% Adds the stamp of an independent current source with a
% value of "val" (Amperes) connected between nodes n1 and
% n2 to the the source vector b in circuit representation.
%
%                    val
%                   /  \
%      n1 O-------(--->)-------O n2   where J=val (Amperes)
%                   \  /
%
%      n1: The node at the tail of the current arrow!
%      n2:  "    "    "    " head  "    "    "       "  !
%     val: The value of the current source (Amp)
%---------------------------------------------------------
global b   %define global variable

if (n1 ~= 0)
    b(n1,1) = val;
end

if (n2 ~= 0)
    b(n2,1) = val;
end

end %func
```

Source Code 6: MNA Stamps - Current Source

4.2.6 Voltage Controlled Voltage Source Stamp

```
function vcvs(nd1,nd2,ni1,ni2,val)
% Adds the stamp of a dependent voltage-controlled
% voltage-source(VCVS)to the matrices in circuit
% representation.
%
%   ni1 O-------O          |----------o nd1
%                          |
%                         /+\
%                      | /   \     Vnd1-Vnd2 = val*(Vni1-Vni2)
%               Ivcvs  | \   /
%                      V  \-/
%                          |
%   ni2 O-------O          |----------o nd2
%
%  (1) "nd1 & nd2" are the nodes across the dependent
%                  voltage source.
%  (2) "ni1 & ni2" are the nodes corresponding to the
```

```
%                  controller voltage
%
%   nd1: (+) node   \
%   nd2: (-) node   |----->  Vnd1-Vnd2 = val*(Vni1-Vni2)
%   ni1: (+) node   |
%   ni2: (-) node   /
%-------------------------------------------------------------
global G C b   %define global variables

d = size(G,1); % current size of the MNA
xr = d+1;      % new (extera)row/column

% Using an index bigger than the current size,  Matlab automatically
... increases the size of the matrix:
G(xr,xr) = 0; % add new row/column
C(xr,xr) = 0;
b(xr) = 0;     % add new row

if (nd1 ~= 0)
    G(nd1,xr) = 1;
    G(xr,nd1) = 1;
end
if (nd2 ~= 0)
    G(nd2,xr) = -1;
    G(xr,nd2) = -1;
end

if (ni1 ~= 0)
    G(xr,ni1) = G(xr,ni1)-val;
end
if (ni2 ~= 0)
    G(xr,ni2) = G(xr,ni2)+val;
end
end %func
```

Source Code 7: MNA Stamps - Voltage Controlled Voltage Source