WIC Self Learning New Skills

ENGR 416

Ethan Shingleton

934437388

4/19/2026

In order to fulfill the coding requirement necessary for the Structures and Recovery HALE (High Altitude Liquid Engine Rocket) capstone, a completely autonomous reactive system must be orchestrated in order to react systematically with the sensors and alter the angle of the fins in relation to the fuselage in order to stabilize the rocket. The final coding language must be C++ to corroborate with the current HALE avionics package. So in order to quickly and efficiently design and simulate the rocket and its resulting systems, I chose to learn Simulink as both Lafayette Systems [1] and BPS Space [2], two eminent youtubers who document their workflow, have published videos in which they go over their models and workflow.

With the intention of finding an efficient way to code such a complex system in a language with minimal experience, I began by going to youtube and seeing what people do for their particular situation. This led me to finding Simulink and doing the opening tutorial represented in Figure 1 [3].

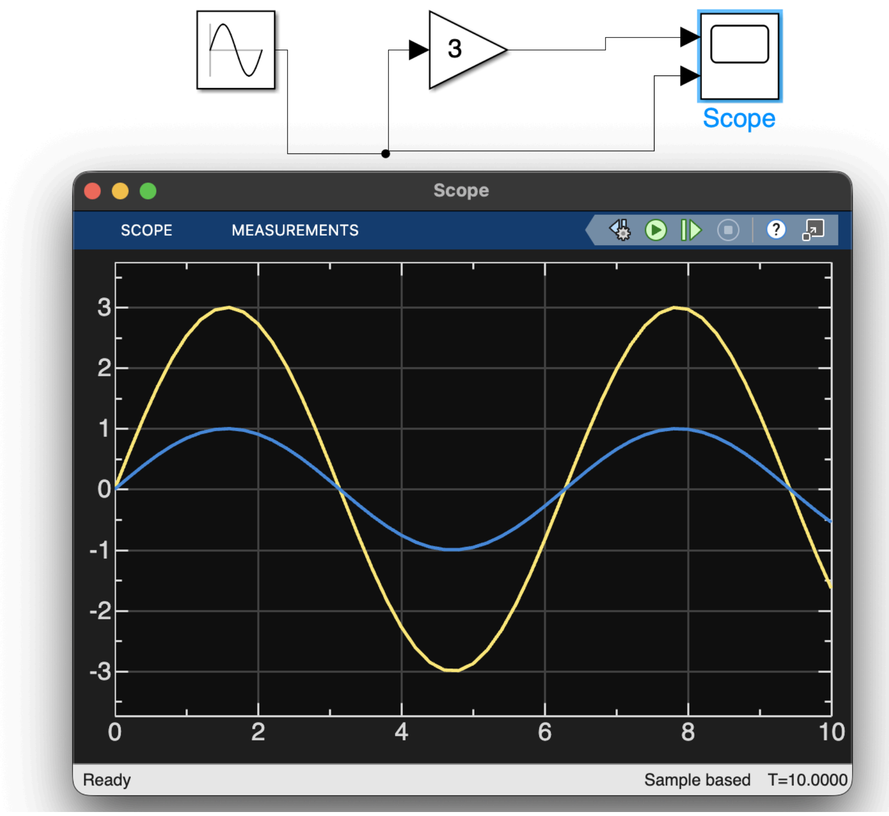

Figure 1. Simulink Tutorial Practice

I then went and watched the videos Modeling and Simulation of Advanced Amateur Rockets [1] and Modeling a Thrust Vectored Rocket In Simulink[2] in which the former directly applied to my system while the latter was for thrust vectoring. However, Lafayette did not publish their model and only showed images of it, making it difficult to use beyond just being a reference. BPS Space on the other hand has uploaded their model and made it public access as shown in Figure 2. I used it as a reference for how to set up complex altitude related relationships and model the resulting byproducts.

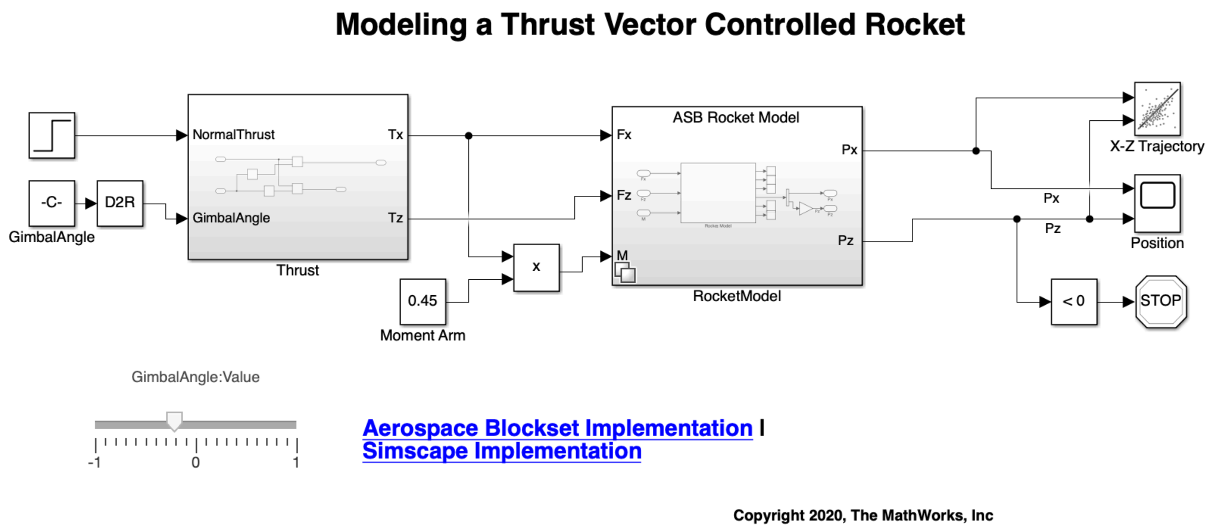

Figure 2. BPS Space Opensource Simulink Thrust Vectoring Model

Using BPS Space as the primary reference combined with the workflow talked through by Lafayette Systems, I began my own model from scratch and attempted to begin a baselayer for what would become my Capstones primary code package. This led me to Designing a Guidance System in MATLAB and Simulink[6] which established a complex example that delved deeply into how to build an autopilot missile targeting system, demonstrated in Figure 3. While the autopilot and technical details fell far beyond the morality and feasibility of my project, its blocks for atmosphere and airframe can be directly applied to my model. Which will be incorporated into more advanced iterations in the future.

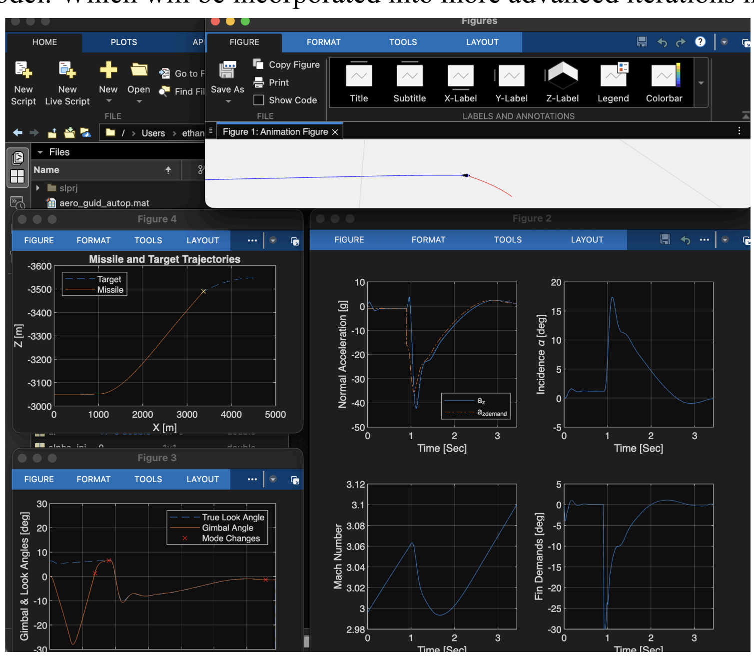

Figure 3. Missile Targeting System

To move beyond tutorials and begin applying Simulink directly to my capstone, I created a simplified closed loop roll control model for the rocket. The model used a desired roll rate of zero, compared that target to the simulated roll response, and used the resulting error to generate a corrective control command. A gain block represented the controller, a saturation block represented actuator limits, and a first order transfer function represented simplified rocket roll dynamics. A disturbance input was added at one second to test whether the model could respond to unwanted motion. The resulting scope output showed that the system remained at zero before the disturbance, then responded smoothly and approached a stable steady value over time represented by flatlining in Figure 4.

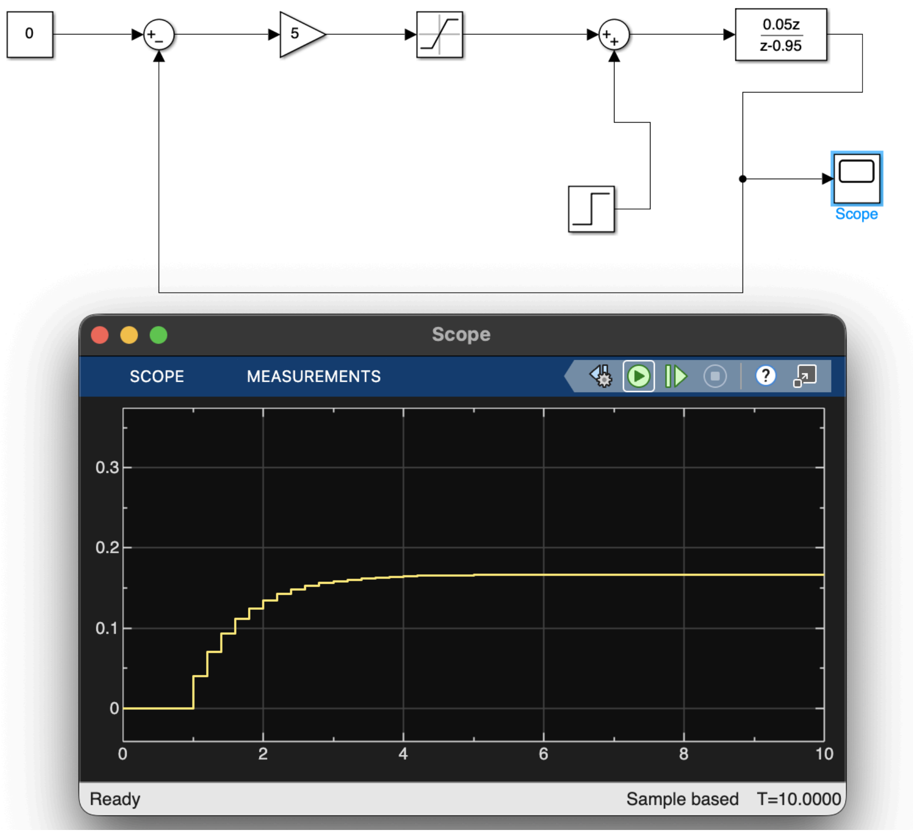

Figure 4. My Capstone's Initial Simulink Setup for Roll Control

This confirmed that the feedback loop was connected and functioning correctly. Although this model is still a simplified first iteration and is subsequently very rudimentary, it established the baseline control architecture needed for future development. Overall, this self learning process allowed me to move from having no Simulink experience to building and running an applied control model directly related to my HALE capstone project, taking the first steps to a working code.

References:

[1] Lafayette Systems, “Modeling and Simulation of Advanced Amateur Rockets,” *YouTube*, Aug. 19, 2024. https://www.youtube.com/watch?v=sEzRzkGRpDQ
[2] BPS Space, “Modeling a Thrust Vectored Rocket In Simulink,” *www.youtube.com*, Oct. 14, 2020. https://www.youtube.com/watch?v=nwgd1CV__rs (accessed Jan. 22, 2024).
[3] MATLAB, “Getting Started with Simulink, Part 1: How to Build and Simulate a Simple Simulink Model,” *YouTube*. Dec. 04, 2017. Available: https://www.youtube.com/watch?v=iOmqgewj5XI
[4] BPS Space, “- YouTube,” *www.youtube.com*, Sep. 27, 2018. https://www.youtube.com/watch?v=LIu2MedOHLc (accessed Apr. 20, 2026).
[5] Lafayette Systems, “- YouTube,” *www.youtube.com*, Jan. 05, 2024. https://www.youtube.com/watch?v=DisqUKxrcfo (accessed Apr. 20, 2026).
[6] Matlab, “Designing a Guidance System in MATLAB and Simulink - MATLAB & Simulink,” *www.mathworks.com*. https://www.mathworks.com/help/simulink/slref/designing-a-guidance-system-in-matlab-and-simulink.html