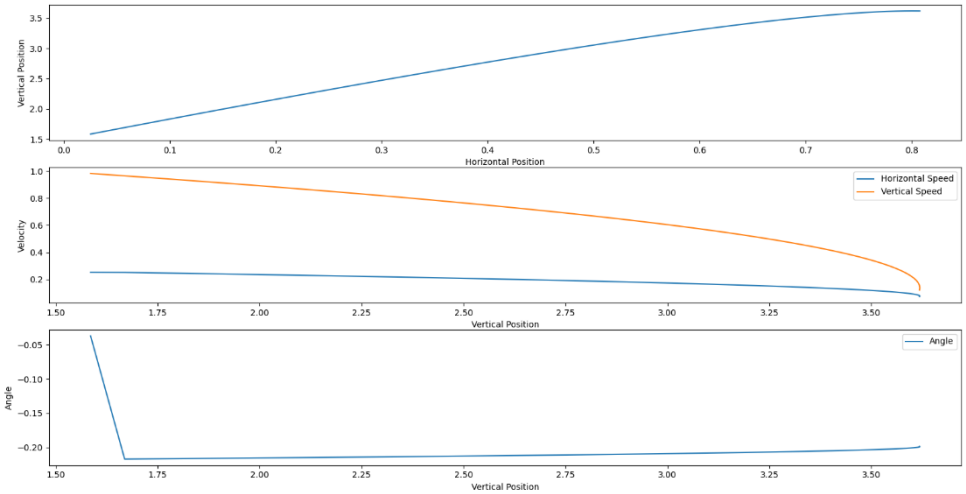
Initial Conditions: [dx, dy, Vx, Vy, theta] = [0, 1.5, 0.25, 1, 1/7]

Initial Loss: 26.818

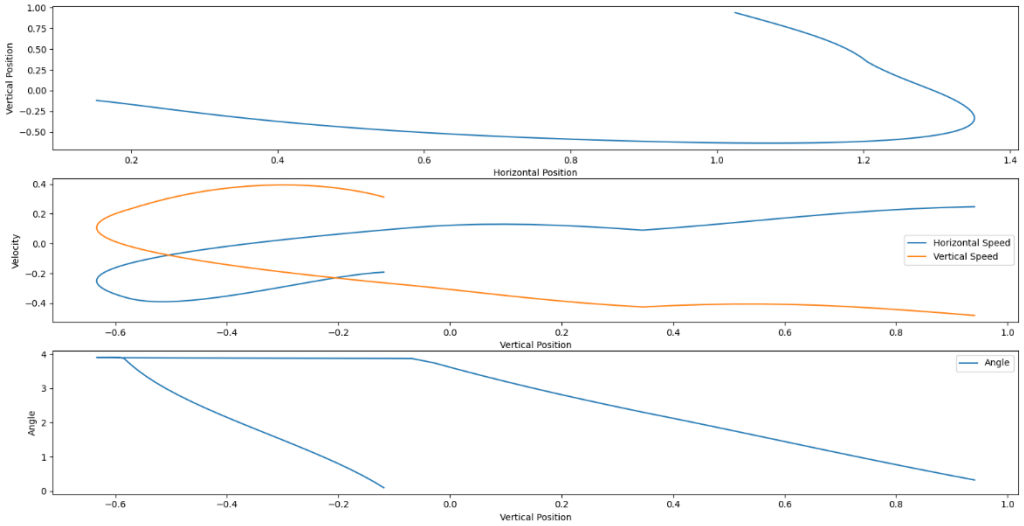
Loss: 16.297



Initial Conditions: [dx, dy, Vx, Vy, theta] = [1, 1, 0.25, -0.5, 1/7]

Initial Loss: 33.851

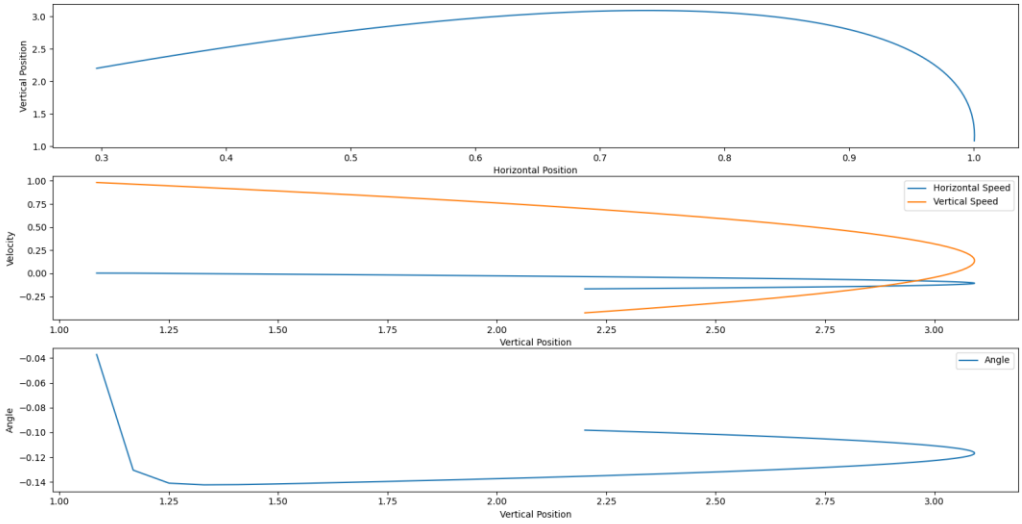
Loss: 0.249



Initial Conditions: [dx, dy, Vx, Vy, theta] = [1, 1, 0, 1, 1/7]

Initial Loss: 20.866

Loss: 9.560



With these 3 different initial conditions the rocket was not able to reach the landing pad with a loss of 0 in any of the instances. I assume this is due to a fundamental lack of understanding on what purpose the state_tensor serves as I, apparently don't understand it and probably won't in the foreseeable future. Going to office hours to ask now that the project is over is probably my best bet. Despite this, the program was able to minimize the loss which is what it's supposed to do. Even in Case 2 it got something resembling what the answer was supposed to be which tickled me pink. Accounting for drag and using a distribution of initial states I was able to come up with this code which gave me the "optimal" solution.