Project 1: Checkpoint

1 Part 1

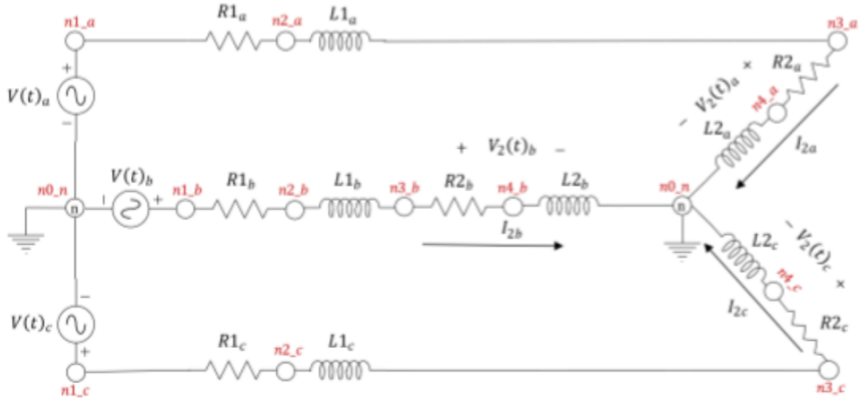


Figure 1: **Part 1 RL Circuit**

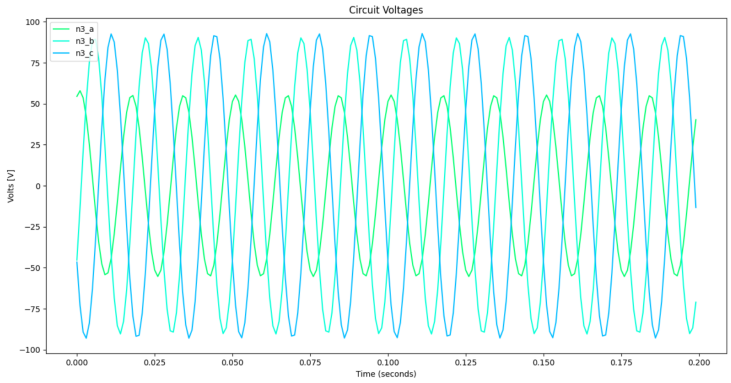
(a) **Perform a time domain simulation for 0.2s** Results are shown in Figure 2

(b) **Compute and plot the output voltages (V_{2a},V_{2b},V_{2c}) and output currents (I_{2a},I_{2b},I_{2c}). All voltages should be on one figure, and all currents should be on one figure.** Results are shown in Figure 2

(c) **Analyse the response of the output voltages and currents. No need for a lengthy analysis; a few sentences are sufficient so long as the thoughts are clear and coherent.** Our circuit forms three RL circuits. With an RL circuit, we expect transients in the output current. Recall that the time constant for a RL circuit is $\tau = L/R$ and it is proportional to the time it will take the current to converge to its final bounds. In this circuit, we have an increasing trend in the time constant. The resistance and inductances vary somewhat arbitrarily, so it is hard to compare the time constant effects across lines. However, the current transients are clear in the Simulink results as well as my simulator outputs, Figure 3 and 2, respectively.

(d) **Explore the computational efficiency of your simulation when using dense v.s sparse matrices. Comment on the results.** When using dense matrices, the simulation takes 88,000,000 nanoseconds, with a low variance, but when using sparse matrices, it takes 88,000,000 nanoseconds, with a higher variance. In this case, the sparse matrix methodology is likely not able to significantly boost execution speeds due to the small size of the system. I ran auxilliary tests with the scipy sparse matrix computation features, and the sparse matrix solver does not consistently exceed the performance of Numpy's dense matrix solver until about $n = 100$ (matrix size). At $n = 1000$, the sparse matrix solver is an order of magnitude faster. [*However, it should be noted that the systems solved had simple solutions that would not cause convergence issues for sparse matrix solvers].

My Simulator Three-Phase RL Voltages



My Simulator Three-Phase RL Currents

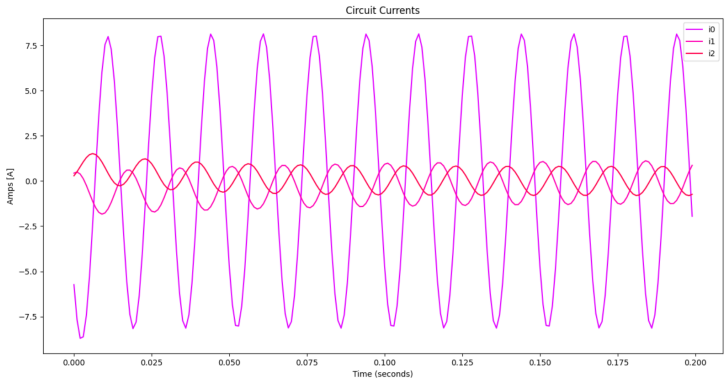


Figure 2: **RL Circuit Results**: i0,i1 and i2 correspond to i2a,i2b and i2c

(e) **Compare your solution against the signals produced by the Simulink model (validate/project1_rl_circuit.slx) provided in the project distribution folder.** My results do not quite match the simulink results. Even the proportions of the currents and voltages are off. My voltages are updated according to the equation:

$$v(t) = \sqrt{\frac{2}{3}}(amp\_ph\_ph\_rms)cos(2\pi f_c t + radians(phase\_deg)) \tag{1}$$

Still, one thing that is promising is that the phases of the voltages and currents are shifted and the current appears to converge as we would expect in an RL circuit. My $\Delta t = 0.001$ in these results, but smaller values did not appear to change the output. I believe that my initial conditions could be the issue. I assumed all the initial voltages and currents in companion models were 0, instead of deriving them from a DC version of the circuit at $t = 0$.

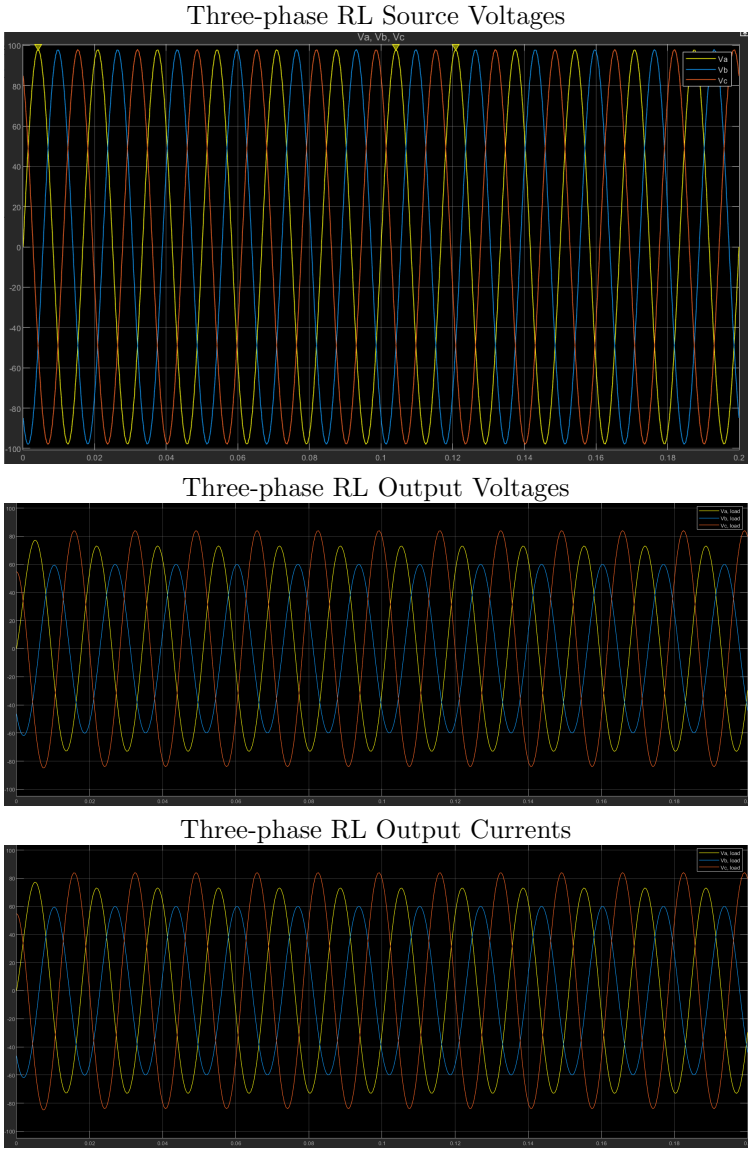


Figure 3: **RL Circuit Reference**:Results from the Simulink version of the circuit