MXET 375
Applied Dynamic Systems

Multidisciplinary
Engineering Technology
COLLEGE OF ENGINEERING

LABORATORY # 1

Simulink Introduction & Tutorial

Submission Date: 09/05/2024

Name: Kyle Rex
Section: 906
UIN: 932008894

Introduction

The purpose of Lab 1 is to provide general problems that are meant to expand an individual's knowledge on how to use Simulink and MATLAB. The objectives include gaining a general Simulink functionality understanding, learning to create block diagram models using Simulink libraries, and appropriately formatting figures for use in creating professional lab reports. Task 1 focuses on creating a simple Simulink model with proper formatting and following video instructions to create another Simulink model. Task 2 focuses on simple signal generation by creating a modified sine wave and a pyramid signal. Task 3 focuses on creating a simple signal of a car's path modeling its displacement, velocity, and acceleration. At the end of the lab the individual should have a better understanding of creating, plotting, configuring, and formatting models using Simulink.

Procedure & Lab Results

This lab has 3 tasks total. Task 1 has 2 parts, Part A and Part B. Task 2 has 2 parts, Part A and Part B. Lab 3 has 1 part. Each task includes a detailed description of the setup, procedure, results, relevant figures, and discussion focusing on developing a better understanding and interpretation of what the results mean and how they were derived. For this lab specifically it is important to understand and recognize that the models are not representative of an actual model and are just simple plots. This means that the y-axis and what the lines represent are undefined.

Task 1

Task 1 follows instructions from a simple Simulink model tutorial provided and an introduction video model. The purpose of these tasks is to become more familiar with Simulink, its libraries, creating block models, and configuring blocks. This task is straightforward and doesn't require much thought other than following the extremely specific instructions.

Task 1 Part A

Task 1 Part A provided the task of creating a simple Simulink model using a sine wave block, integrator block, and bus creator. It also introduces the scope and its value to modeling the result of block diagrams. A MATLAB Simulink introduction tutorial webpage that was provided walks through the exact setup, configuration, and procedure for creating the block diagram. This task is intended to teach the user how to create their first model ever using Simulink and goes into immense detail.

The procedure for this began with importing all the required blocks and setting them up in the correct setup configuration. The setup configuration of the required block diagram to produce the correct plot can be seen in Figure 1. To correctly format the block diagram, it must include all the block names. This can be done by highlighting the blocks, right clicking on them, clicking "Format", then "Show block name", and finally "On". With the setup complete the next step was to configure the model with the appropriate values. In the case of this problem that is done by setting the Model Configuration Parameter Max Step Size to 0.2 and changing the stop time to 20 seconds. Once that is complete the final step is to run the system and format the result with these specified requirements: a white background, axes labels with units, legend, and title. The final result with correct formatting can be seen in Figure 2. This can be done following these procedures:

> Refrain from using a screenshot to capture and save an image of your plot. Unwanted clipping can occur, and it produces a poor result. The better way to save your plot is to do the following: 1. In your scope window, click on “File” in the top left corner. 2. Click “Print to Figure”. This will open a new window that allows you to do more sophisticated editing of your plots. Here we can save the figure as multiple different file types. 3. Click on “File” and then “Save as...”. This will allow you to save the image with your specified file name and to your specified directory. 4. Select the image format of your choice, .jpg or .png will be sufficient. [1]
>
> After simulating a model, you can view the simulation results in a Scope window. 1. Double-click the Scope block. The Scope window opens and displays the simulation results. The plot shows a sine wave signal with the resulting cosine wave signal. 2. On the Scope window toolbar, click the Style button. A Style dialog box opens with display options. 3. Change the appearance of the display. For example, select white for the Figure color and Axes color background (icons with a pitcher). 4. Select black for the Axes color ticks, labels, and grid colors (icon with a paintbrush). 5. Change signal line colors for the Sine Wave to blue and the Integrator to red. To see your changes, click OK or Apply. [2]
>
> The following process is for adding relevant information: 1. Open the plot. 2. Click the settings wheel icon in the top left. 3. In the Time tab, check the bottom box to display the x-axis. 4. Change the time units to seconds. 5. In the Display tab, change the title to an appropriate one for your task 6. Check the box to display the legend. [1]

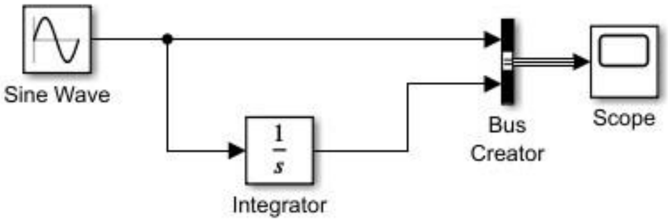

Figure 1: Task 1 Part A Block Diagram.

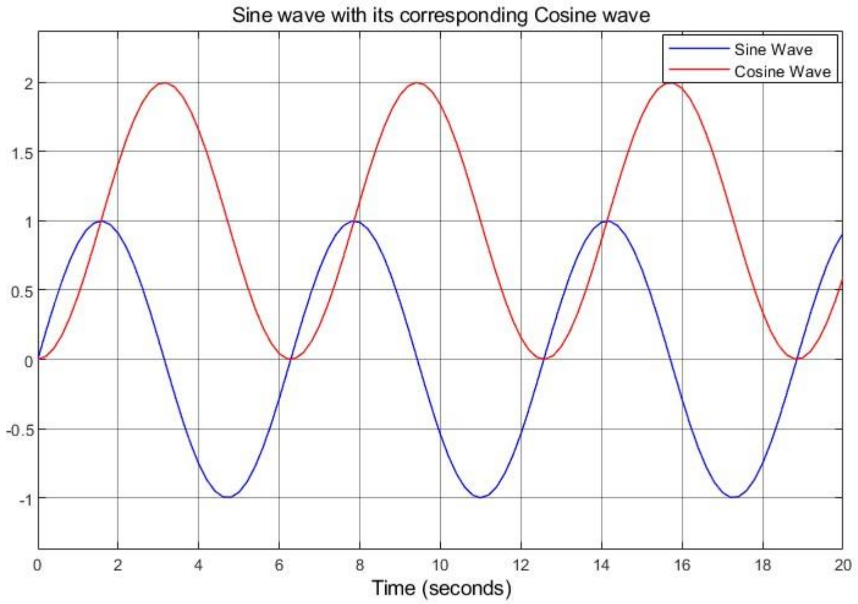

Figure 2: Task 1 Part A Sine and Cosine Wave Plot.

Based on the resulting block diagram and plot it can be inferred that the integrator was used to integrate the sine wave to form a cosine wave (Because the integral of sine is negative cosine). It is a fairly simple model to understand with the only configurations being to the Model Configuration Parameter settings setting the Max Step Size to 0.2 and changing the stop time to 20 seconds. It can be assumed these parameters were changed to create a better model and give a more wholistic picture of the output of the system.

Task 1 Part B

Task 1 Part B provided the task of creating an electrical system represented by a first order differential equation using a step block, integrator block, sum block, and gain block. A Simulink introduction tutorial video that was provided walks through the exact setup, configuration, and procedure for creating the block diagram.

The procedure for this began with importing all the required blocks and setting them up in the correct setup configuration. The setup configuration of the required block diagram to produce the correct plot can be seen in Figure 3. With the setup complete the next step was to configure each block with the appropriate values. In the case of this problem that is done by setting the gain block to a value of -5 and changing the initial condition of the integrator to -2. Once that is complete the final step is to run the system and format the result to the specified requirements described in Task 1 Part A. This result can be seen in Figure 4.

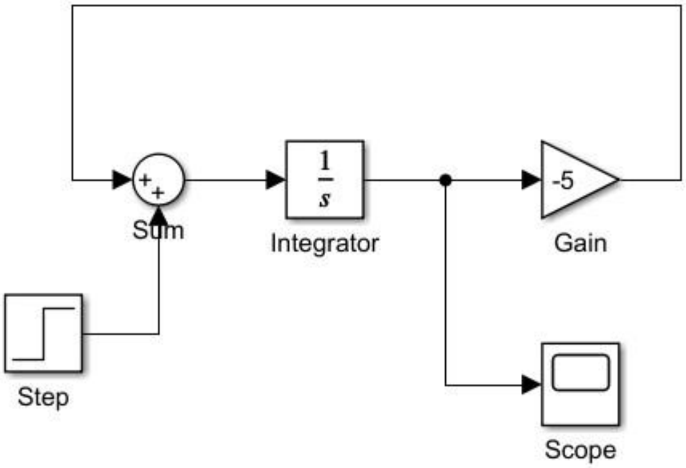

Figure 3: Task 1 Part B Block Diagram.

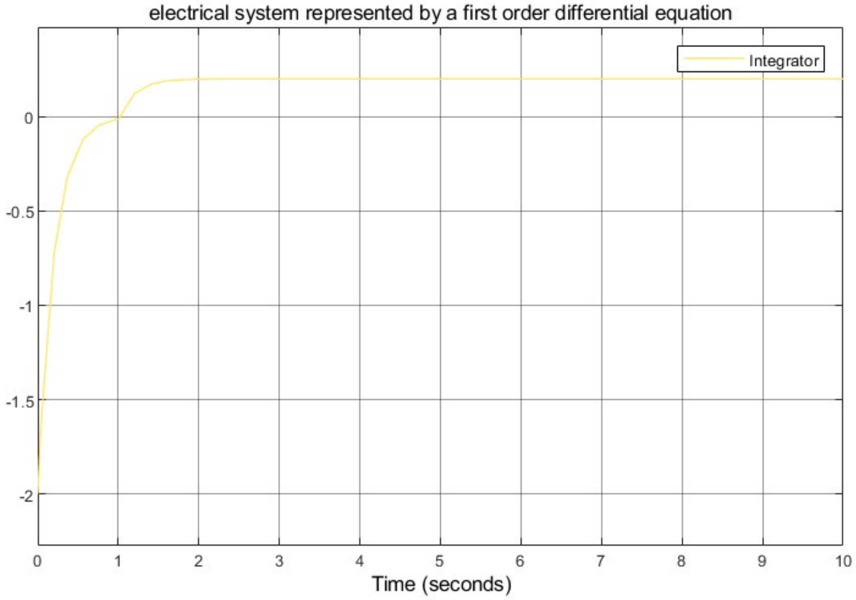

Figure 4: Task 1 Part B Electrical System Plot.

Based on the resulting block diagram and plot it can be inferred that this model is meant to represent an electrical system that is modeled by a differential equation. The equation itself can be replaced by the system model used above. The equation that is converted to this system model is as follows:

$$y(t) = 0.2(1 - 11e^{-5t})u(t) \quad (1)$$

Based on the equation it can be inferred that the model itself requires an adder, an integrator, and a scalar. Based on the equation the values for these blocks were selected and configured as described previously. With this the output can be analyzed to find useful information for the system. The response of the system is divided into two distinct areas. From 0 to 2 seconds the system is non-linear and unstable known as the transient state. While the system higher than 2 seconds is linear and stable known as the steady state response of the system.

Task 2

With the completion of Task 1, which provided a general understanding of how to use Simulink, the focus shifts to applying those skills in a practical sense. Task 2 focuses on creating a few simple signals using the skills learned in Task 1. This involves creating the block diagrams without them being provided and configuring the blocks based purely on inferences from the provided problem plots.

Task 2 Part A

Task 2 Part B provided the task of creating a Modified Sine Wave using a sine wave, pulse generator, an unknown block, and a gain block. The one limitation is that the parameters of the sine wave block could not be changed.

The procedure for this began with importing all the required blocks and setting them up in the correct setup configuration. The setup configuration of the required block diagram to produce the correct plot can be seen in Figure 5. With the setup complete the next step was to configure each block with the appropriate values. In the case of this problem that is done by setting the gain block to a value of 2 and changing the Amplitude, Period, and Pulse Width of the Pulse Generator to -1, 0.5, and 50% respectively. Once that is complete the final step is to run the system and format the result to the specified requirements described in Task 1 Part A. This result can be seen in Figure 6.

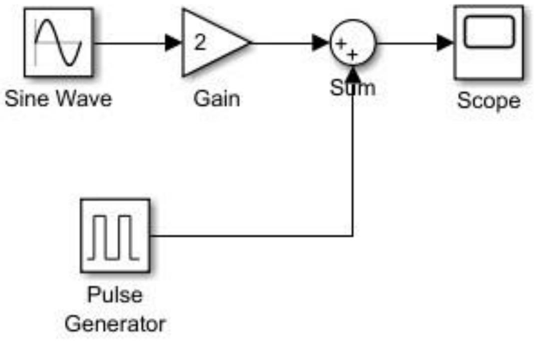

Figure 5: Task 2 Part A Block Diagram.

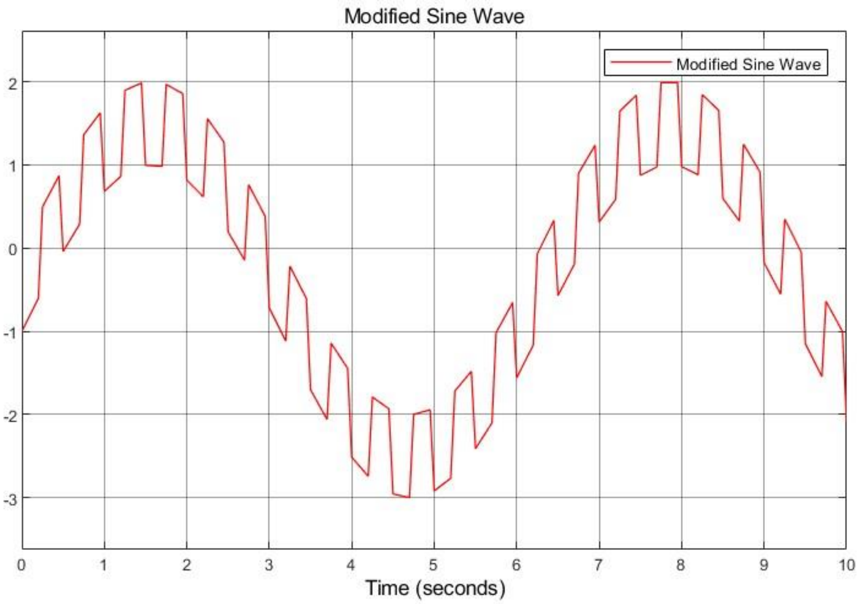

Figure 6: Task 2 Part A Modified Sine Wave Plot.

The first step to solving this problem was figuring out what other block was required to complete this block diagram. Based on the completion of Task 1 Part B it can be concluded that a sum block to add the output of the sine wave and pulse generator together would be required for the scope to receive only one input. Therefore, the sum block was the unknown block.

Based on the provided problem plot it could be inferred that the gain of the sine wave had been increased by a factor of 2 because instead of oscillating between 1 and -1, like a normal sine wave with a gain of 1, it was oscillating between 2 and -2. So, this resulted in using the gain block with a factor of 2 to multiply the sine wave to get the intended output. Having solved this and knowing that the sine wave itself was not to be configured, this only meant that the pulse generator could be configured. Based on the provided problem plot it could be inferred that the amplitude of the pulse generator had to be -1 as throughout the entirety of the sine wave it would be brought down -1 repeatedly. A normal sine wave starts at 0 so since this modified sine wave started at -1 it had to be stepped down by -1. Based on these inferences it can be concluded that the Amplitude of the Pulse Generator should be set to -1. With the Amplitude of the Pulse Generator solved the next step was to find the Period and Pulse Width.

Based on the provided problem plot it could be inferred that the Period (Time from start of low step to end of high step) of the step waveform impacting the sine waveform was 0.5 seconds (Look at the first 0.5 seconds of the plot to see an example of this). It can also be seen that during the first half of that 0.5 seconds the step waveform would be low and the second half of that 0.5 seconds the step waveform would be high (This showed that there was no delay in the pulse generator). Based on this it could be inferred that the Period of the Pulse Generator should be 0.5 seconds and the Pulse Width (% of Period) of the Pulse Generator should be 50 percent.

Task 2 Part B

Task 2 Part B provided the task of creating a Pyramid signal using two ramp functions, and an addition/subtraction function, delaying one ramp function by a certain amount of time to achieve the shape and configuring the slope of the ramp appropriately. There are multiple configurations that achieve the intended outcome, but the focus will be on the configuration described below.

The procedure for this began with importing all the required blocks and setting them up in the correct setup configuration. The setup configuration of the required block diagram to produce the correct plot can be seen in Figure 7. With the setup complete the next step was to configure each block with the appropriate values. In the case of this problem that is done by setting one of the ramp blocks to have a slope of 1, start time of 0, and initial output of 0 and by setting the other ramp block to have a slope of -2, start time of 5, and initial output of 0. Once that is complete the final step is to run the system and format the result to the specified requirements described in Task 1 Part A. This result can be seen in Figure 8.

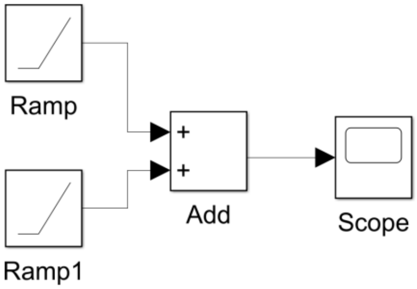

Figure 7: Task 2 Part B Block Diagram.

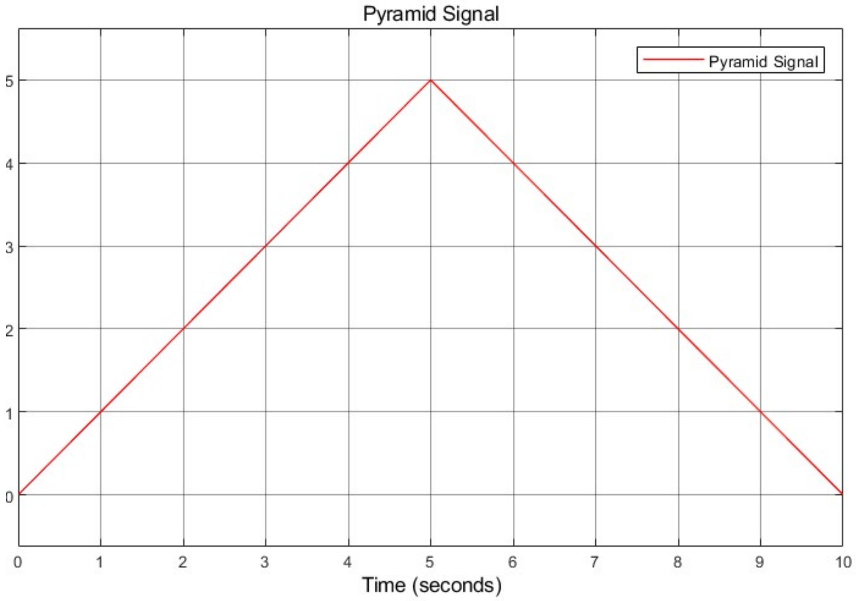

Figure 8: Task 2 Part B Pyramid Signal Plot.

Based on the provided problem plot it could be inferred that there is a linear function with a constant slope of +1 (Can be found using rise/run method: 1/1 = 1) increasing to 5 seconds. Then at 5 seconds this linear function begins decreasing with a constant slope of -1 (Can be found using rise/run method: -1/1 = -1) and continues until 10 seconds. It can also be inferred that since it takes the same amount of time for the ramp function to go from 0 to 5 on the y-axis as it takes to go from 5 to 0 on the y-axis that the functions are inversely proportionate which supports the +1 and -1 slope claim. Based on these inferences it is known that there must be an initial ramp function that starts at 0 and starts increasing with a slope of 1. This represents the configuration for the first ramp block. Then because the signal of the first ramp block and second ramp block are being added together by an adder block the second ramp block must interrupt the first ramp block at 5 seconds to get the desired output. This interruption is done by putting a start time of 5 seconds on the second ramp block and assigning it a slope of -2. -2 because when the first and second ramp block slopes are added together, instead of outputting a slope of +1 while the second ramp block is off for the first 5 seconds, now they are added together resulting in a slope of -1 (-2+1 = -1) until the stop time of 10 seconds returning the function to 0.

Task 3

Task 3 provided the task of creating a simple signal of a car’s path using three step blocks, a sum block, two integrator blocks, and a bus creator. It is given that the three step functions will need to be adjusted to create the provided problem plot. It is recommended to reference the acceleration plot (step plot) specifically to find the values to use in the configuration of the step functions. The plots for the velocity and displacement of the car are created automatically from the integrations of the acceleration plot. This task is straightforward and doesn’t take as much thought as Task 2.

The procedure for this began with importing all the required blocks and setting them up in the correct setup configuration. The setup configuration of the required block diagram to produce the correct plot can be seen in Figure 9. With the setup complete the next step was to configure each block with the appropriate values. In the case of this problem that is done by setting one of the step blocks to have a step time of 1, an initial value of 0, and a final value of 1. Another step block needs to have a step time of 5, an initial value of 1, and a final value of -1. The final step block needs to have a step time of 9, an initial value of -1, and a final value of 0. Once that is complete the final step is to run the system and format the result to the specified requirements described in Task 1 Part A. This result can be seen in Figure 10.

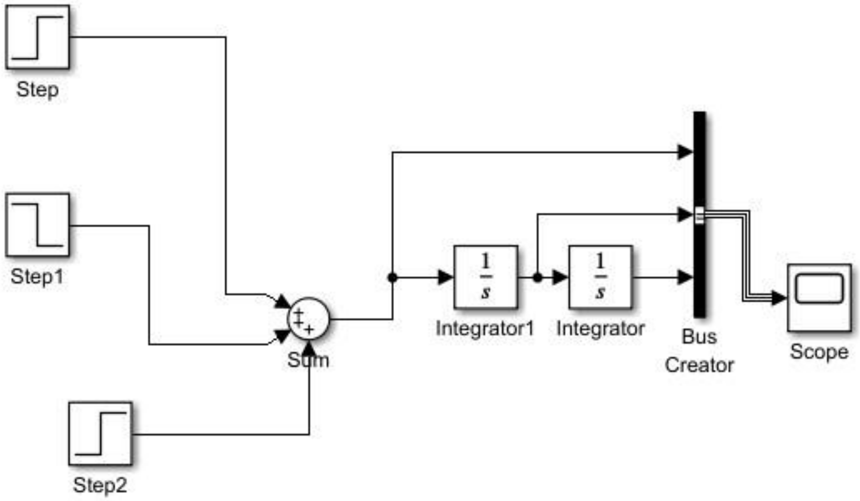

Figure 9: Task 3 Block Diagram.

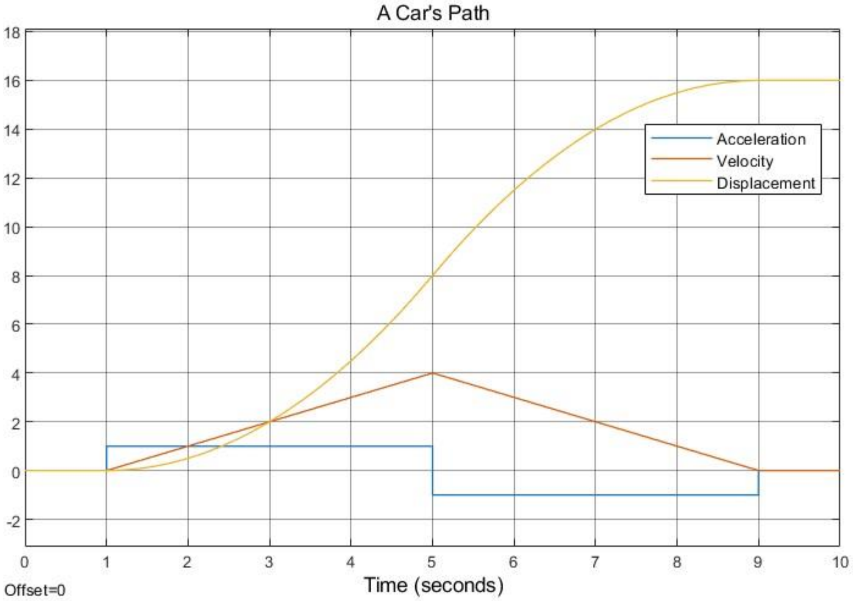

Figure 10: Task 3 Path of Car Plot with Acceleration, Velocity, and Displacement.

Based on the provided problem plot, specifically referencing the acceleration function, it could be inferred that a step of +1 was required at 1 second (The blue acceleration function increasing from 0 to 1 on the y-axis), a step of -2 was required at 5 seconds (The blue acceleration function decreasing from 1 to -1 on the y-axis), and a step of +1 was required at 9 seconds (The blue acceleration function increasing from -1 to 0 on the y-axis). Based on these inferences it can be seen that the acceleration steps directly related to what the step blocks in the block diagram were required to be configured too. After the acceleration steps were configured appropriately the velocity and displacement functions automatically corrected themselves to match the provided problem plot as expected. This plot accurately depicts a car's acceleration, velocity, and displacement. Assuming the y-axis represents meters the following regarding each function can be inferred. The car begins moving at 1 second when its acceleration changes abruptly to a constant 1 m/s^2 for 4 seconds. While this acceleration occurs, the velocity is linearly increasing with a positive slope of 1 m/s until reaching its highest speed of 4 m/s at the 5 second interval. Similarly, during this time, the displacement of the car from its original position is increasing exponentially to 8 meters at the 5 second interval. Then at 5 seconds there is an abrupt change in the acceleration that results in it decreasing to -1 m/s^2 for 4 seconds. While this deceleration occurs, the velocity is linearly decreasing with a negative slope of -1 m/s until returning back to a speed of 0 m/s at the 9 second interval. Unlike the acceleration and velocity, during this time, the displacement of the car from its original position is still increasing exponentially but is beginning to level out to 16 meters at the 9 second interval when finally, the car stops, and the acceleration makes a final abrupt change to 0 m/s^2 and remains there for the last second. The velocity and the displacement follow suit and remain at their previous and final values of 0 m/s and 16 meters respectively.

Post-Lab Questions

1. From Task 3, If the resulting plots describe the path of a car, how would a driver generate this pattern? (When to accelerate/decelerate & stop?)

For the purpose of the question, assume the y-axis represents meters. To generate the resulting plot the driver would need to begin accelerating 1 m/s^2 at 1 second and continue until 5 seconds. Then the driver would need to begin decelerating -1 m/s^2 at 5 seconds and continue until 9 seconds. Finally at 9 seconds the driver needs to return to a constant speed not accelerating or decelerating. Doing this will result in the pattern shown in the plot.

2. If we assume length is measured in meters, how far has the car traveled at 5 seconds?

Based on the displacement curve, which depicts the distance/displacement of the car from the starting point, it can be concluded that at 5 seconds the car had traveled 8 meters.

3. Keeping the same assumption, what is the velocity of the car at 7 seconds?

For the purpose of the question, assume the y-axis represents meters. Based on the velocity curve, which depicts the velocity/speed of the car at a particular time, it can be concluded that at 7 seconds the car has a velocity of 2 m/s.

Conclusion

In conclusion, not only did this lab focus on developing a better understanding of creating, plotting, configuring, and formatting models using Simulink, but it also provided the opportunity to begin to learn how to analyze these plots and apply them to relevant problems. This can be seen in Task 3 and the post lab questions where the focus is on analyzing the plot as a Car's path finding the displacement, velocity, and acceleration of the car at certain time intervals from 0 to 10 seconds. All the tasks in this lab focused on how to create the correct system configuration for desired designs, how to properly use & configure a diverse range of blocks to produce the correct results for those designs, and how to accurately depict relevant results of said system designs. The final conclusion from this lab is the understanding that MATLAB and Simulink are useful simulation tools for solving many practical engineering problems and furthermore developing the skills required to properly utilize them is extremely important.

References

[1] Author not listed, *Texas A&M University MXET375 - Lab 01 Part 1 - Simulink Introduction*. College Station, TX, USA: Date not listed.
[2] Author not listed, *MATLAB Simulink Tutorials Texas A&M University MXET375 - Lab 01 Part 2 - Simulink Tutorial*. College Station, TX, USA: Date not listed.
[3] A. Kuzechie, *MATLAB Tutorials - Introduction to Simulink*. USA: Youtube, 2011.