

CARLETON UNIVERSITY

MODELLING OF INTEGRATED DEVICES
ELEC 4700

Assignment 4 - Circuit Modeling

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



Carleton
UNIVERSITY

Contents

1	Noiseless Circuit Modeling	2
1.1	DC Sweep	2
1.2	AC Sweep	2
1.3	Random Perturbations of Capacitance	4
1.4	Transient and Frequency Spectrum Simulation	5
2	Noisy Circuit Modelling	9
2.1	Noisy Signal	9
2.2	Varying C_n	10
2.3	Varying Δt	11
3	Non-Linearity	12
4	Appendix	13
4.1	Main Code	13
4.2	Stamps	22
4.2.1	Resistor Stamp	22
4.2.2	Capacitor Stamp	22
4.2.3	Inductor Stamp	23
4.2.4	Voltage Source Stamp	23
4.2.5	Current Source Stamp	24
4.2.6	Voltage Controlled Voltage Source Stamp	24

List of Figures

1	DC Sweep Simulation Results	3
2	Frequency Sweep Simulation Results	3
3	Random C Perturbations Simulation Results	4
4	Step Input Simulation Results	5
5	Sin Input Simulation Results	6
6	Sin Input (Different Frequency) Simulation Results	7
7	Impulse Input Simulation Results	8
8	Impulse Input With Noise Transient Simulation Results	9
9	Varying C_n With Noise Simulation Results	10
10	Varying Time Step With Noise Simulation Results	11

List of Source Codes

1	Full Assignment 4 Code	21
2	MNA Stamps - Resistor	22
3	MNA Stamps - Capacitor	22
4	MNA Stamps - Inductor	23
5	MNA Stamps - Voltage Source	24
6	MNA Stamps - Current Source	24
7	MNA Stamps - Voltage Controlled Voltage Source	25

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,

$$\mathbf{C} \frac{d\mathbf{V}}{dt} + \mathbf{G}\mathbf{V} = \mathbf{F} \quad (1)$$

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

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

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

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

$$\mathbf{F} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

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

$$\mathbf{V} = \mathbf{G} \backslash \mathbf{F} \quad (3)$$

Where the ' \backslash ' 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.

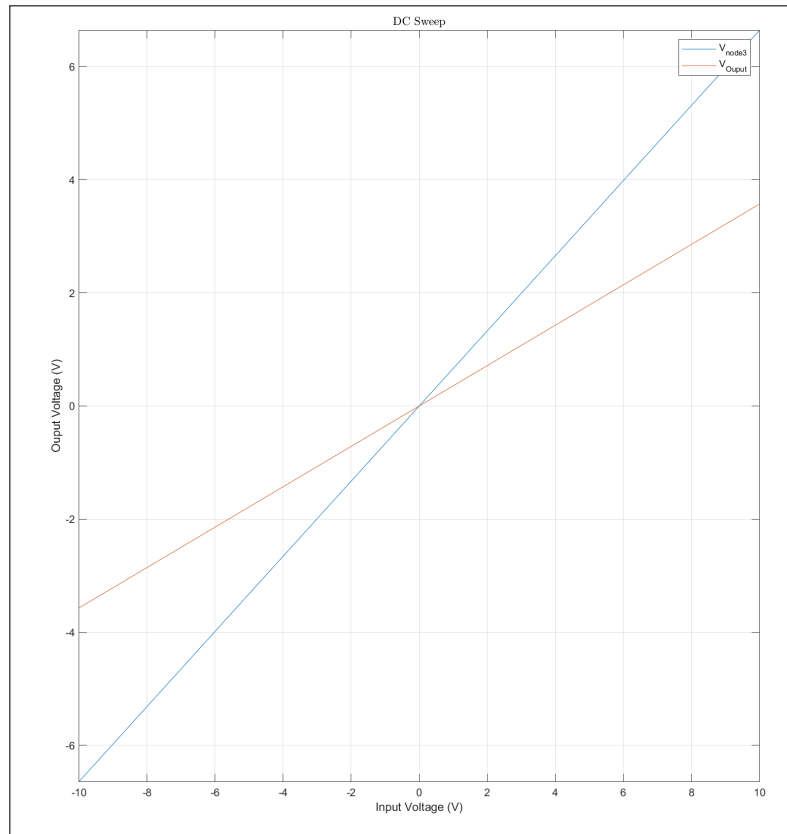


Figure 1: DC Sweep Simulation Results

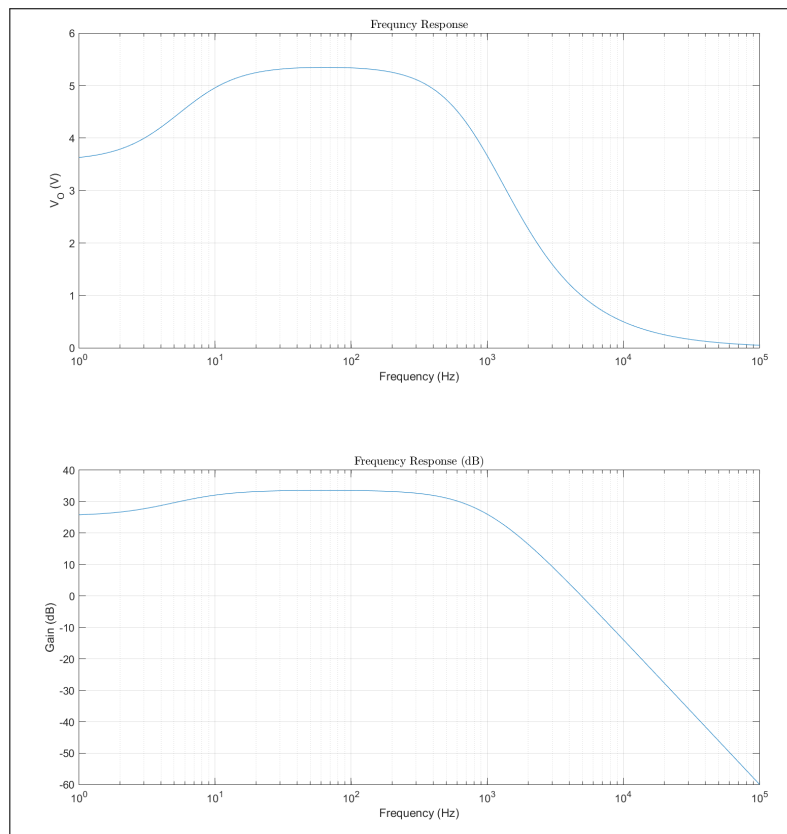


Figure 2: Frequency Sweep Simulation Results

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.

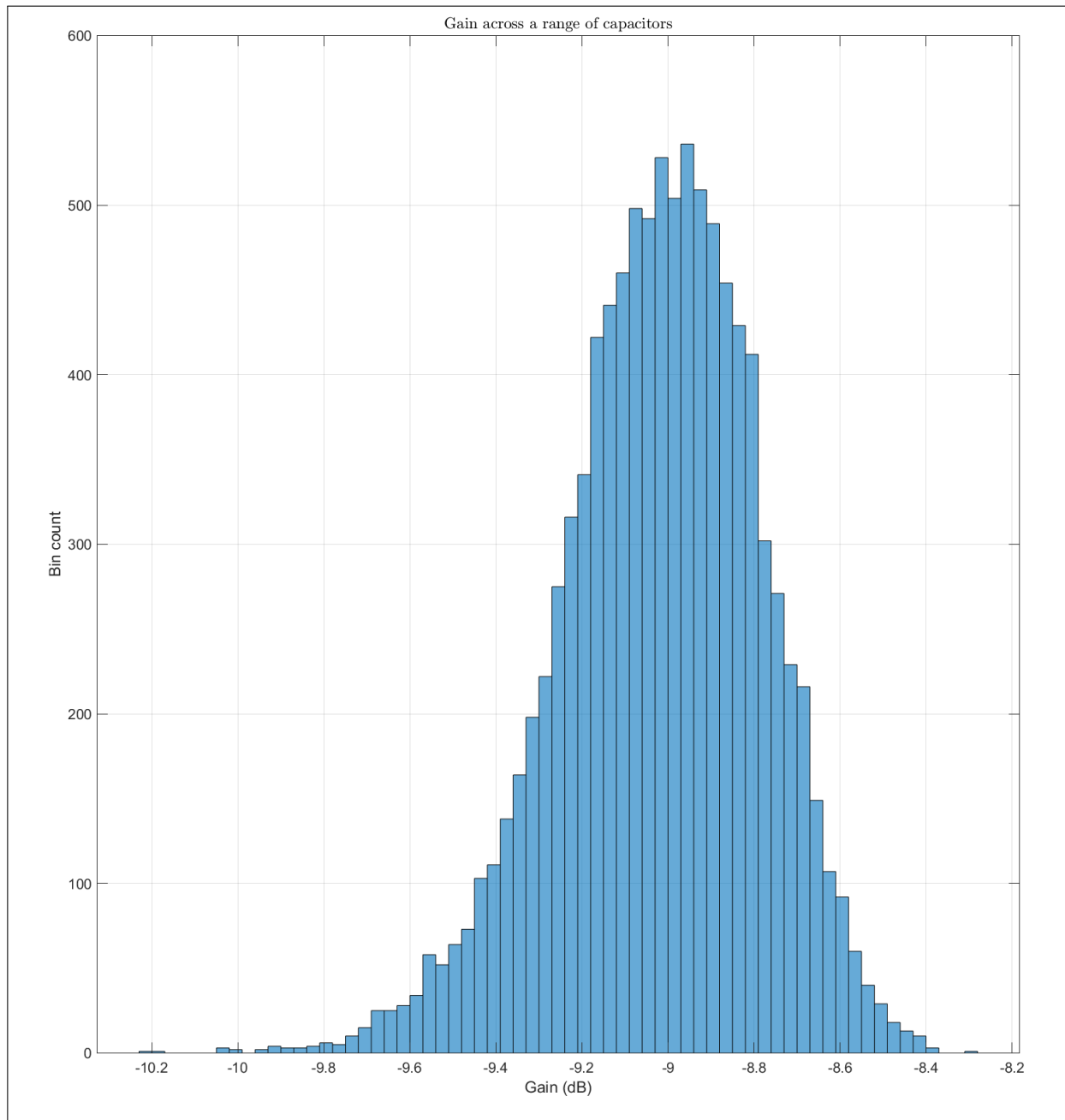


Figure 3: Random C Perturbations Simulation Results

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.

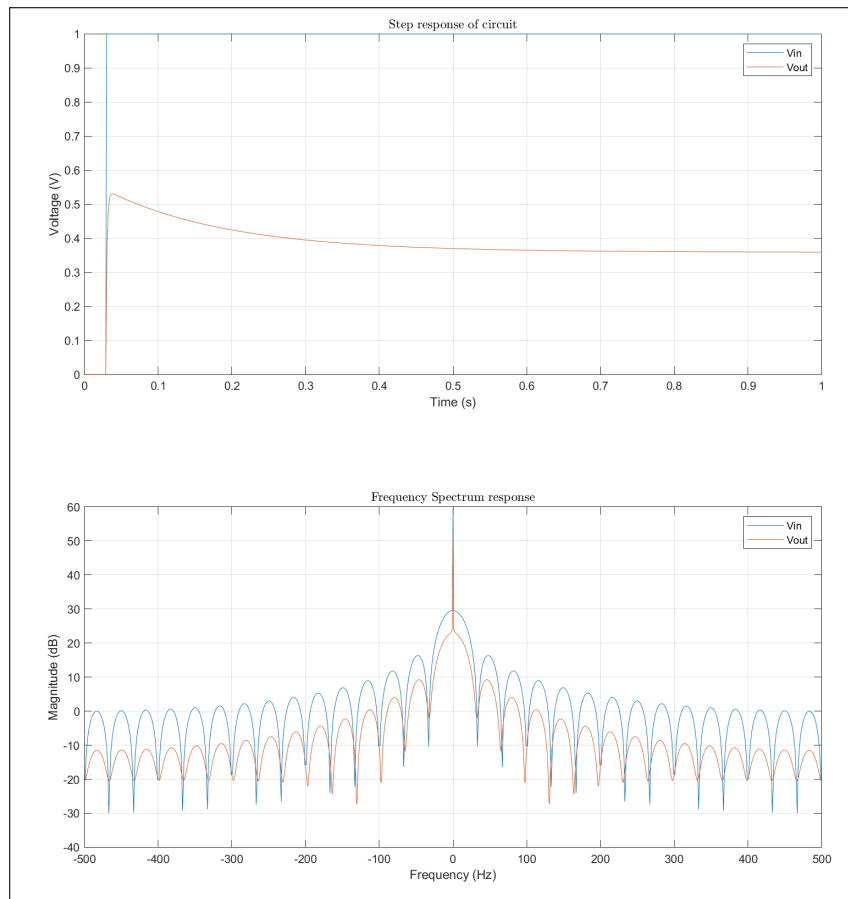


Figure 4: Step Input Simulation Results

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

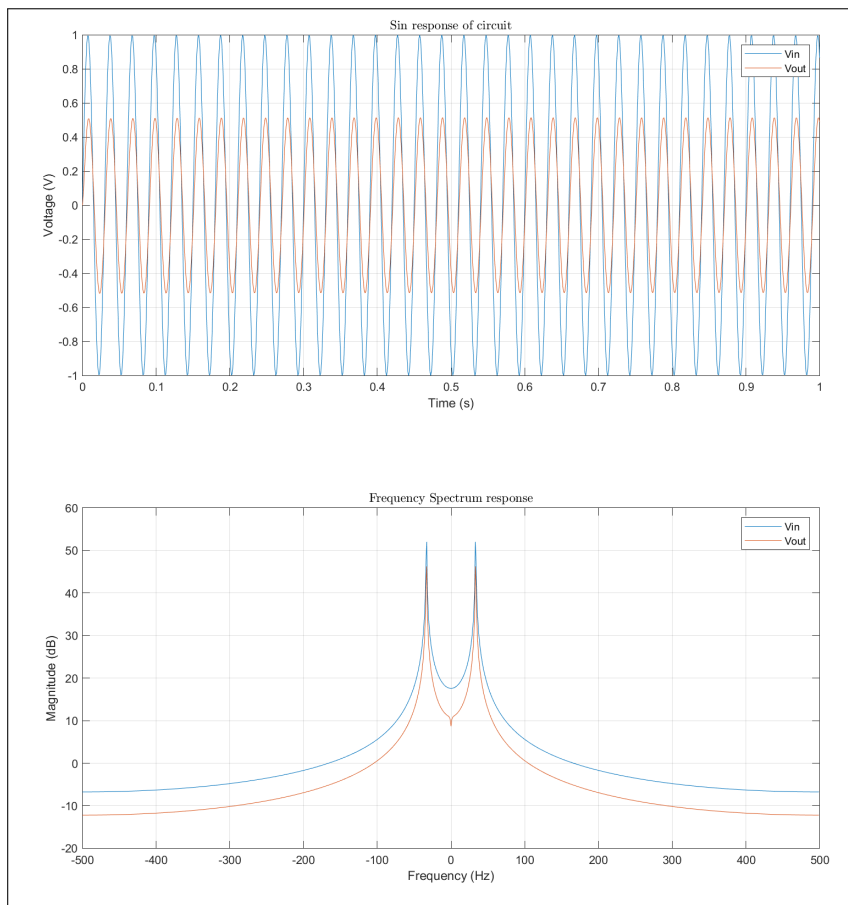


Figure 5: Sin Input Simulation Results

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.

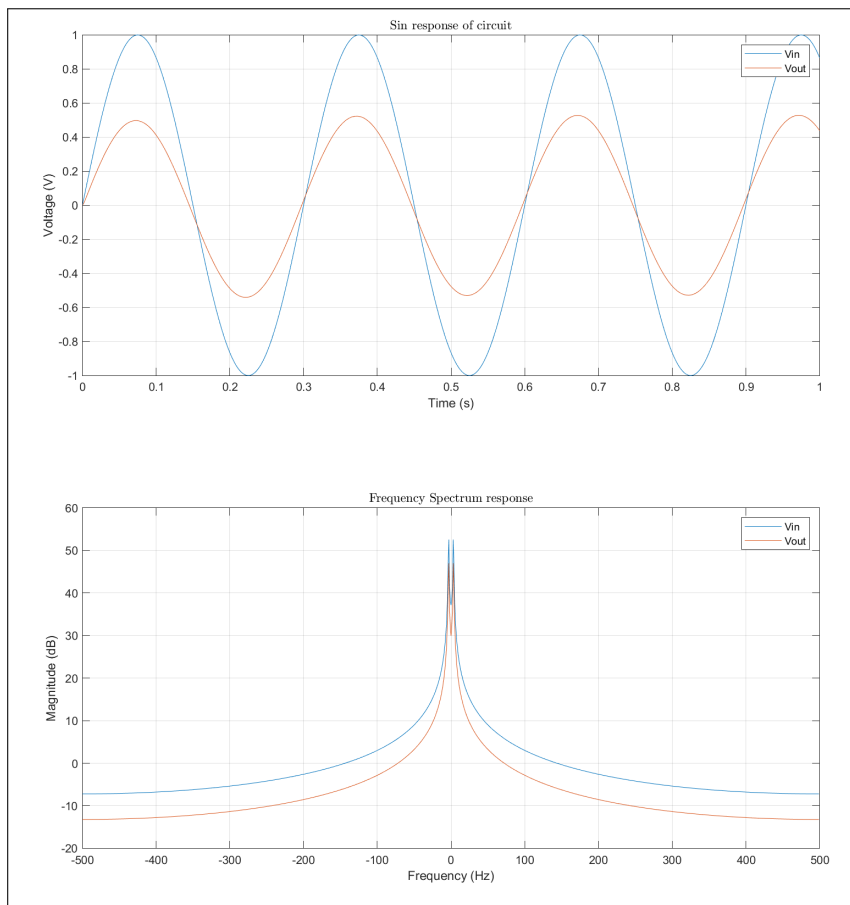


Figure 6: Sin Input (Different Frequency) Simulation Results

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

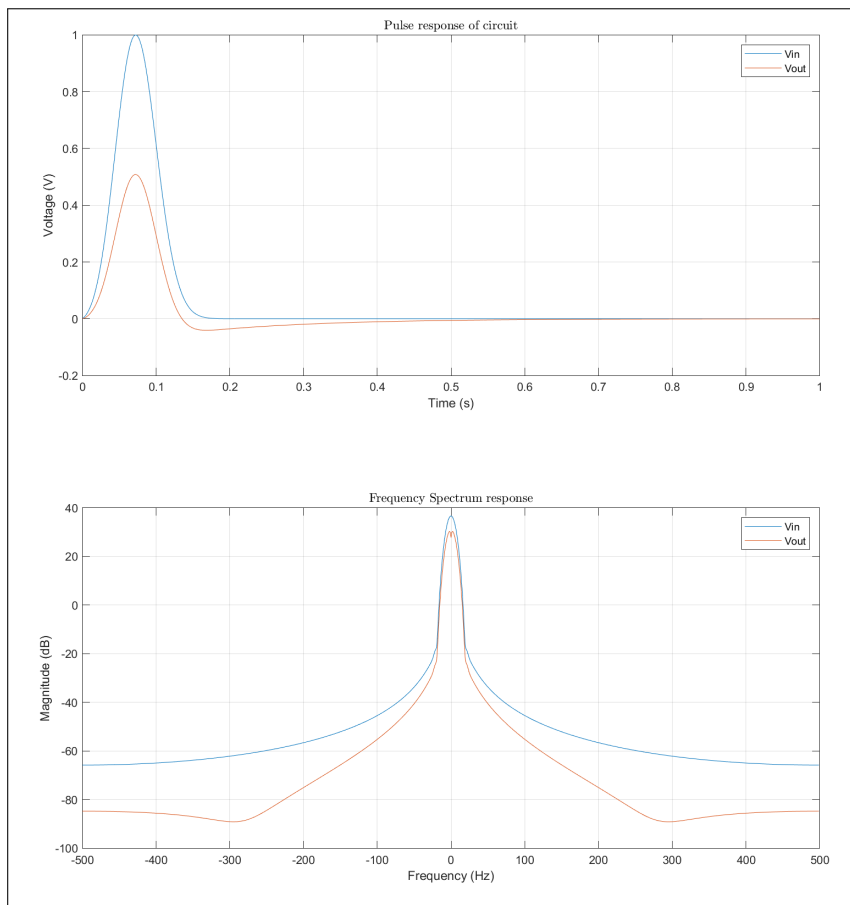


Figure 7: Impulse Input Simulation Results

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.

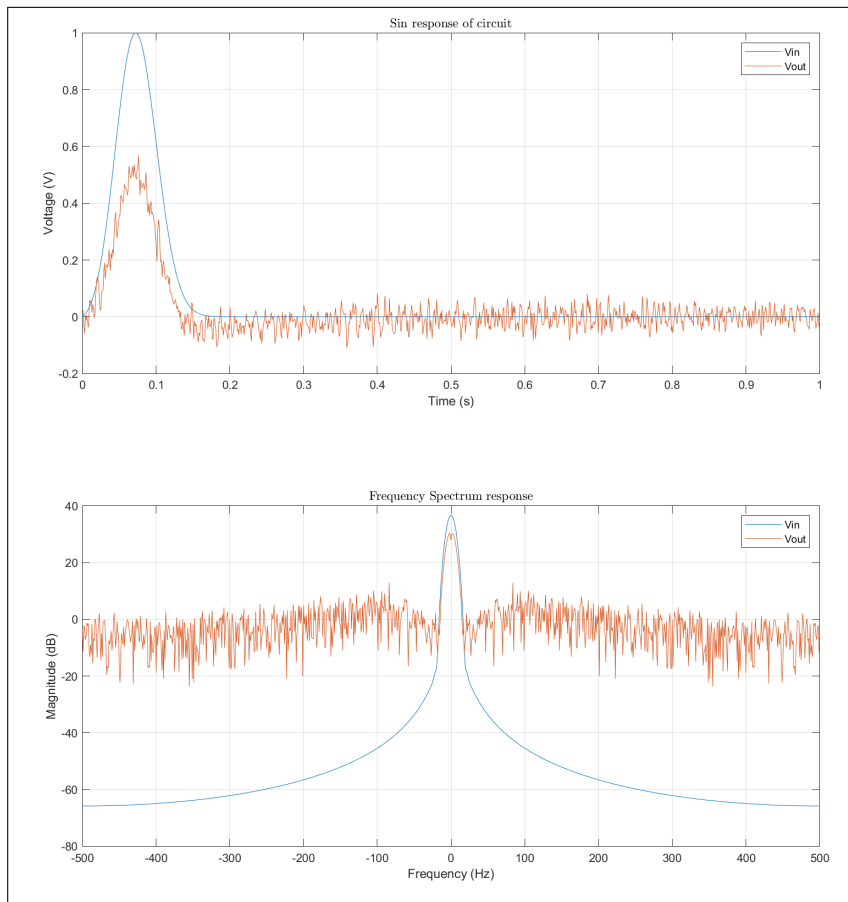


Figure 8: Impulse Input With Noise Transient Simulation Results

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.

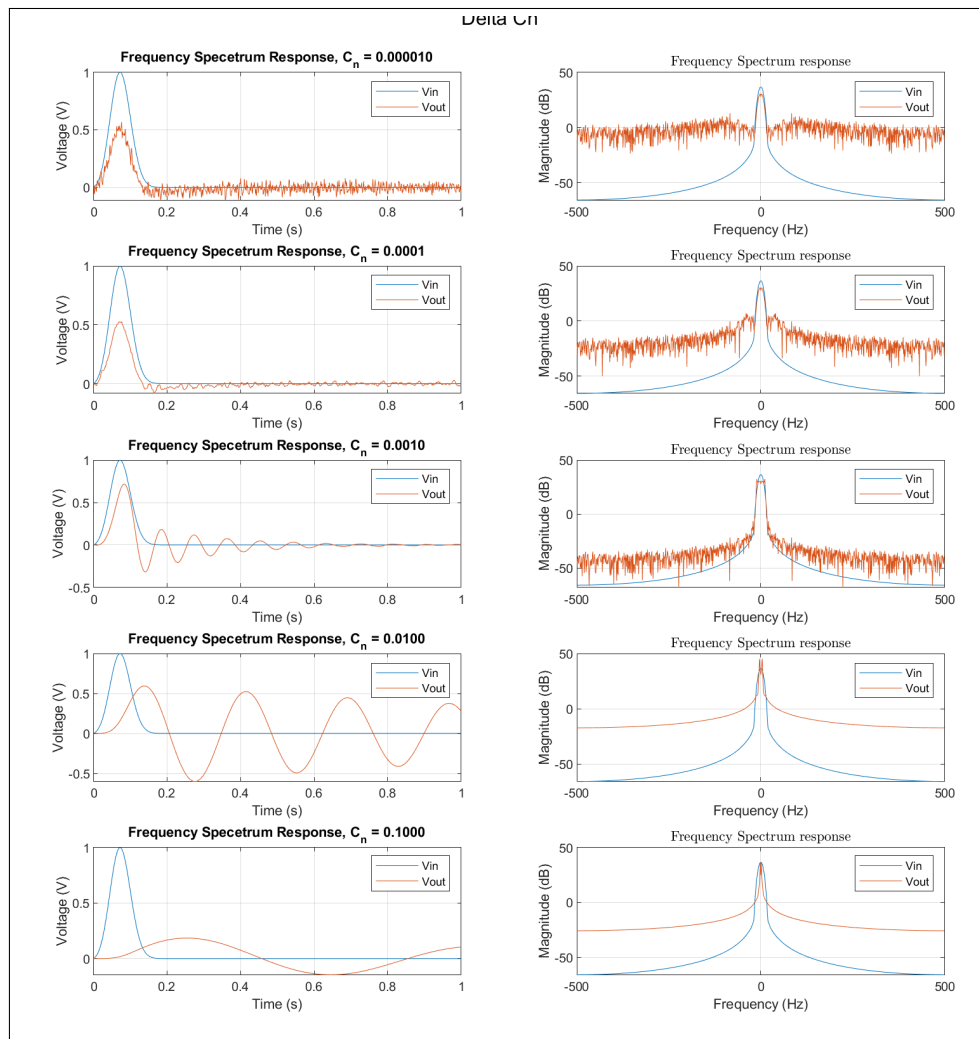


Figure 9: Varying C_n With Noise Simulation Results

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.

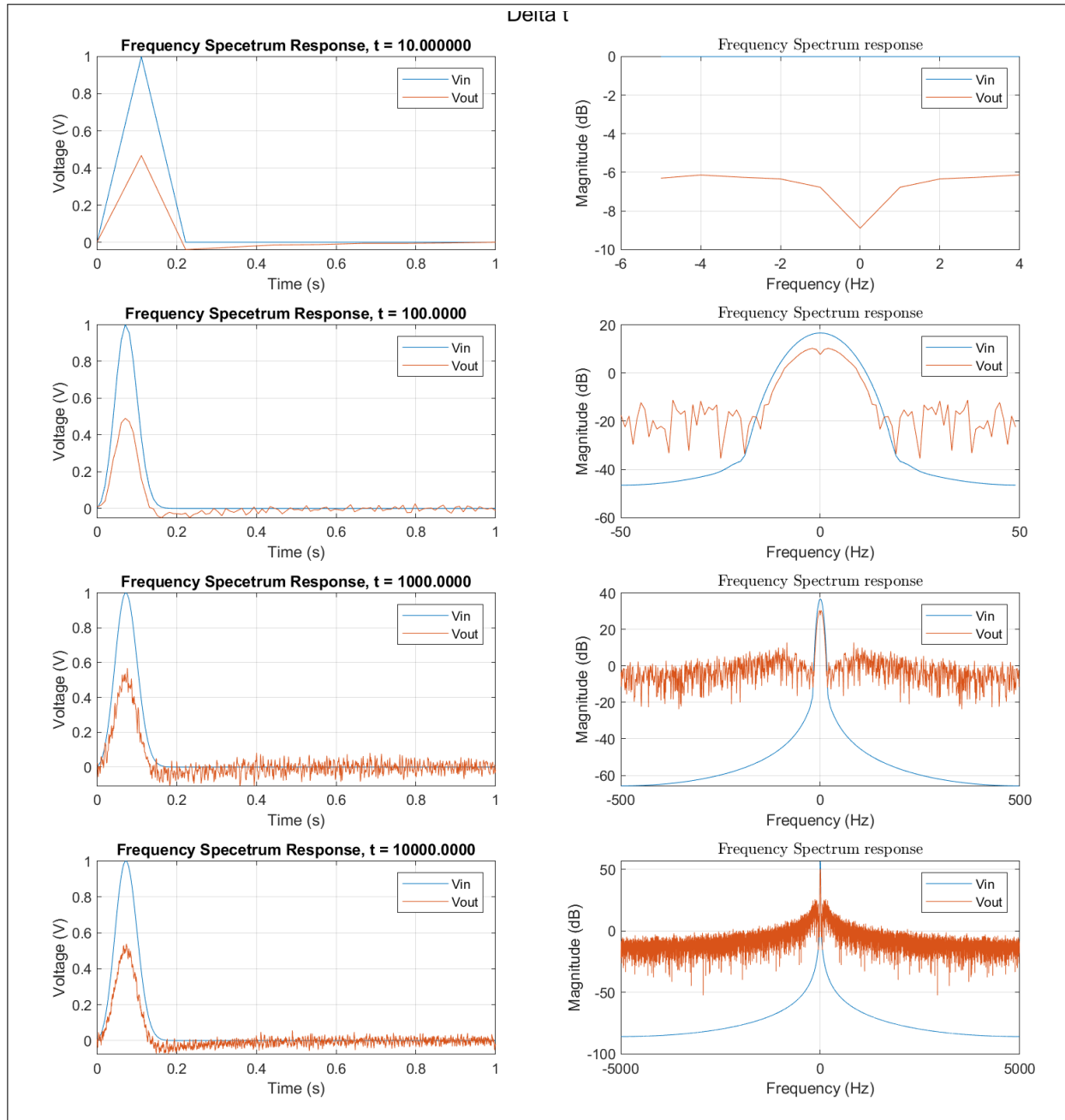


Figure 10: Varying Time Step With Noise Simulation Results

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,

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

Where to solve this system to find the voltage,

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

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

```

1  %{
2  Author : Maharshi Gurjar
3  Elec 4700 Assignment 4 - Circuit Modeling
4  Part 1 - No noise simulation
5  %}
6  clc; close all; clear;
7  set(0, 'DefaultFigureWindowStyle', 'docked')
8  %% Find value for R3 using FDM from assignment 2
9  %{
10 Author : Maharshi Gurjar
11 Elec 4700 Assignment 2 - Finite Difference Method
12 Part 2
13 %}
14 %% Define G,C,b matrix and fill in
15 global G C b
16 NrNodes = 5; % The total number of nodes in the circuit
17 % Define G, C, b, for a circuit (do not include additional variables).
18 G = zeros(NrNodes,NrNodes);
19 C = zeros(NrNodes,NrNodes);
20 b = zeros(NrNodes,1);
21 %set up component values
22 vol(1,0,1);
23 res(1,2,1)
24 cap(1,2,0.25)
25 res(2,0,2)
26 ind(2,3,0.2)
27 res(3,0,186)
28 vcvs(4,0,3,0,(100/186))
29 res(4,5,0.1)
30 res(5,0,1000)
31 %% DC Sweep of Circuit
32 VNode3 = [];
33 Vout = [];
34 for n= -10:1:10
35     b(6) = n; %Mapping sweep for the input
36     Voltage = G\b; % Solve for Voltage
37     VNode3 = [VNode3 Voltage(3)]; %Voltage at node 3
38     Vout = [Vout Voltage(5)]; %Voltage at output node
39     %gain = Vout(n)/Vin(n); % Vout/Vin gain
40 end
41 figure('name','DC Sweep');
42 plot(-10:1:10,VNode3, -10:1:10, Vout);
43 title('DC Sweep','interpreter','latex')
44 xlabel('Input Voltage (V)')
45 ylabel('Output Voltage (V)')
46 legend(['V_{node3}'; 'V_{Output}'])
47 axis tight;
48 grid on;
49 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
    ↳ Results','[Part1]DC Sweep.png'),'png')
50 %% Frequency Response of circuit
51 Freq = logspace(0, 5, 5000); %Define the frequency of the simulation
52 for n=1:length(Freq)
53     w = 2*pi*Freq(n); %define omega at frequency point
54     s = 1i*Freq(n); %define s at frequency point
55     A = G + (s.*C); %define full G C matrix
56     Voltage = A\b; %solve for voltage at frequency
57     Vout(n) = abs(Voltage(5)); %output at frequency
58     gain(n) = 20*log(abs(Voltage(5))); %gain at frequency

```

```

59 end
60 figure('name','reseponse');
61 subplot(2,1,1)
62 semilogx(Freq, Vout);
63 xlabel('Frequency (Hz)');
64 ylabel('V_0 (V)');
65 title('Frequency Response','interpreter','latex');
66 axis auto;
67 grid on;
68 subplot(2,1,2)
69 semilogx(Freq, gain);
70 xlabel('Frequency (Hz)');
71 ylabel('Gain (dB)');
72 title('Frequency Response (dB)','interpreter','latex');
73 axis auto;
74 grid on;
75 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↵ Results','[Part1]FrequencySweep.png'),'png')
76 %% Create histogram with capacitance pertubations
77 std_dev = 0.05;
78 range = 10e3;
79 Rand_Cap = 0.25*std_dev.*randn(range,1);
80 for n=1:range
81     C(1,1) = Rand_Cap(n);
82     C(2,2) = Rand_Cap(n);
83     C(1,2) = -Rand_Cap(n);
84     C(2,1) = -Rand_Cap(n);
85     s = 2*pi;
86     A = G + (s.*C);
87     V = A\b;
88     gain(n) = 20*log10(abs(abs(V(5)))/abs(V(1)));
89 end
90 figure('name','Histogram of gain across a range of capacitors')
91 histogram(gain)
92 xlabel('Gain (dB)')
93 ylabel('Bin count')
94 title('Gain across a range of capacitors','interpreter','latex')
95 grid on;
96 axis auto;
97 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↵ Results','[Part1]Histogram.png'),'png')
98 %% Transient Analysis
99 clc; clear;
100 % Define G,C,b matrix and fill in
101 global G C b
102 NrNodes = 5; % The total number of nodes in the circuit
103 % Define and create simulation parameters
104 Sim_Time = 1;
105 Steps = 1000;
106 delta = Sim_Time/Steps;
107 Time = linspace(0,1,Steps);
108 % Step the frequency of the input
109 Frequency = 1/0.03;
110 Frequency2 = 1/0.3;
111 % Define the three input signals
112 Input_Step = zeros(1,Steps);
113 Input_Sin = zeros(1,Steps);
114 Input_Sin2 = zeros(1,Steps);
115 Input_Guassian = zeros(1,Steps);
116
117 % To make the step input an actual step input
118 Transition_Time = 0.03;
119 Input_Step = double(Time>=Transition_Time);
120

```

```

121 % Fill in sin input with sin values
122 Input_Sin = sin(2*pi*Frequency*Time);
123 Input_Sin2 = sin(2*pi*Frequency2*Time);
124 % Create Guassian signal values
125 GaussianDist = makedist('Normal', 'mu', 0.06, 'sigma', 0.03);
126 GaussianPulse = pdf(GaussianDist, Time);
127 Input_Pulse = (GaussianPulse.*sin(pi*Time))/max(GaussianPulse.*sin(pi*Time));
128
129 % Do trans. sim for step input
130 Vin = Input_Step;
131 for i = 1:Steps
132     % Define G, C, b, for a circuit (do not include additional variables).
133     G = zeros(NrNodes,NrNodes);
134     C = zeros(NrNodes,NrNodes);
135     b = zeros(NrNodes,1);
136     %set up component values
137
138     res(1,2,1)
139     cap(1,2,0.25)
140     res(2,0,2)
141     ind(2,3,0.2)
142     res(3,0,185)
143     vcvs(4,0,3,0,(100/185))
144     res(4,5,0.1)
145     res(5,0,1000)
146     vol(1,0,Vin(i));
147     if i == 1
148         NewVoltage = (G + (C./delta)) \ b;
149     else
150         NewVoltage = (G + (C./delta)) \ (b + (C./delta)*OldVoltage);
151     end
152     Voutput(i) = NewVoltage(5);
153     OldVoltage = NewVoltage;
154 end
155 Xaxis = (-length(Vin)/2:length(Vin)/2-1);
156 figure('Name','Trans Sim Step')
157 subplot(2,1,1)
158 plot(Time,Vin,Time,Voutput)
159 grid on;
160 axis auto;
161 legend('Vin','Vout')
162 xlabel('Time (s)')
163 ylabel('Voltage (V)')
164 title('Step response of circuit','interpreter','latex')
165 subplot(2,1,2)
166 plot(Xaxis, mag2db(abs(fftshift(fft(Vin))))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
167 title('Frequency Spectrum response','interpreter','latex')
168 xlabel('Frequency (Hz)')
169 ylabel('Magnitude (dB)')
170 grid on;
171 axis auto;
172 legend('Vin','Vout')
173 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↵ Results','[Part1]Step input response.png'),'png')
174
175 % Do trans. sim for Sin input
176 Vin = Input_Sin;
177 for i = 1:Steps
178     % Define G, C, b, for a circuit (do not include additional variables).
179     G = zeros(NrNodes,NrNodes);
180     C = zeros(NrNodes,NrNodes);
181     b = zeros(NrNodes,1);
182     %set up component values
183     vol(1,0,Vin(i));

```



```

184     res(1,2,1)
185     cap(1,2,0.25)
186     res(2,0,2)
187     ind(2,3,0.2)
188     res(3,0,185)
189     vcvs(4,0,3,0,(100/185))
190     res(4,5,0.1)
191     res(5,0,1000)
192
193     if i == 1
194         NewVoltage = (G + (C./delta)) \ b;
195     else
196         NewVoltage = (G + (C./delta)) \ (b + (C./delta)*OldVoltage);
197     end
198     Voutput(i) = NewVoltage(5);
199     OldVoltage = NewVoltage;
200 end
201 Xaxis = (-length(Vin)/2:length(Vin)/2-1);
202 figure('Name','Trans Sim Sin')
203 subplot(2,1,1)
204 plot(Time,Vin,Time,Voutput)
205 grid on;
206 axis auto;
207 legend('Vin','Vout')
208 xlabel('Time (s)')
209 ylabel('Voltage (V)')
210 title('Sin response of circuit','interpreter','latex')
211 subplot(2,1,2)
212 plot(Xaxis, mag2db(abs(fftshift(fft(Vin))))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
213 title('Frequency Spectrum response','interpreter','latex')
214 xlabel('Frequency (Hz)')
215 ylabel('Magnitude (dB)')
216 grid on;
217 axis auto;
218 legend('Vin','Vout')
219 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↵ Results','[Part1]Sin1.png'),'png')
220
221 %Impulse response
222 Vin = Input_Pulse;
223 for i = 1:Steps
224     % Define G, C, b, for a circuit (do not include additional variables).
225     G = zeros(NrNodes,NrNodes);
226     C = zeros(NrNodes,NrNodes);
227     b = zeros(NrNodes,1);
228     %set up component values
229
230     res(1,2,1)
231     cap(1,2,0.25)
232     res(2,0,2)
233     ind(2,3,0.2)
234     res(3,0,185)
235     vcvs(4,0,3,0,(100/185))
236     res(4,5,0.1)
237     res(5,0,1000)
238     vol(1,0,Vin(i));
239     if i == 1
240         NewVoltage = (G + (C./delta)) \ b;
241     else
242         NewVoltage = (G + (C./delta)) \ (b + (C./delta)*OldVoltage);
243     end
244     Voutput(i) = NewVoltage(5);
245     OldVoltage = NewVoltage;
246 end

```

```

247 Xaxis = (-length(Vin)/2:length(Vin)/2-1);
248 figure('Name','Trans Sim Impulse')
249 subplot(2,1,1)
250 plot(Time,Vin,Time,Voutput)
251 grid on;
252 axis auto;
253 legend('Vin','Vout')
254 xlabel('Time (s)')
255 ylabel('Voltage (V)')
256 title('Pulse response of circuit','interpreter','latex')
257 subplot(2,1,2)
258 plot(Xaxis, mag2db(abs(fftshift(fft(Vin))))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
259 title('Frequency Spectrum response','interpreter','latex')
260 xlabel('Frequency (Hz)')
261 ylabel('Magnitude (dB)')
262 grid on;
263 axis auto;
264 legend('Vin','Vout')
265 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↵ Results','[Part1]Impluse.png'),'png')
266
267 % Do trans. sim for Sin input`
268 Vin = Input_Sin2;
269 for i = 1:Steps
270     % Define G, C, b, for a circuit (do not include additional variables).
271     G = zeros(NrNodes,NrNodes);
272     C = zeros(NrNodes,NrNodes);
273     b = zeros(NrNodes,1);
274     %set up component values
275     res(1,2,1)
276     cap(1,2,0.25)
277     res(2,0,2)
278     ind(2,3,0.2)
279     res(3,0,185)
280     vcvs(4,0,3,0,(100/185))
281     res(4,5,0.1)
282     res(5,0,1000)
283     vol(1,0,Vin(i));
284     if i == 1
285         NewVoltage = (G + (C./delta)) \ b;
286     else
287         NewVoltage = (G + (C./delta)) \ (b + (C./delta)*OldVoltage);
288     end
289     Voutput(i) = NewVoltage(5);
290     OldVoltage = NewVoltage;
291 end
292 Xaxis = (-length(Vin)/2:length(Vin)/2-1);
293 figure('Name','Trans Sim Sin diff Freq')
294 subplot(2,1,1)
295 plot(Time,Vin,Time,Voutput)
296 grid on;
297 axis auto;
298 legend('Vin','Vout')
299 xlabel('Time (s)')
300 ylabel('Voltage (V)')
301 title('Sin response of circuit','interpreter','latex')
302 subplot(2,1,2)
303 plot(Xaxis, mag2db(abs(fftshift(fft(Vin))))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
304 title('Frequency Spectrum response','interpreter','latex')
305 xlabel('Frequency (Hz)')
306 ylabel('Magnitude (dB)')
307 grid on;
308 axis auto;
309 legend('Vin','Vout')

```

```

310 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪ Results','[Part1]Sin2.png'),'png')
311 %% Circuit with noise
312 clc; clear;
313 % Define G,C,b matrix and fill in
314 global G C b
315 NrNodes = 5; % The total number of nodes in the circuit
316 % Define and create simulation parameters
317 Sim_Time = 1;
318 Steps = 1000;
319 delta = Sim_Time/Steps;
320 Time = linspace(0,1,Steps);
321 % Step the frequency of the input
322 Frequency = 1/0.03;
323
324 Input_Guassian = zeros(1,Steps);
325
326 GaussianDist = makedist('Normal', 'mu', 0.06, 'sigma', 0.03);
327 GuassianPulse = pdf(GaussianDist, Time);
328 Input_Pulse = (GuassianPulse.*sin(pi*Time))/max(GuassianPulse.*sin(pi*Time));
329
330 Vin = Input_Pulse;
331 Vout = zeros(1,Steps);
332 Cn = 0.00001;
333 In = 0.001;
334 h = 1/10e3;
335 W = 0:h:1;
336 CurrNoise = In*randn(1,numel(W));
337 for i = 1:Steps
338     % Define G, C, b, for a circuit (do not include additional variables).
339     G = zeros(NrNodes,NrNodes);
340     C = zeros(NrNodes,NrNodes);
341     b = zeros(NrNodes,1);
342     %set up component values
343     vol(1,0,Vin(i));
344     cur(3,0,CurrNoise(i));
345     res(1,2,1)
346     cap(1,2,0.25)
347     cap(3,0,Cn)
348     res(2,0,2)
349     ind(2,3,0.2)
350     res(3,0,185)
351     res(4,5,0.1)
352     res(5,0,1000)
353     vcvs(4,0,3,0,100/185);
354     if i == 1
355         NewVoltage = (G + (C./delta)) \ b;
356     else
357         NewVoltage = (G + (C./delta)) \ (b + (C./delta)*OldVoltage);
358     end
359     Voutput(i) = NewVoltage(5);
360     OldVoltage = NewVoltage;
361 end
362
363 C
364
365 Xaxis = (-length(Vin)/2:length(Vin)/2-1);
366 figure('Name','Trans Sim Noise')
367 subplot(2,1,1)
368 plot(Time,Vin,Time,Voutput)
369 grid on;
370 axis auto;
371 legend('Vin','Vout')
372 xlabel('Time (s)')

```

```

373 ylabel('Voltage (V)')
374 title('Sin response of circuit','interpreter','latex')
375 subplot(2,1,2)
376 plot(Xaxis, mag2db(abs(fftshift(fft(Vin))))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
377 title('Frequency Spectrum response','interpreter','latex')
378 xlabel('Frequency (Hz)')
379 ylabel('Magnititude (dB)')
380 grid on;
381 axis auto;
382 legend('Vin','Vout')
383 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↪ Results','[Part2]NoisySignal.png'),'png')
384 % Varrying Cn
385
386 Cn = [0.00001 0.0001 0.001 0.01 0.1];
387 for i = 1:numel(Cn)
388     for j = 1:Steps
389         % Define G, C, b, for a circuit (do not include additional variables).
390         G = zeros(NrNodes,NrNodes);
391         C = zeros(NrNodes,NrNodes);
392         b = zeros(NrNodes,1);
393         %set up component values
394         vol(1,0,Vin(j));
395         cur(3,0,CurrNoise(j));
396         res(1,2,1)
397         cap(1,2,0.25)
398         cap(3,0,Cn(i))
399         res(2,0,2)
400         ind(2,3,0.2)
401         res(3,0,185)
402         res(4,5,0.1)
403         res(5,0,1000)
404         vcvs(4,0,3,0,100/185);
405         if j == 1
406             NewVoltage = (G + (C./delta)) \ b;
407         else
408             NewVoltage = (G + (C./delta)) \ (b + (C./delta)*OldVoltage);
409         end
410         Voutput(j) = NewVoltage(5);
411         OldVoltage = NewVoltage;
412     end
413     Xaxis = (-length(Vin)/2:length(Vin)/2-1);
414     sgtitle('Delta Cn')
415     if i == 1
416         figure(21)
417         subplot(5,2,i)
418         plot(Time,Vin,Time,Voutput)
419         grid on;
420         axis auto;
421         legend('Vin','Vout')
422         xlabel('Time (s)')
423         ylabel('Voltage (V)')
424         title(sprintf('Frequency Specetrum Response, C_n = %f',Cn(i)))
425         hold on;
426         subplot(5,2,i+1)
427         plot(Xaxis, mag2db(abs(fftshift(fft(Vin))))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
428         title('Frequency Spectrum response','interpreter','latex')
429         xlabel('Frequency (Hz)')
430         ylabel('Magnititude (dB)')
431         grid on;
432         axis auto;
433         legend('Vin','Vout')
434         hold on;
435     else

```

```

436     figure(21)
437     subplot(5,2,(i*2-1))
438     plot(Time,Vin,Time,Voutput)
439     grid on;
440     axis auto;
441     legend('Vin','Vout')
442     xlabel('Time (s)')
443     ylabel('Voltage (V)')
444     title(sprintf('Frequency Specetrum Response, C_n = %.4f',Cn(i)))
445     hold on;
446     subplot(5,2,i*2)
447     plot(Xaxis, mag2db(abs(fftshift(fft(Vin))))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
448     title('Frequency Spectrum response','interpreter','latex')
449     xlabel('Frequency (Hz)')
450     ylabel('Magnititude (dB)')
451     grid on;
452     axis auto;
453     legend('Vin','Vout')
454     hold on;
455 end
456 end
457 %saveas(gcf,fullfile('D:\School Work\ELEC 4700\My 4700 Code\Assignment 4\Simulation
↵ Results','[Part2]VarryingCn.png'),'png')
458
459 %Varrying Time
460
461 Steps = [10 100 1e3 1e4];
462
463 Cn = 0.00001;
464 for i = 1: numel(Steps)
465     delta = 1/Steps(i);
466     Time = linspace(0,1,Steps(i));
467     GaussianDist = makedist('Normal', 'mu', 0.06, 'sigma', 0.03);
468     GuassianPulse = pdf(GaussianDist, Time);
469     Input_Pulse = (GuassianPulse.*sin(pi*Time))/max(GuassianPulse.*sin(pi*Time));
470     Vin = Input_Pulse;
471     Voutput=zeros(1,Steps(i));
472     for j = 1:Steps(i)
473         % Define G, C, b, for a circuit (do not include additional variables).
474         G = zeros(NrNodes,NrNodes);
475         C = zeros(NrNodes,NrNodes);
476         b = zeros(NrNodes,1);
477         %set up component values
478         vol(1,0,Vin(j));
479         cur(3,0,CurrNoise(j));
480         res(1,2,1)
481         cap(1,2,0.25)
482         cap(3,0,Cn)
483         res(2,0,2)
484         ind(2,3,0.2)
485         res(3,0,185)
486         res(4,5,0.1)
487         res(5,0,1000)
488         vcvs(4,0,3,0,100/185);
489         if j == 1
490             NewVoltage = (G + (C./delta)) \ b;
491         else
492             NewVoltage = (G + (C./delta)) \ (b + (C./delta)*OldVoltage);
493         end
494         Voutput(j) = NewVoltage(5);
495         OldVoltage = NewVoltage;
496     end
497     Xaxis = (-length(Vin)/2:length(Vin)/2-1);
498     sgtitle('Delta t')

```

```

499     if i == 1
500         figure(22)
501         subplot(4,2,i)
502         plot(Time,Vin,Time,Voutput)
503         grid on;
504         axis auto;
505         legend('Vin','Vout')
506         xlabel('Time (s)')
507         ylabel('Voltage (V)')
508         title(sprintf('Frequency Specetrum Response, t = %f',Steps(i)))
509         hold on;
510         subplot(4,2,i+1)
511         plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
512         title('Frequency Spectrum response','interpreter','latex')
513         xlabel('Frequency (Hz)')
514         ylabel('Magnititude (dB)')
515         grid on;
516         axis auto;
517         legend('Vin','Vout')
518         hold on;
519     else
520         figure(22)
521         subplot(4,2,(i*2-1))
522         plot(Time,Vin,Time,Voutput)
523         grid on;
524         axis auto;
525         legend('Vin','Vout')
526         xlabel('Time (s)')
527         ylabel('Voltage (V)')
528         title(sprintf('Frequency Specetrum Response, t = %.4f',Steps(i)))
529         hold on;
530         subplot(4,2,i*2)
531         plot(Xaxis, mag2db(abs(fftshift(fft(Vin)))),Xaxis, mag2db(abs(fftshift(fft(Voutput)))));
532         title('Frequency Spectrum response','interpreter','latex')
533         xlabel('Frequency (Hz)')
534         ylabel('Magnititude (dB)')
535         grid on;
536         axis auto;
537         legend('Vin','Vout')
538         hold on;
539     end
540 end
541 %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

```

542 function res(n1,n2,val)
543 % Adds the stamp of a resistor with a value of "val" (Ohms)
544 % connected between nodes n1 and n2 to the G matrix in
545 % circuit representation.
546 %
547 %           val
548 %   n1 0-----/\ /\ /\----0 n2   where R=val (ohms)
549 %-----
550 global G %define global variable
551
552 if (n1 ~= 0)
553     G(n1,n1) = G(n1,n1) + 1/val;
554 end
555
556 if (n2 ~= 0)
557     G(n2,n2) = G(n2,n2) + 1/val;
558 end
559
560 if (n1 ~= 0) && (n2 ~= 0)
561     G(n1,n2) = G(n1,n2) - 1/val;
562     G(n2,n1) = G(n2,n1) - 1/val;
563 end
564
565 end %func

```

Source Code 2: MNA Stamps - Resistor

4.2.2 Capacitor Stamp

```

566 function cap(n1,n2,val)
567 % Adds the stamp of a capacitor with a value of "val"
568 % (Farads) connected between nodes n1 and n2 to the
569 % C matrix in circuit representation.
570 %
571 %           val
572 %   n1 0---//---0 n2   where C= val (Farads)
573 %-----
574 global C %define global variable
575
576 if (n1 ~= 0)
577     C(n1,n1) = C(n1,n1) + val;
578 end
579
580 if (n2 ~= 0)
581     C(n2,n2) = C(n2,n2) + val;
582
583 if (n1 ~= 0) && (n2 ~= 0)
584     C(n1,n2) = C(n1,n2) - val;
585     C(n2,n1) = C(n2,n1) - val;
586 end
587
588 end %func

```

Source Code 3: MNA Stamps - Capacitor

4.2.3 Inductor Stamp

```

589 function ind(n1, n2, val)
590 global G C b %define global variables
591 d = size(G,1); % current size of the MNA
592 xr = d+1; % new (extra) row/column
593
594 % Using an index bigger than the current size, Matlab automatically
595 ... increases the size of the matrix:
596
597 G(xr,xr) = 0; % add new row/column
598 C(xr,xr) = 0;
599 b(xr) = 0; % add new row
600
601 if (n1 ~= 0)
602     G(n1,xr) = 1;
603     G(xr,n1) = 1;
604 end
605 if (n2 ~= 0)
606     G(n2,xr) = -1;
607     G(xr,n2) = -1;
608 end
609
610 C(xr,xr) = - val;
611
612 end

```

Source Code 4: MNA Stamps - Inductor

4.2.4 Voltage Source Stamp

```

613 function vol(n1,n2,val)
614 % Adds the stamp of an independent voltage source with a value
615 % of "val" (Volts) connected between nodes n1 and n2 to the
616 % matrices in circuit representation.
617 %
618 %
619 %           val
620 %           /  \
621 %  n1 0-----(+  -)-----0 n2   where Vsrc= val (volts)
622 %           \  /
623 %           Isrc ---->
624 %-----
625
626 global G b C % define global variables
627
628 d = size(G,1); % current size of the MNA
629 xr = d+1; % new (extra) row/column
630
631 % Using an index bigger than the current size, Matlab
632 ...automatically increases the size of the matrix:
633
634 G(xr,xr) = 0; % add new row/column
635 C(xr,xr) = 0;
636
637 if (n1 ~= 0)
638     G(n1,xr) = 1;
639     G(xr,n1) = 1;
640 end
641
642 if (n2 ~= 0)

```



```

642     G(n2,xr) = -1;
643     G(xr,n2) = -1;
644 end
645 b(xr) = val;
646
647 end %func

```

Source Code 5: MNA Stamps - Voltage Source

4.2.5 Current Source Stamp

```

648 function cur(n1,n2,val)
649 % Adds the stamp of an independent current source with a
650 % value of "val" (Amperes) connected between nodes n1 and
651 % n2 to the source vector b in circuit representation.
652 %
653 %
654 %           val
655 %           /  \
656 %  n1 0-----(->)-0 n2   where J=val (Amperes)
657 %           \  /
658 %
659 %  n1: The node at the tail of the current arrow!
660 %  n2: " " " " head " " " " " !
661 %  val: The value of the current source (Amp)
662 %-----
663 global b %define global variable
664
665 if (n1 ~= 0)
666     b(n1,1) = val;
667 end
668
669 if (n2 ~= 0)
670     b(n2,1) = -val;
671 end
672 end %func

```

Source Code 6: MNA Stamps - Current Source

4.2.6 Voltage Controlled Voltage Source Stamp

```

673 function vcvs(nd1,nd2,ni1,ni2,val)
674 % Adds the stamp of a dependent voltage-controlled
675 % voltage-source(VCVS)to the matrices in circuit
676 % representation.
677 %
678 %  ni1 0-----0           /-----o nd1
679 %           |
680 %           /+ \
681 %           / /  \      Vnd1-Vnd2 = val*(Vni1-Vni2)
682 %           / \  /
683 %           V  \-/
684 %           |
685 %  ni2 0-----0           /-----o nd2
686 %
687 % (1) "nd1 & nd2" are the nodes across the dependent
688 %       voltage source.
689 % (2) "ni1 & ni2" are the nodes corresponding to the

```

```

690 %               controller voltage
691 %
692 %   nd1: (+) node   \
693 %   nd2: (-) node   /----->   Vnd1-Vnd2 = val*(Vni1-Vni2)
694 %   ni1: (+) node   /
695 %   ni2: (-) node   /
696 %-----
697 global G C b   %define global variables
698
699 d = size(G,1); % current size of the MNA
700 xr = d+1;      % new (extra) row/column
701
702 % Using an index bigger than the current size, Matlab automatically
703 ... increases the size of the matrix:
704 G(xr,xr) = 0; % add new row/column
705 C(xr,xr) = 0;
706 b(xr) = 0;    % add new row
707
708 if (nd1 ~= 0)
709     G(nd1,xr) = 1;
710     G(xr,nd1) = 1;
711 end
712 if (nd2 ~= 0)
713     G(nd2,xr) = -1;
714     G(xr,nd2) = -1;
715 end
716
717 if (ni1 ~= 0)
718     G(xr,ni1) = G(xr,ni1)-val;
719 end
720 if (ni2 ~= 0)
721     G(xr,ni2) = G(xr,ni2)+val;
722 end
723 end %func

```

Source Code 7: MNA Stamps - Voltage Controlled Voltage Source