ELEC 4700

Assignment 4

Circuit Modelling

Brandon Schelhaas
101036851
April 11th, 2021
Prof. Tom Smy

Parts 1 and 2: Resistor Simulation

The purpose of this part was to use Assignment 3's solution to implement a voltage sweep, and measure the current to find the resistance with a fixed bottle-neck width. The value of this resistance could then be used in the circuit simulation. This procedure provides the opportunity to use one simulation's output as an input to another simulation.

While the original intention was to use Assignment 3's solution, the previous implementation was very slow very every simulation. First, even with a singular input voltage, approximately 250 electrons within the region, and 150 timesteps (which allowed the current to settle to a near-constant value which included scattering effects), each input voltage took approximately three minutes to fully simulate. Therefore, for an input voltage sweep, the simulation could take hours to fully complete before a resistance value was obtained.

Due to this, Assignment 2's solution of using the finite different method (FDM) was used. Recall Assignment 2 applied a voltage to the left boundary of a region (as in Assignment 3), then FDM solved for the voltage distribution within the region, areas within the region were assigned a conductivity, and then the electric field and current were calculated. Since this solution primarily used matrix-math, the input voltage sweep took significantly less time to complete.

To begin, the solution from Assignment 2 was copied into Assignment 4's solution. This code left off with solving the x- and y- components of the current with the region for each mesh element/node. This current can then be used to find the effective resistance of the region. This can likely be done with many different methods, but the method implemented in this Assignment was that the region was split into five different regions to help find the effective resistance of the overall region:

1. The top (first) box;
2. The bottom (second) box;
3. The left side of the region;
4. The right side of the region;
5. The middle of the region (between the boxes and the sides);

This method was used to best estimate the resistance because each region had different conductivities and currents. Due to the larger conductivity of the left and right regions, the effective resistance is small. Conversely, due to the smaller conductivity of the boxes, the effective resistance is large. Finally, the middle region is forced to have a large current and a large resistance, due to its environment. Therefore, separating these regions can help to find an effective resistance for each region, rather than averaging over the overall region.

This behaviour was implemented by creating five current matrices of the size of the region. Then, after solving the currents for the overall region, these currents were mapped into the five region matrices, according to their respective locations. Then, to calculate the effective resistance, the voltage of the left and right boundaries (high and low voltage) of the separated regions were averaged, and ultimately used to find calculate the resistance with Ohm's law. After each region's resistance value was calculated, the effective resistance was calculated with the following relation:

$$R_{eq} = R_{left} + \left(\frac{1}{\frac{1}{R_{box1}} + \frac{1}{R_{middle}} + \frac{1}{R_{box2}}} \right) + R_{right}$$

This is valid because the left and right regions are in series with the parallel combination of the top/bottom box and the middle region. With the effective resistance measured, the final calculation was to find the overall current, calculated by the input voltage divided by the effective resistance. This procedure was repeated for the entire voltage sweep (0.1V-10V, in steps of 100mV) to measure the current vs. each input voltage, and lastly, a linear fit was completed with *polyfit().* The current vs. voltage results are shown in Figure 1.

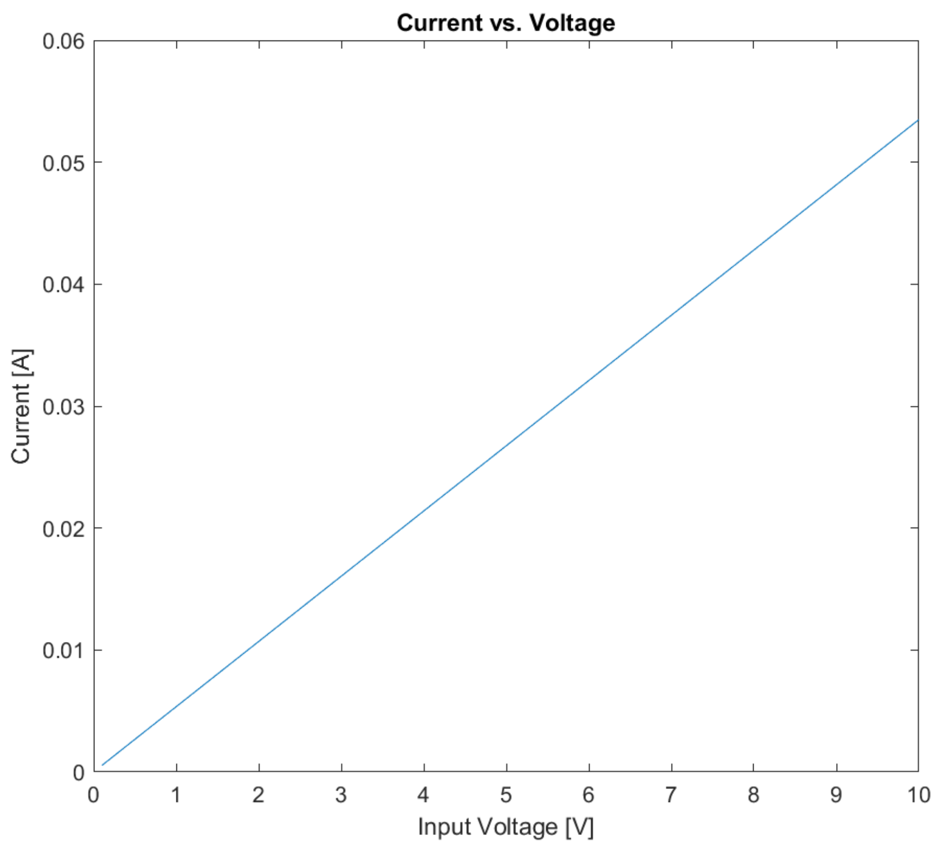

Figure 1: Current vs. Input Voltage

As expected, the results were linear, but also because it was calculated by simple division (which is linear anyways).

Regardless, *polyfit()* was then used to find the linear function that fit the current vs. voltage relationship. Its results were:

linearFit = [0.00534975664713800 3.84519501159861e-18]

This result meant that $I = 0.00534975664713800V + 3.84519501159861\text{e} - 18$. However, $I = \frac{V}{R}$, and therefore, R is the inverse of the first index of linearFit. Therefore, the fitted resistance was:

$$R_{fitted} = \frac{1}{0.00534975664713800} = \mathbf{186.9244\ \Omega}$$

The next step was then to use this value for R3 in the following parts.

Part 3: MNA PA

The purpose of this part was to formally comment on the modified nodal analysis (MNA) PA, but using the previously-fitted resistance from Parts 1 and 2 for R3.

The start of the MNA PA was to take the provided circuit, write the KCL expressions, and create the C and G matrices. The equations were defined as follows, though this is just one version (dependent on which direction is defined as positive/negative):

```
% KCL Equations:
% (1) (v1-v2)/R1 + sC1(v1-v2) + Iin = 0
% (2) (v2-v1)/R1 + sC1(v2-v1) + v2/r2 + IL = 0
% (3) v3/r3 - IL = 0
% (4) (v4-v5)/r4 + Ix = 0
% (5) (v5-v4)/r4 + v5/ro = 0
% (6) v2-v3 = L (dIL/dt) --> -sL + v2-v3 = 0
% (7) v4 - (alpha/r3)*v3 = 0     (create a VCVS)
% (8) v1 = Vin
```

Using these equations, the C and G matrices could be created by using the coefficients of the different voltages/nodes. The C matrix will be constructed with all of the time derivative coefficients, whereas the G matrix will be constructed with all of the normal-time voltages (voltage across a resistor, for example). However, for the PA and this Assignment, stamp functions were created to construct the C, G, and F matrices. This was done so that any circuit could be constructed, rather than just satisfying one circuit, but it is important to note that the stamp functions still take KCL into account. After using the stamps to create the matrices, the C and G matrices were set as follows:

```
G =

    1.0000   -1.0000         0         0         0         0         0    1.0000
   -1.0000    1.5000         0         0         0    1.0000         0         0
         0         0    0.0053         0         0   -1.0000         0         0
         0         0         0   10.0000  -10.0000         0    1.0000         0
         0         0         0  -10.0000   10.0010         0         0         0
         0    1.0000   -1.0000         0         0         0         0         0
         0         0   -0.5350    1.0000         0         0         0         0
    1.0000         0         0         0         0         0         0         0


C =

    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   -0.2000         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
```

Figure 2: C and G matrices for Part 3, including Fitted Resistance for R3

While not explicitly shown, the stamps also created the F matrix which is the matrix that holds any ideal sources. This step means that the circuit is now ready for simulation.

The first simulation was the DC simulation. For this simulation, the input voltage was swept from -10V to 10V to be able to plot the voltages at node 3 (controlling the dependent source) and the output. Since $\omega = 0$ for DC, $s = 0$, and therefore, the C-matrix is not used in the calculation. The solution, for each input voltage, can be found with left division:

$$V = G\backslash F$$

Since V is a vector of all the unknowns, the nodal voltages had to be extracted (vout = V(5), v3 = V(3), in this case), and the results were then plotted, shown in Figure 3.

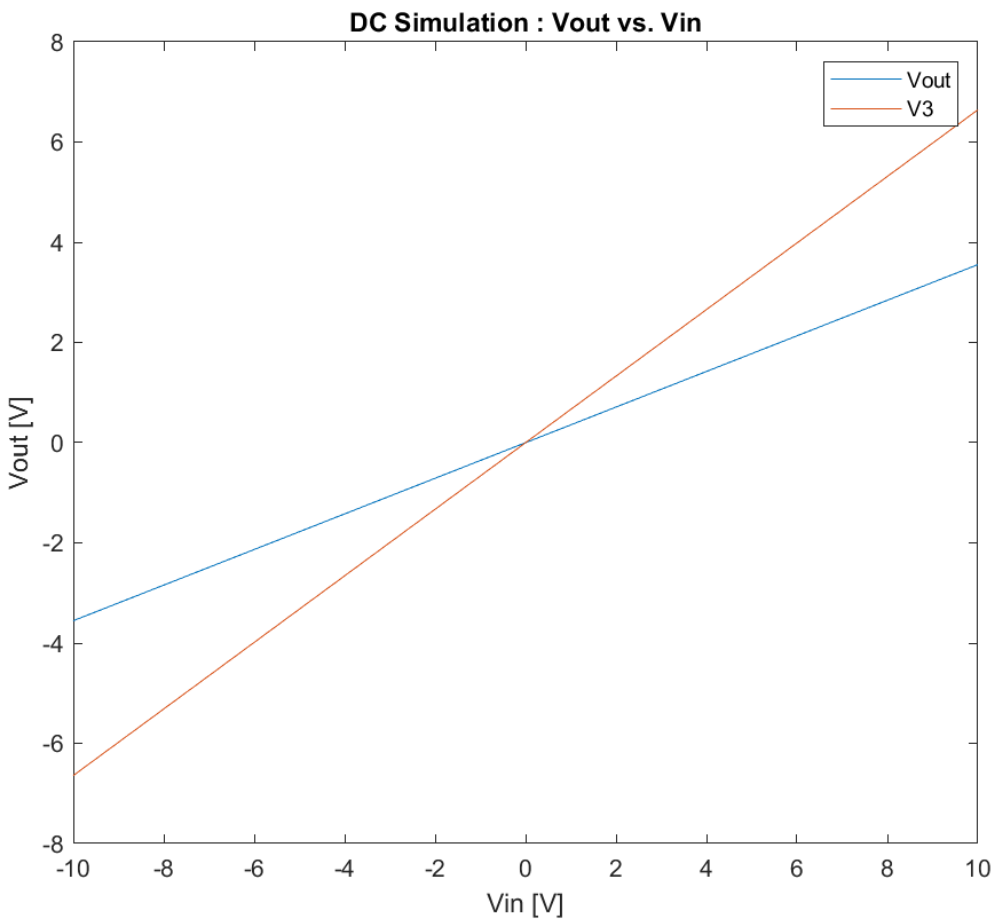

Figure 3: DC Sweep with Fitted Resistance

As can be seen in this result, the output voltage grew with the input voltage, as expected. However, due to using the fitted resistance, this circuit no longer provides the same level of gain as it did in the MNA PA. Rather, the V3 voltage increased, and the output voltage decreased. This is obviously because a different resistance value was used.

The second simulation to complete was an AC sweep. This sweep provides the gain vs. frequency results. The setup involved creating a frequency vector (arbitrarily chosen to be 1 to 50 Hz, in this case), and then creating an ω vector by multiplying each frequency by 2π. Once completed, the nodal voltages could be solved with the following equation:

$$V = (G + j\omega C)\backslash F$$

This is because the solution is frequency-dependent, and thus, the C matrix is now required. Once again, the V vector contained all of the unknowns, so the nodal voltages had to be extracted each frequency step. Once completed, the results were plotted, shown in Figure 4.

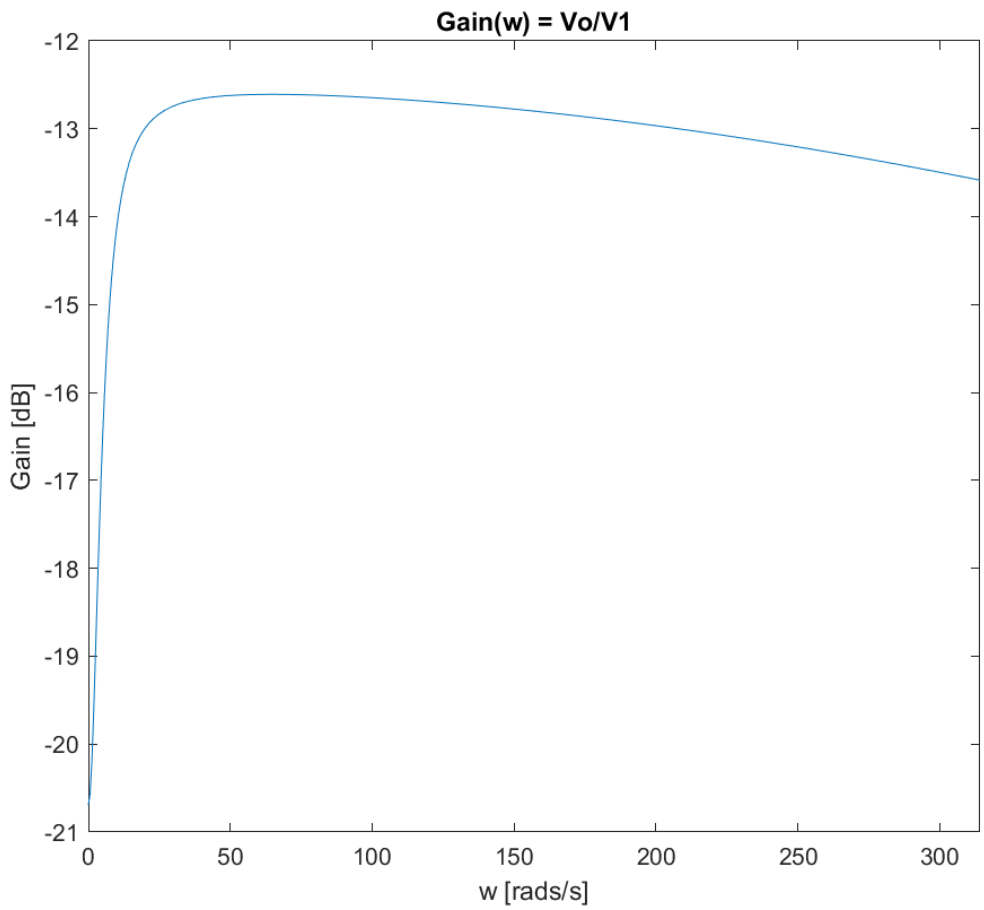

Figure 4: AC Sweep with Fitted Resistance

As can be seen in this result, the output voltage was attenuated through the frequency sweep. In addition to this, it is clear this circuit has a bandpass characteristic, but in this case, the passband still attenuates the input.

The final simulation was to simulate random perturbations on C. This is especially helpful as real capacitance values can vary from the nominal value by quite a large margin, depending on the precision of the manufacturing. To do so, a normal distribution, with $\sigma = .05$ was created and used to vary the effective capacitance value. Then, the gain could be re-simulated, at a constant frequency of $\omega = \pi$, to see the effect of a varying capacitance. The voltage calculations were the same as before, and the results were plotted, shown in Figure 5.

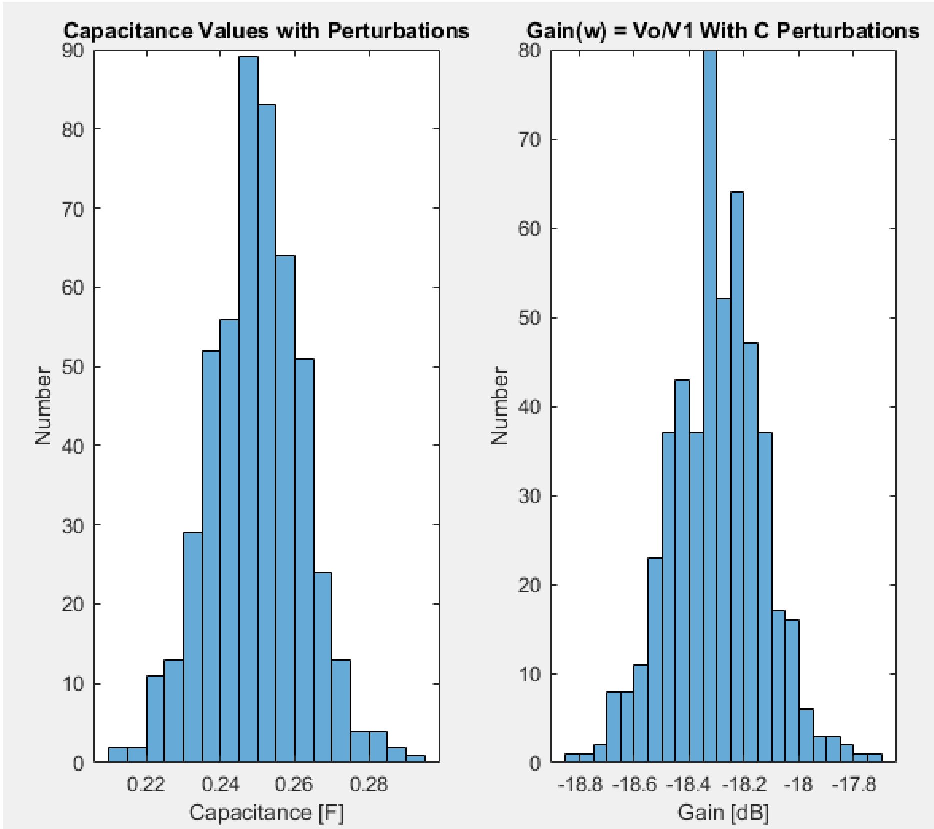

Figure 5: Gain vs. Capacitance

As can be seen in this result, small variations in capacitance tend to have a significant effect on the gain. This is evident where small variations on the capacitance can change the gain by approximately $\pm 0.5\ dB$. While this may not be very significant in this specific case, this result shows a more general conclusion, where, depending on the application, designers should take variations of capacitance into account if bandwidth or gain are of major concern.

Part 4: Transient Simulation

The purpose of this part was to introduce transient simulations for this circuit. This allows any input to be created at the software-level and be used to model the circuit's response to that given input. This can be advantageous over the hardware solution as any response can be easily created and tested.

However, there may be certain limitations to this circuit. By inspection, the previous results have shown a "bandpass-like" characteristic where low and high frequencies will be attenuated, but that, in theory, this is also an amplifier circuit (due to the current dependent voltage source). However, attenuation depends on the input frequency. This bandpass characteristic is because of the series inductor and capacitor. The capacitor will block DC, but there is a DC response due to its parallel resistance. The inductor will block high frequencies, but pass DC. The combination of the two will both block low and high frequencies, but pass the mid-band frequencies – thereby giving a bandpass response. Therefore, it is important to know that this characteristic is in place when determining capacitance and inductance values, and similarly, input frequencies within the test inputs.

The first part to complete is the setup, involving the simulation setup and the input waveforms setup. For the simulation setup, the intention was to simulate for 1s and 1000 timesteps (which implies $dt = 0.001$). Therefore, a time vector was created with these characteristics, specifically using the *linspace()* function. Next, the intended input signals were a step voltage, sine wave, and a Gaussian pulse.

The step voltage was created by creating a voltage vector where any time indices less than 0.03s were set to 0, and any time indices greater than or equal to 0.03s were set to 1. This creates an input step that transitions to 1 at 0.03s. Next, the sinewave was simply created with built-in MATLAB behaviour, by inputting the time vector into the *sin()* function, with a frequency of $\frac{1}{0.03} = 33.33\ Hz$. Finally, the Gaussian pulse was created by:

1. Making a normal distribution, with $\mu = 0.06$ and $\sigma = 0.03$ (with the *makedist()* function)
2. Creating a probability distribution function (PDF) from this distribution and the time vector
3. Multiplying this PDF with a sine wave of frequency $\omega = \pi$ and the time vector
4. Normalizing this product to give a magnitude of 1

The functions were plotted and shown in Figure 6.

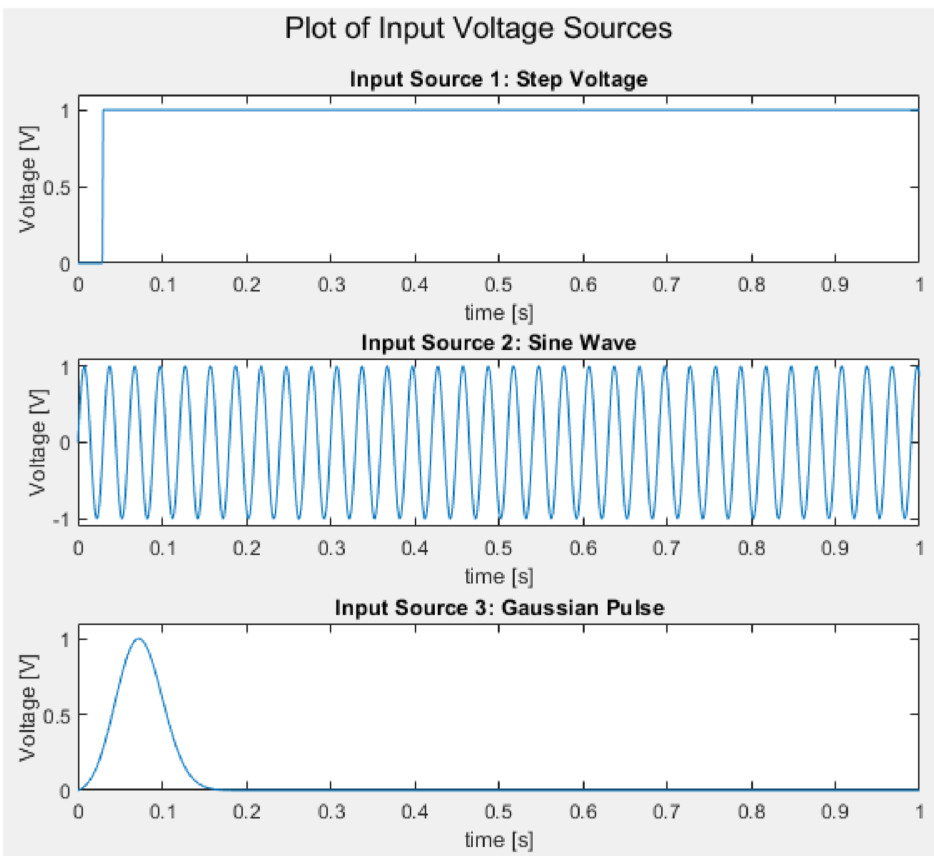

Figure 6: Input Sources

The next step was to find the equation required to solve for the unknown variables of the circuit. This must be derived from the MNA formula:

$$C\frac{dV}{dt} + GV = F$$

Using Backward Euler FDM, the result can be found as follows:

$$C\left(\frac{V_i - V_{i-1}}{dt}\right) + GV = F$$

$$\left(\frac{C}{dt} + G\right)V_i = \left(F + \frac{C}{dt}V_{i-1}\right)$$

$$V_i = \left(\frac{C}{dt} + G\right) \backslash \left(F + \frac{C}{dt}V_{i-1}\right)$$

Therefore, this relation shows that the new voltage requires the old voltage (which implicitly accommodates the derivative). Note "\" is the left-divide operator in MATLAB.

The transient simulation was completed by using a loop to step through each voltage. Two voltage vectors were used: one to calculate the new unknowns, and one to store the previously-calculated unknowns. At the first timestep, there are no previous values, but they were assumed to be 0 because there was no previous excitation to the circuit. Then, the loop completed the transient simulation by looping through each input signal, where the F vector was changed to hold the current value of the input signal at each timestep.

The results were gathered, and shown in Figures 7, 8, and 9, along with their frequency spectra.

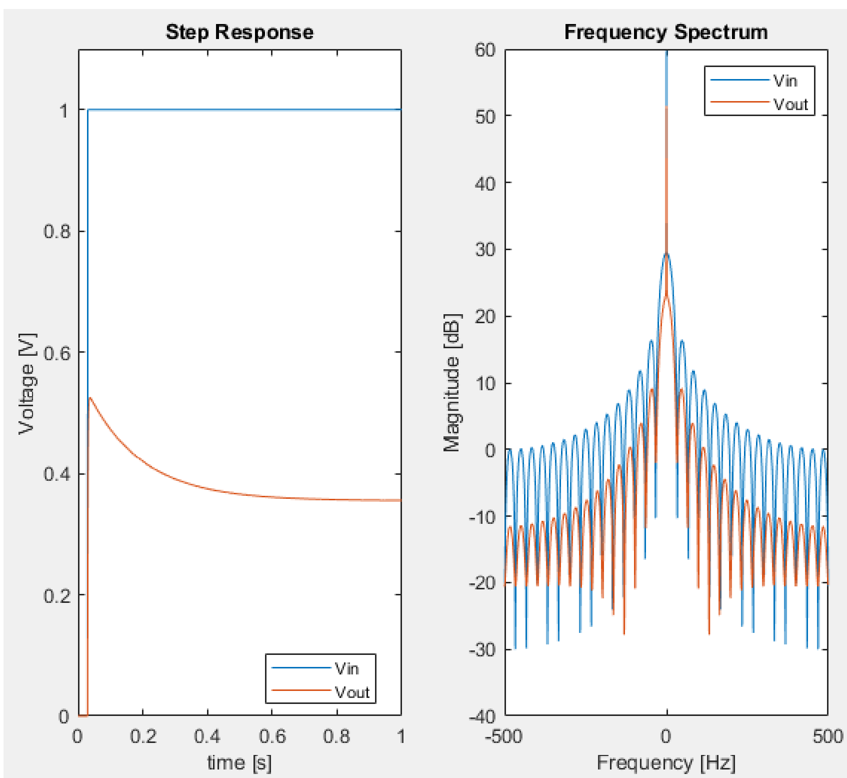

Figure 7: Transient Step Response

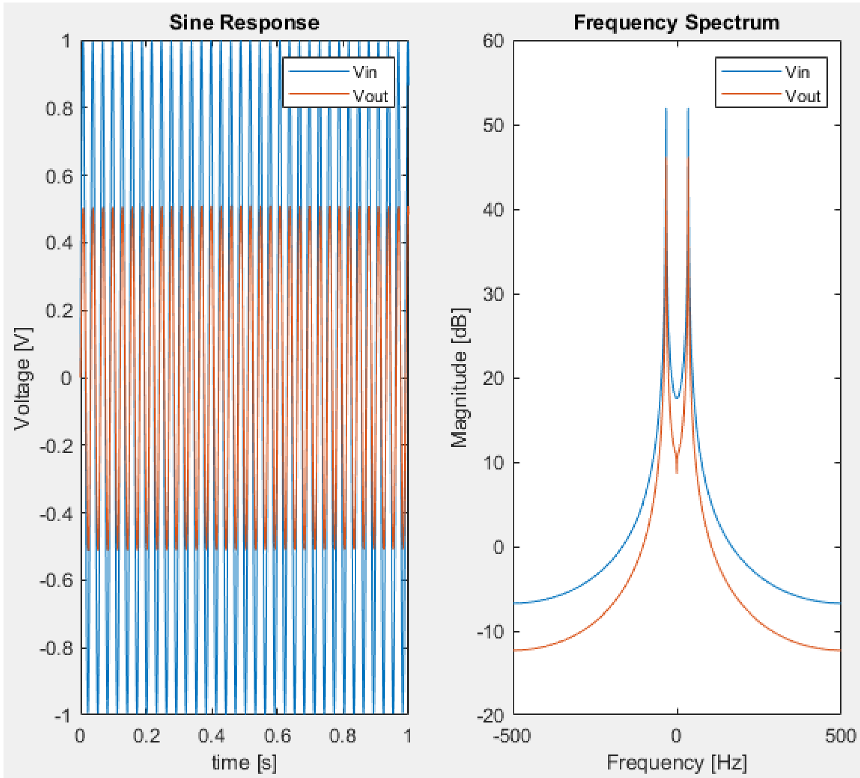

Figure 8: Sine Response

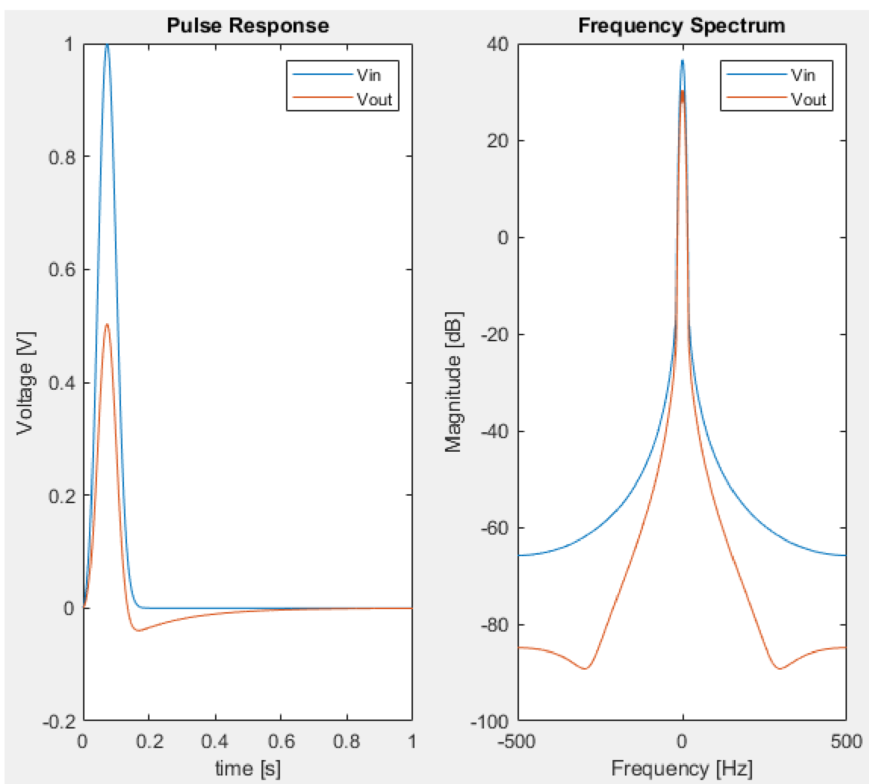

Figure 9: Gaussian Pulse Response

All of these results show the same results as before, where the output has some attenuation compared to the input. However, each response can show some general characteristics of the system:

- The step response shows a slight overshoot in the response before it dies to a constant value.
- The sine wave is just attenuated, which could have been expected due to the frequency response.
- The Gaussian pulse is pulled lower than 0V after the pulse. This would imply a reverse in current in a real circuit.

Therefore, this kind of simulation is very helpful as it can show system behaviour before even creating or manufacturing a costly circuit with the potential of damaging it.

In addition, the frequency spectra results are helpful to visualize the energy at each frequency. As expected, the step response creates a sinc function response, the sine wave has two characteristic spikes at $\pm f_{sine}$, and the Gaussian pulse has a similar shape in the frequency domain as well.

Two more simulations were run to investigate the behaviour of sine waves with different frequencies. The first test was at 8Hz, to achieve approximately max gain, and the second test was at 100 Hz to visualize further attenuation. The results are shown in Figures 10 and 11.

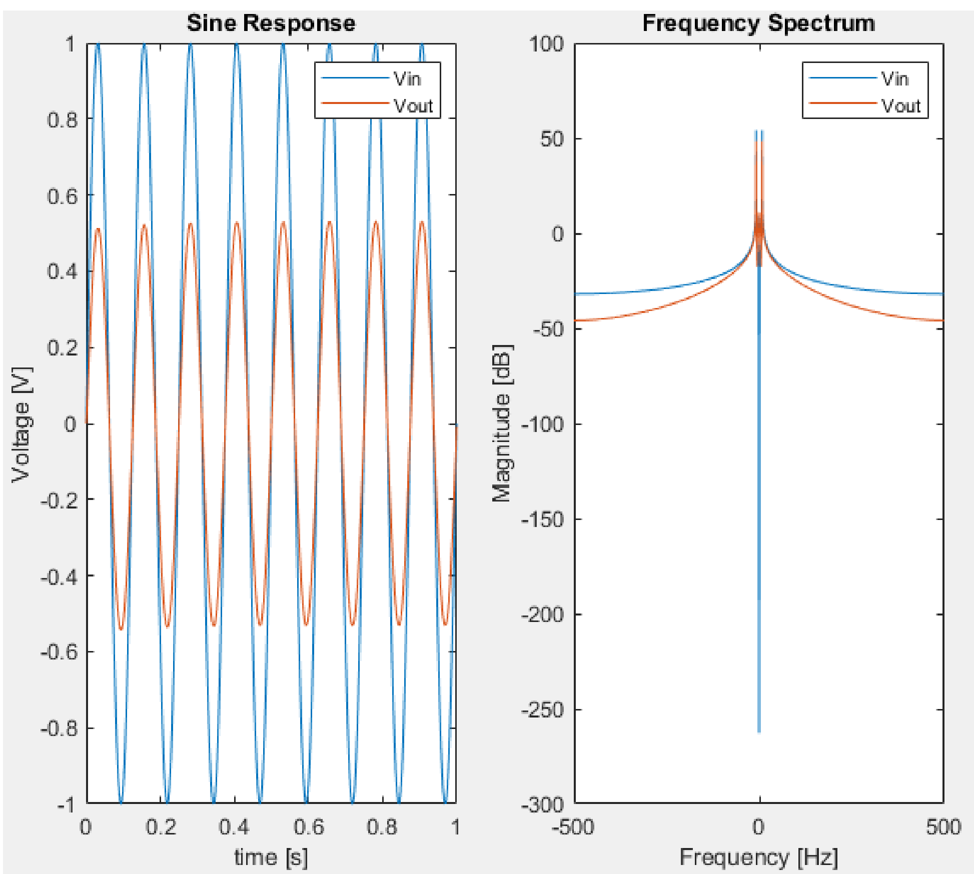

Figure 10: Sine Response, f = 8Hz

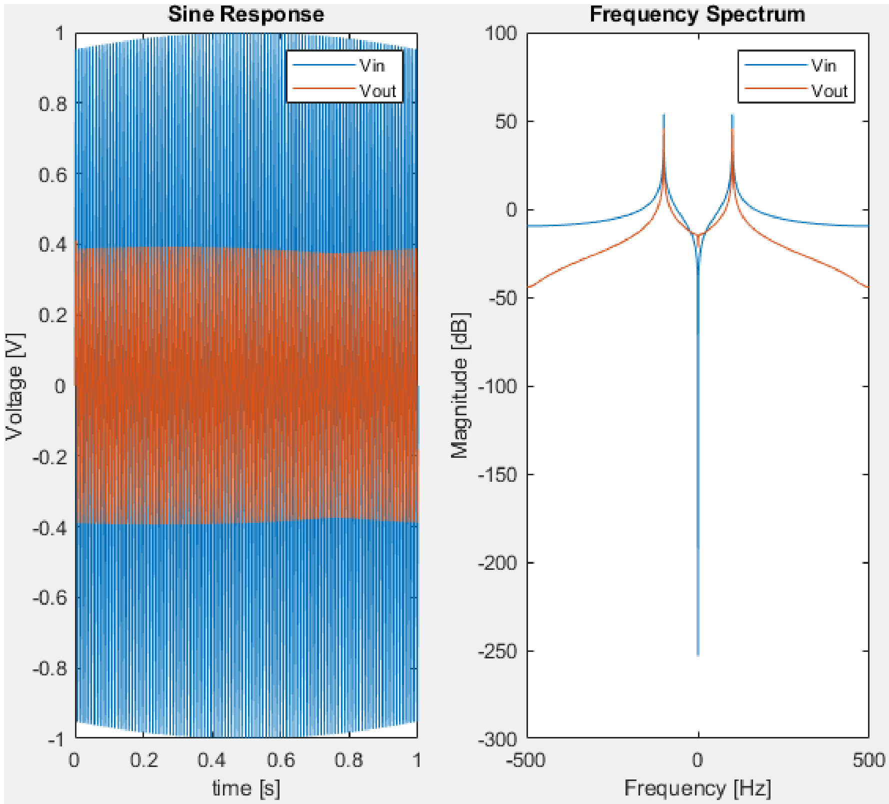

Figure 11: Sine Response, f = 100Hz

These results are consistent with expectations. The output of the 8Hz sine wave is marginally larger than the output of the 100Hz sine wave, ultimately due to the frequency response and its attenuation at higher frequencies. In addition, the spikes in the frequency spectra are very close for the 8Hz sine wave, while further separated for the 100Hz sinewave (as expected). The last notable conclusion is that, due to a 1ms time step, the 100 Hz sinewave is not accurately sampled which is why the amplitude is not constant. Decreasing the timestep will allow for more resolution, and ultimately fix the varying peak amplitude.

Note that all previous results depend on the timestep, but the effects of the timestep will be investigated further in the assignment and the report.

Part 5: Circuit with Noise

The purpose of this part was to add thermal noise associated with R3 to include in the modelling. This can be modelled with a current source and capacitor in parallel with R3, where the current source is random (modelling the noise), and the capacitor help to limit the bandwidth of the noise.

The current source was implemented by creating a new stamp that is used to create the current source itself. Since it is an 'ideal' source, it is simply implemented by changing the F-matrix to include this as a source. The capacitor was implemented by re-using the same capacitor stamp. The updated C-matrix is shown in Figure 12.

```
C =

    0.2500   -0.2500         0         0         0         0         0         0
   -0.2500    0.2500         0         0         0         0         0         0
         0         0    0.0000         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         0
         0         0         0         0         0         0         0         0
```

Figure 12: Updated C-Matrix

Note that the 0.0000 is actually the new capacitance value of 0.00001, there just is not enough precision to show the real value.

With the noise current source added into the circuit as well (using the stamping function), the transient simulation was re-run with the current source sent to a constant value of 0.001. This result is shown in Figure 13.

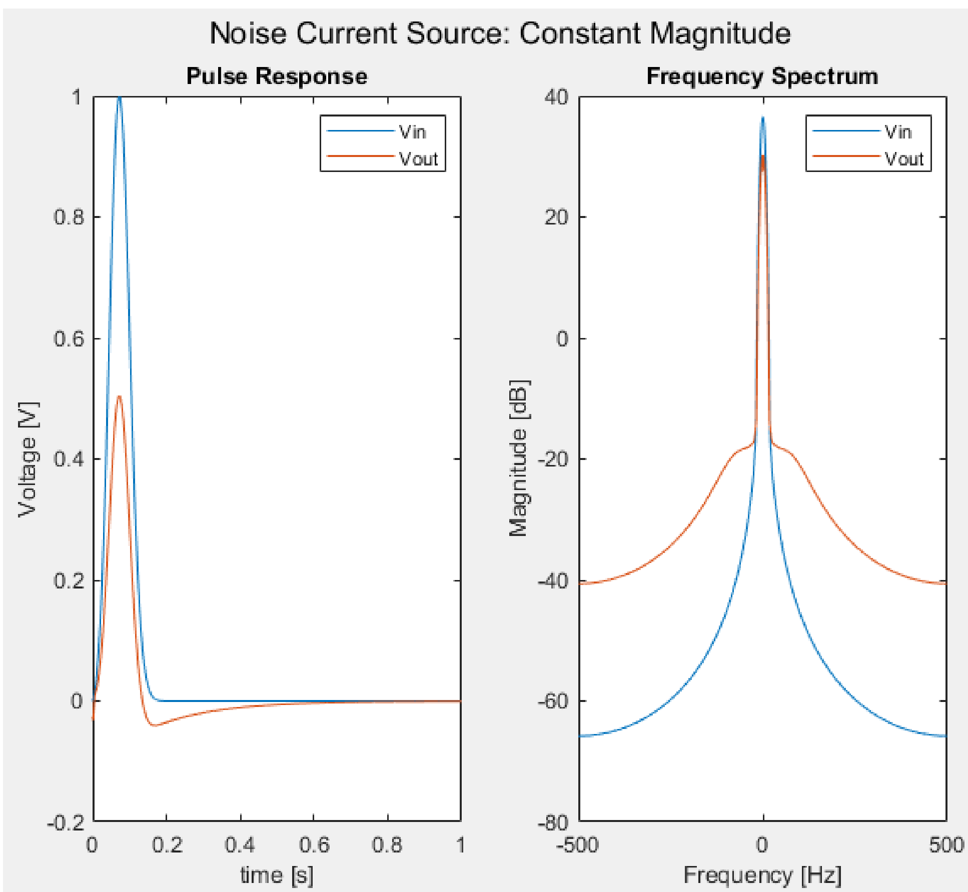

Figure 13: Results for Constant Noise Current, Magnitude = 0.001

As can be seen in this result, there is no noise effects. Currently the source just adds more current to the circuit.

To actually create the noise values for the current source, the default value of $I_n = 0.001$ was used to scale normally-distributed random numbers with length equal to the length of the time vector. Then, for each timestep, the specific index was used to take the random current, and change the F-matrix to use this current value. A sample distribution of values (since they are random) is shown in Figure 14.

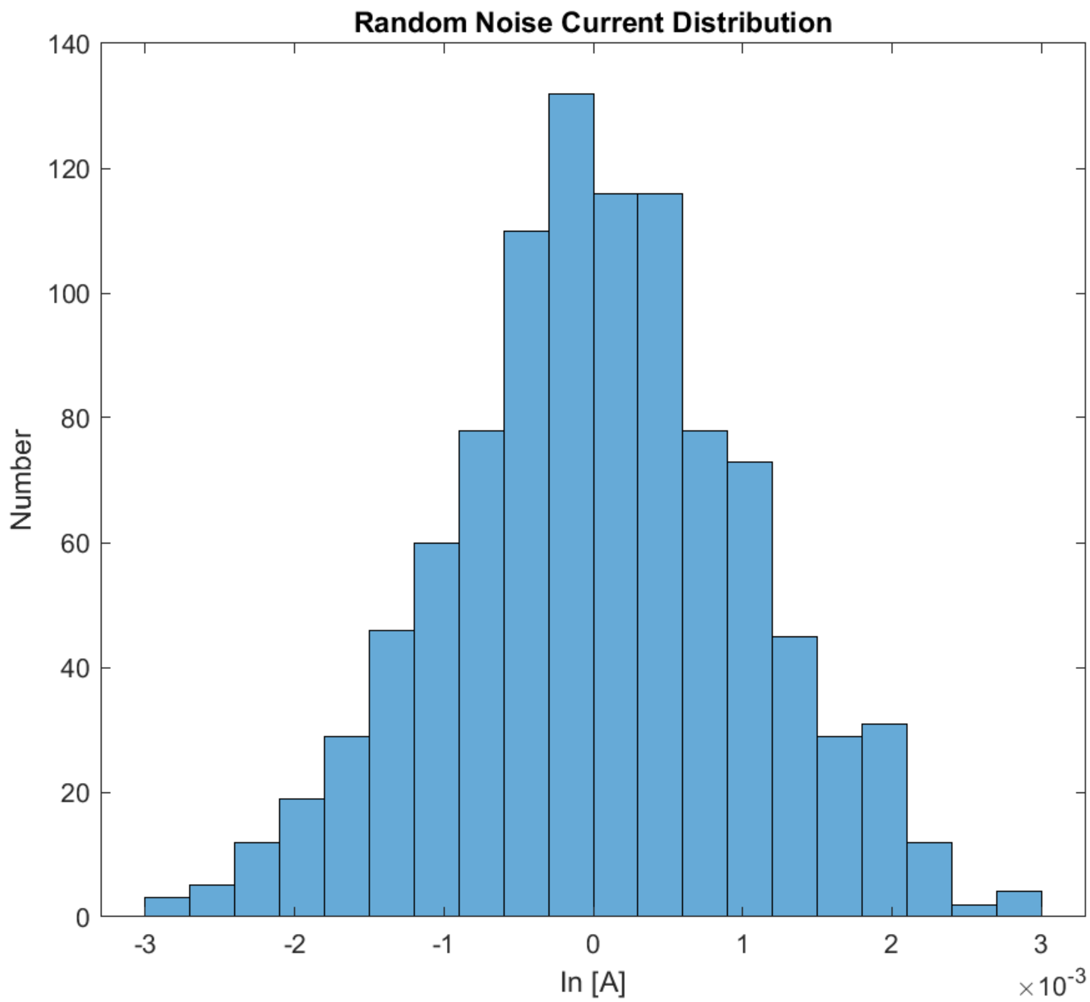

Figure 14: Noise Current Values Distribution

The transient simulation was then re-run with this set of noise current values, and the results is shown in Figure 15.

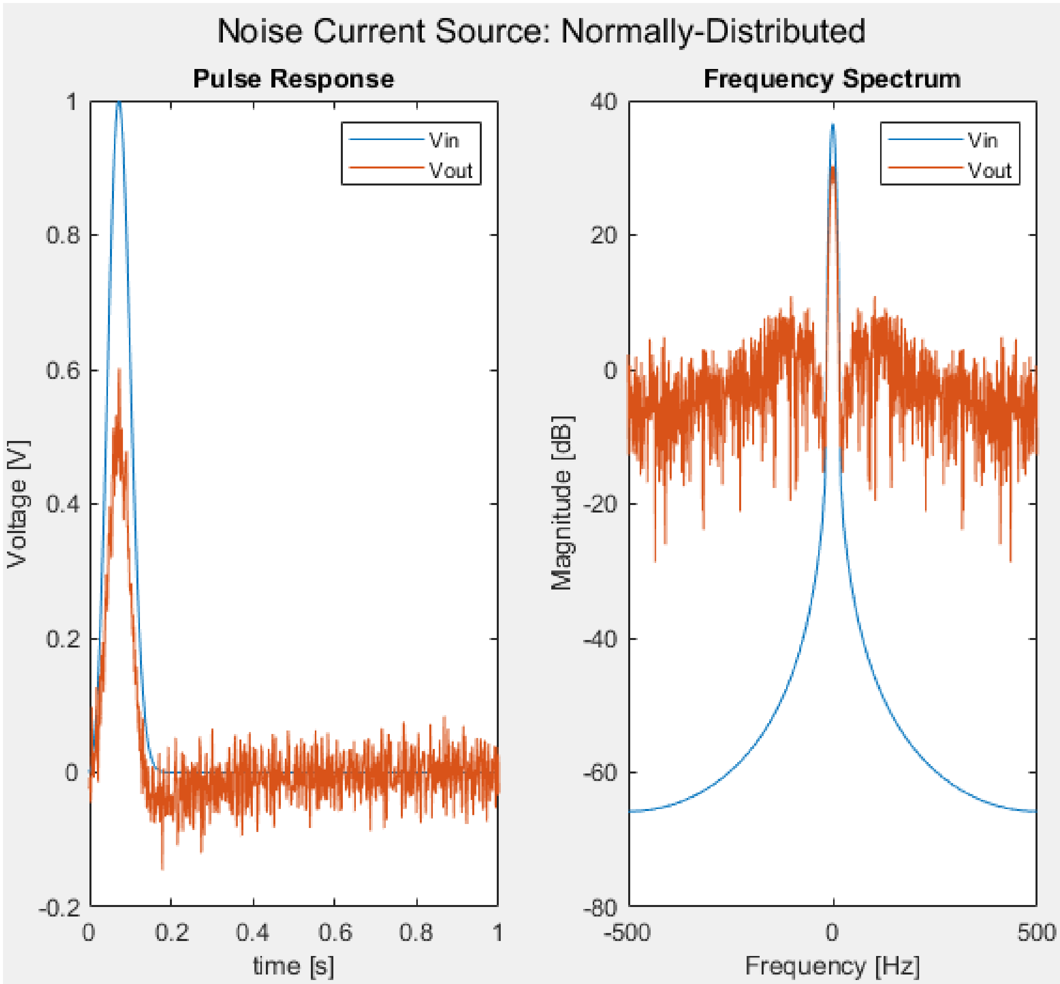

Figure 15: Results for Random Noise Current

As can be seen in this result, the random values present the effect of noise in the circuit. Due to the natural attenuation of the circuit, the output is only slightly above the noise floor. However, both the time-domain and frequency spectrum clearly show the noise and noise floor of the newly-added noise source.

With this noise source operating correctly, the next step was to vary the capacitance to observe how its bandwidth as the capacitance is varied. For simplicity, this test was completed by re-running the same code and changing the Cn capacitance value between simulations. For these tests, the capacitance values were changed from default: 0.00001 to 0.1, 0.01, and 0.001. These results are shown in Figure 16.

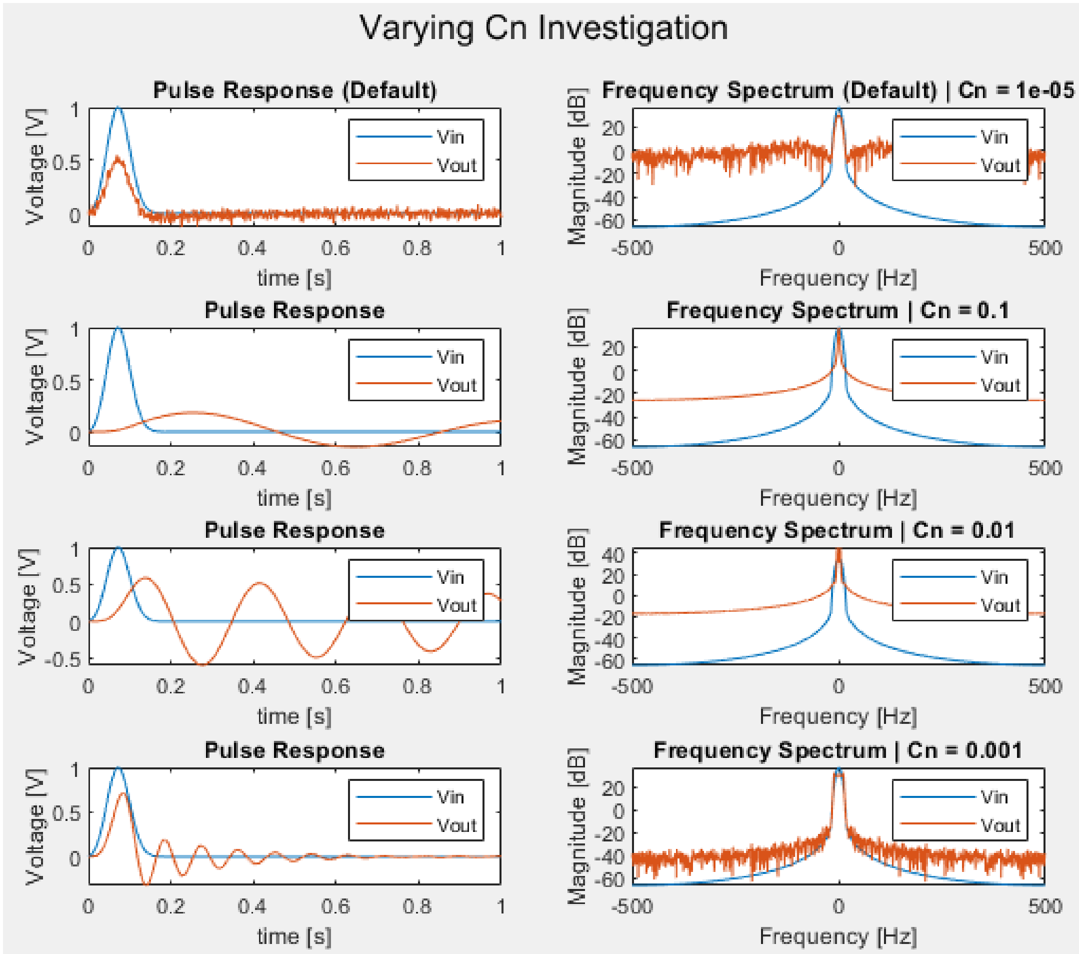

Figure 16: Cn Investigation

These results clearly show how Cn limits the bandwidth as larger capacitances remove the high frequency noise (as anticipated). The first case shows the default value. Here, the capacitance is small enough to pass the high frequency noise, and in doing so, noise is seen both in the time-domain and frequency-domain responses. The second case shows the largest capacitance. Here, the capacitance is large and so it blocks the high frequency. This is shown in the 'sinusoidal' behaviour on the input since the high frequency components are removed (so the Gaussian pulse is not re-created). This is also evident in the frequency spectrum where there is no noise floor. The third case decreases the capacitance to allow more high frequencies to pass to the output. This is evident both in the faster transitions in the output ('higher frequency' sinusoidal wave), and slight noise perturbations in the frequency spectrum. The final case further decreases the capacitance to allow more high frequencies to pass to the output, allowing the Gaussian pulse to be nearly re-created, and a noticeable noise floor in the frequency domain.

Therefore, as Cn increases, the bandwidth is decreased, as anticipated by electronics theory.

The final investigation of this part was to change the timestep and observe how it changes the overall simulation results. Similar to the capacitance, this was accomplished by re-using the same code and changing the value of the timesteps in between simulations. This result is shown in Figure 17.

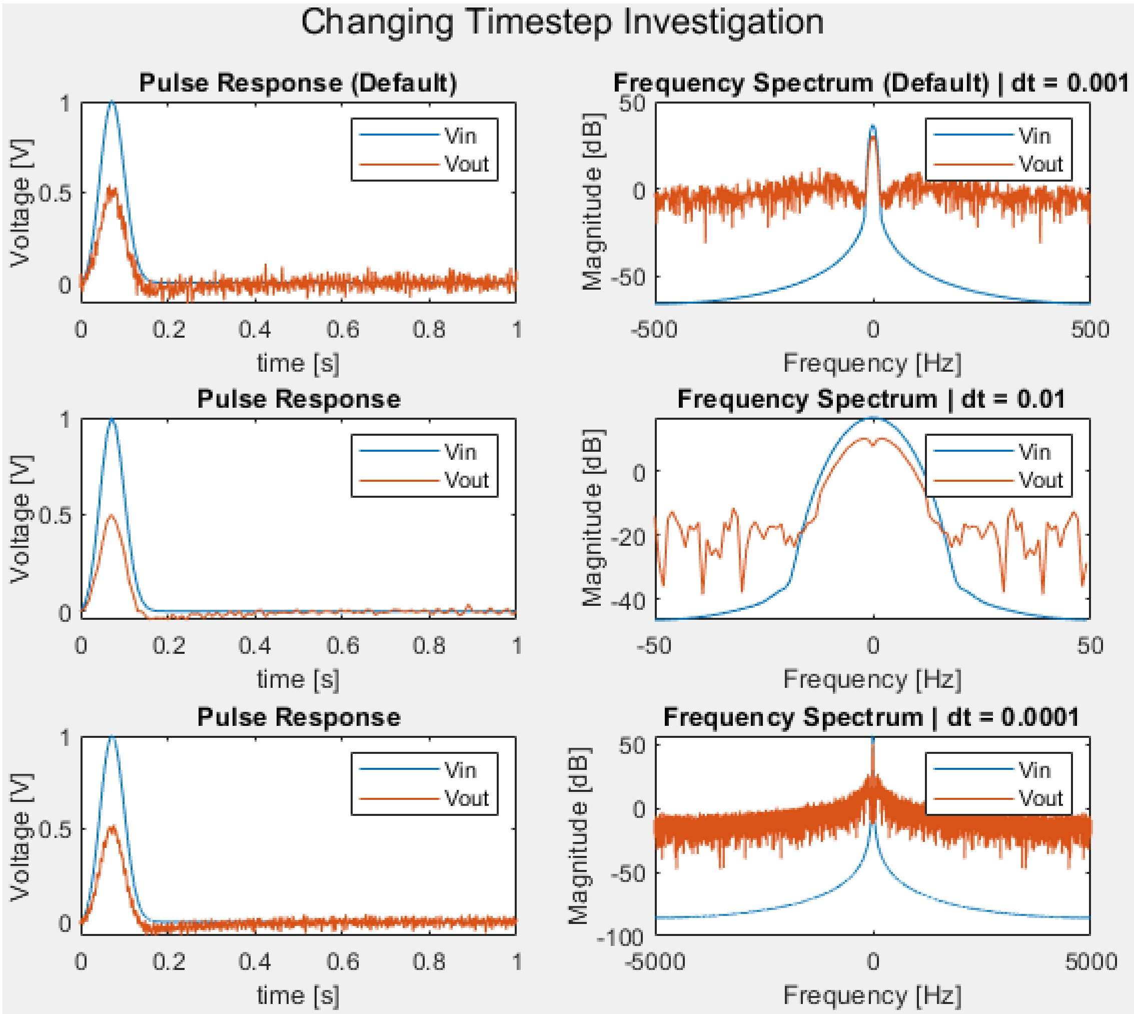

Figure 17: Timestep Investigation

These results clearly show the difference in resolution that the timestep creates. As expected, more timesteps allow for greater resolution because there are more samples/data points. As seen in the results, 1ms and 0.1ms give great insight into the problem. In fact, since the results are nearly the same, it is evident that 1ms would have been enough to understand the behaviour of the circuit, and that, while 0.1ms is more accurate, it is slightly unnecessary. However, when the timestep was 10ms, this clearly shows how the circuit is not sampled enough. It still provides insight, but far less of the noise shows up, and which is most noticeable in the frequency domain.

Therefore, in general, using more timesteps will definitely provide more accuracy and resolution in the results, but general simulations should vary the timestep as well to ensure enough samples are taken to fully understand a generic circuit.

Part 6: Non-Linearity

The purpose of this part would have been to investigate the effects of any non-linear components within the circuit. This could be implemented by replacing $V = \alpha I_3$ with $V = \alpha I_3 + \beta I_3^2 + \gamma I_3^3$.

If this change was done, it would be very easy to implement. In MNA, all non-linear unknowns are placed within the B-matrix. The resulting equation would then be:

$$C\frac{dV}{dt} + GV + B = F$$

where B is a matrix of V (B(V)). Then, to fully implement this, all that would have to change in the transient analysis would be to add the B vector into the calculation, 'alongside' the F vector. This is shown as follows:

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

Therefore, it is clear the same simulations/code could be re-used, and the only challenge would be to adjust the stamping functions to create the B-matrix. Once completed, the B-matrix could just be implemented into previous solution as shown above.

One possible solution to changing the stamps would be to create a new stamp that creates a current controlled voltage source (VCVS), and have its non-linear coefficients inputted into the function as a vector. Then, the coefficients could be manipulated to form the stamp used to help create the B-matrix.