Independent Study Report on Flatness-based Model Predictive Control for Quadrotor Trajectory Tracking

Ahmed Khalil

Department of Mechanical Engineering

University of Wisconsin-Madison

May 9th, 2021

Project Overview

- Course: Mechanical Engineering Projects I
- Term: Spring 2021
- Advisor: Dr. Xiangru Xu
- Topic: Model Predictive Control for Quadrotor Trajectory Tracking

Objective

To replicate and simulate the quadrotor controller algorithms described in the research paper titled “Flatness-based Model Predictive Control for Quadrotor Trajectory Tracking” in MATLAB.

Brief Description

The proposed algorithm, FMPC, takes advantage of the quadrotor’s differential flatness property and couples feedback model predictive control with feedforward linearization. This proposed approach has a computational advantage as it only requires a convex quadratic problem to be solved, allowing the computations to be completed on-board of the quadrotor. The proposed method also claims to have improved trajectory tracking compared to other model predictive control approaches. I was tasked with replicating this method in this paper on MATLAB as the Xu Research Group is currently developing a toolbox for quadrotor simulation.

Methodology

Recreating this algorithm was a difficult task for me, mainly because I had limited MATLAB experience and almost no knowledge about MPCs or quadrotor dynamics. A large portion of my time was dedicated to researching how others created MPCs for different algorithms and then use their work to get the desired outcome. This research included the work previously done by Xu Research Group members as well as work which I found online.

The main.m file is used to run the simulation and to graph the results of the simulation. Before running the simulation, a reference trajectory needs to be chosen as a 3D parametric function. The simulation is then run using a for loop, the dynamics files, and the MPC file called positionMPC.

Results

To test the results of the code, I decided to choose various parametric equations as my reference trajectories. This was a good decision as it allowed me to get a rough idea of how well the code operates. The parametric equations also do not consider the quadrotor's initial conditions, which is important as we can measure how long it takes for the quadrotor to follow the reference trajectory with a high accuracy. The less time it takes for the quadrotor to follow the reference trajectory, the better the MPC.

The upcoming pages include the results of various reference trajectories that are based on parametric equations. This is not the most effective way of testing this MPC and I plan on using the get_trajectory file created by Victor Freire to choose points in space rather than smooth functions.

The following functions were arbitrarily chosen as reference trajectories, except for the last one which is the infinity sign. In future, I plan on finding other trajectories that might give me a better understanding of the FMPC's accuracy.

$$x(t) = 2 * \cos(t), y(t) = 8 * \sin(t), z(t) = \frac{t}{3}$$

(Function 1)

$$x(t) = t * \cos(t), y(t) = t * \sin(t), z(t) = t$$

(Function 2)

$$x(t) = \sin(t), y(t) = \cos(t), z(t) = t$$

(Function 3)

$$x(t) = 2^{-\frac{t}{10}} * \sin(t), y(t) = 2^{-\frac{t}{10}} * \cos(t), z(t) = t$$

(Function 4)

$$x(t) = \cos(t), y(t) = \sin(t) * \cos(t), z(t) = t$$

(Function 5)

Figures 1-6, 7-12, 13-18, 19-24, and 25-30 show the results for functions 1,2,3,4, and 5, respectively. Each simulation took roughly 45 seconds to run on MATLAB, including all the graphing and setting up the simulation.

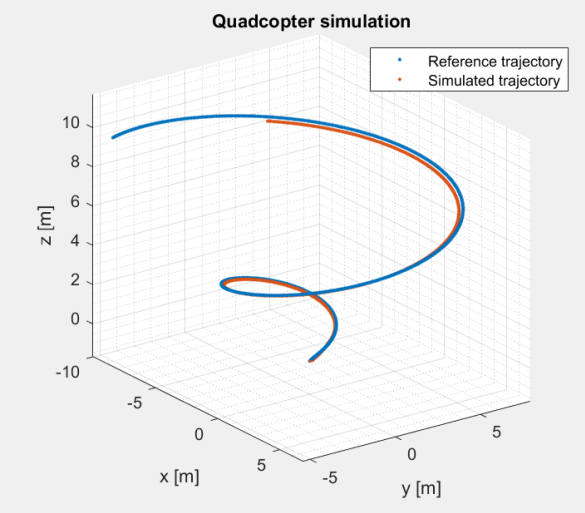

Figure 1: First Trajectory 3D View for 1st Function

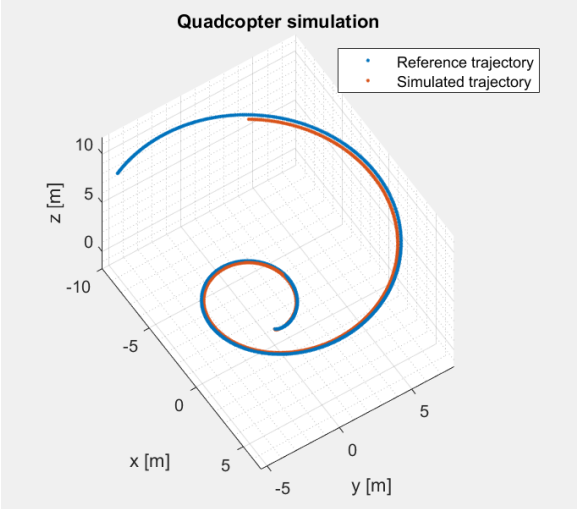

Figure 2: Second Trajectory 3D View for 1st Function

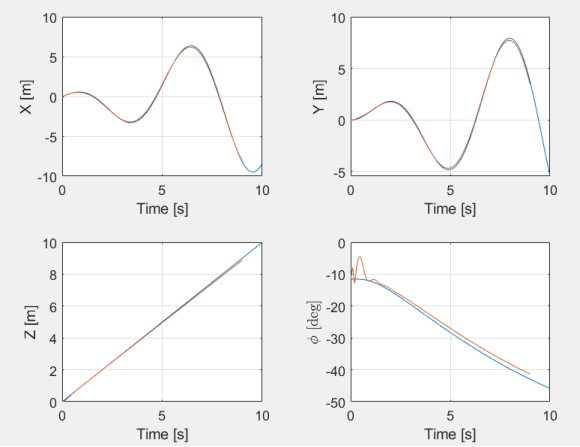

Figure 3: Position and Phi Values

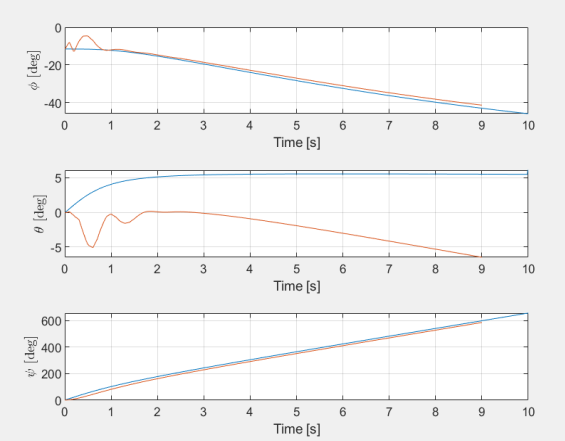

Figure 4: Orientation Values

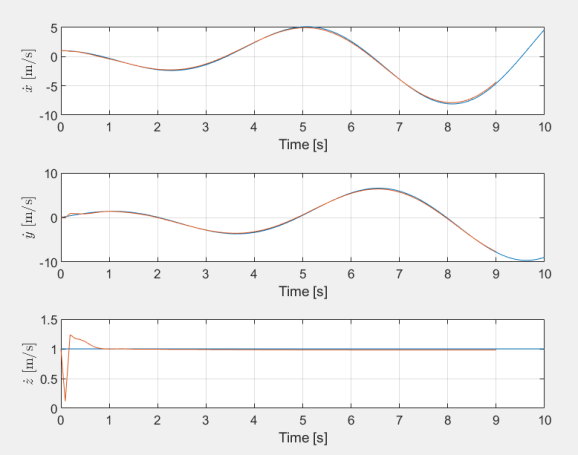

Figure 5: Velocity Values

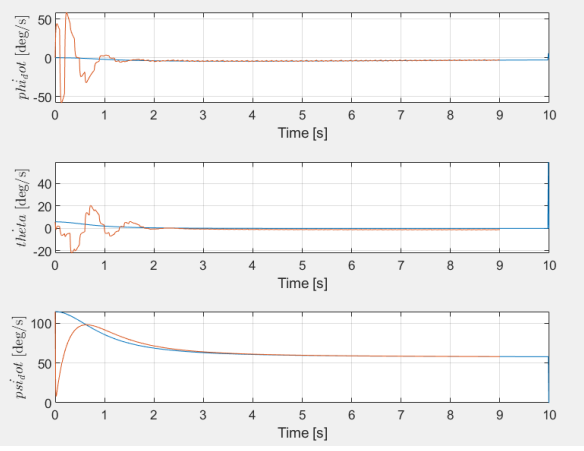

Figure 6: Angular Velocity Values

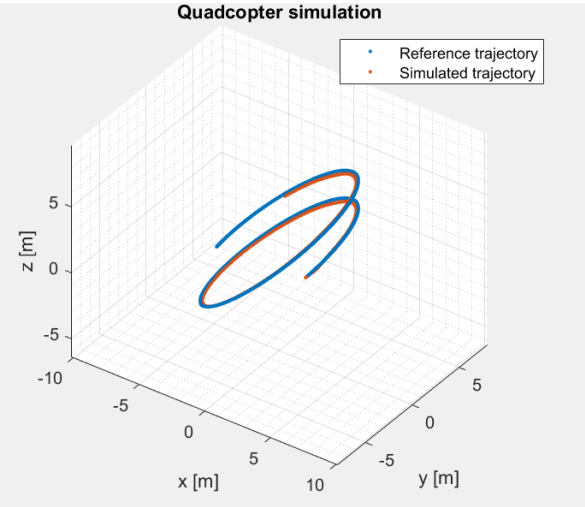

Figure 7: First Trajectory 3D View for 2nd Function

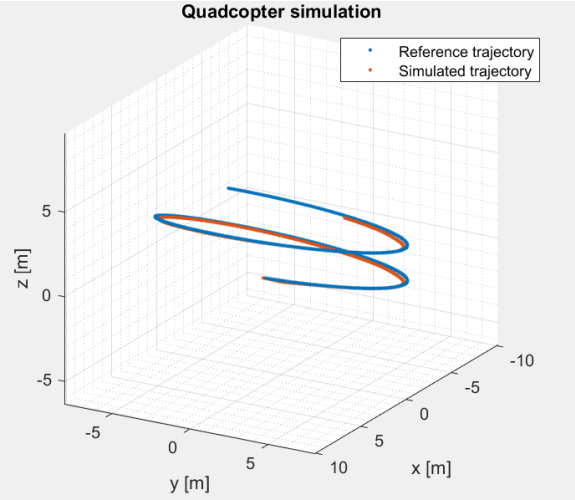

Figure 8: Second Trajectory 3D View for 2nd Function

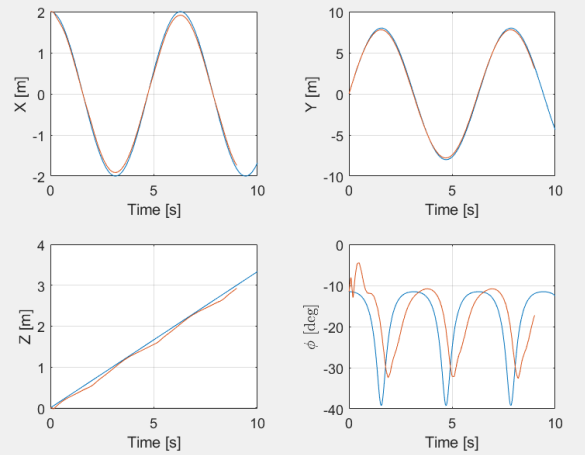

Figure 9: Position and Phi values

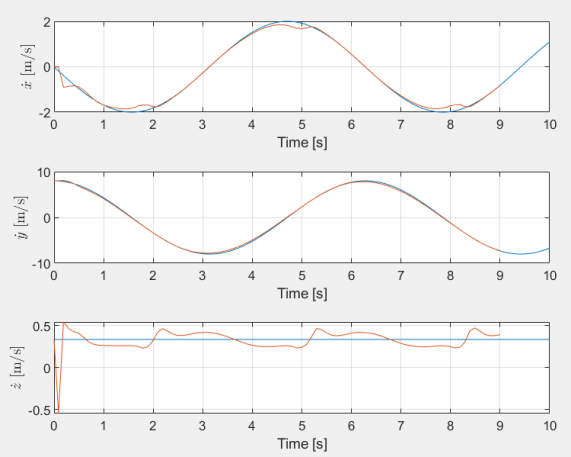

Figure 10: Orientation Values

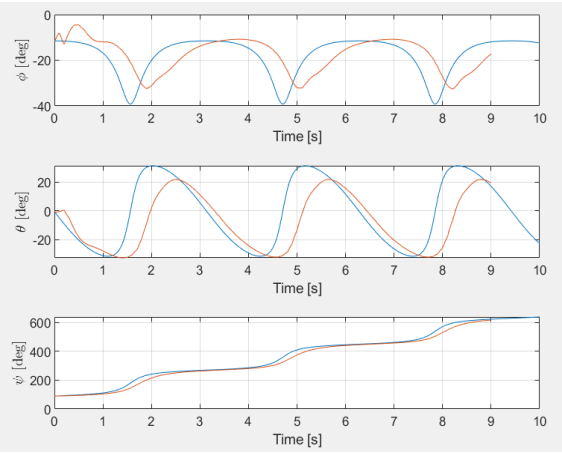

Figure 11: Velocity Values

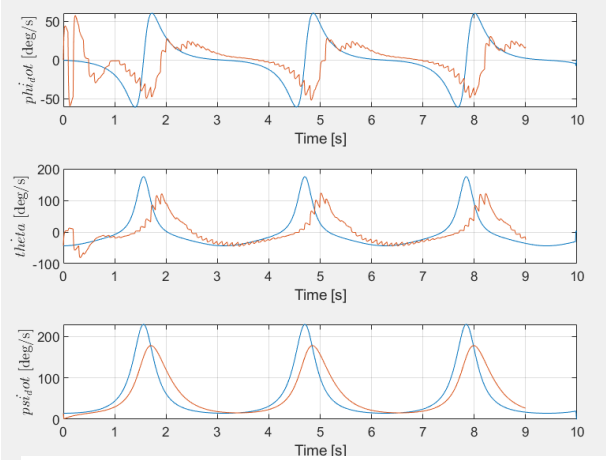

Figure 12: Angular Velocity Values

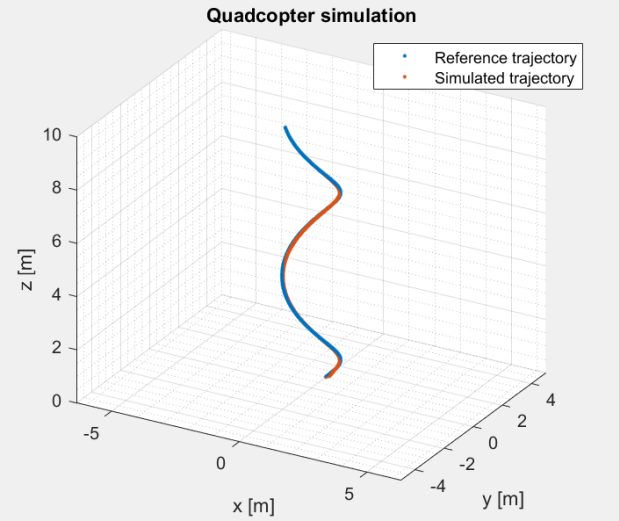

Figure 13: First Trajectory 3D View for 3rd Function

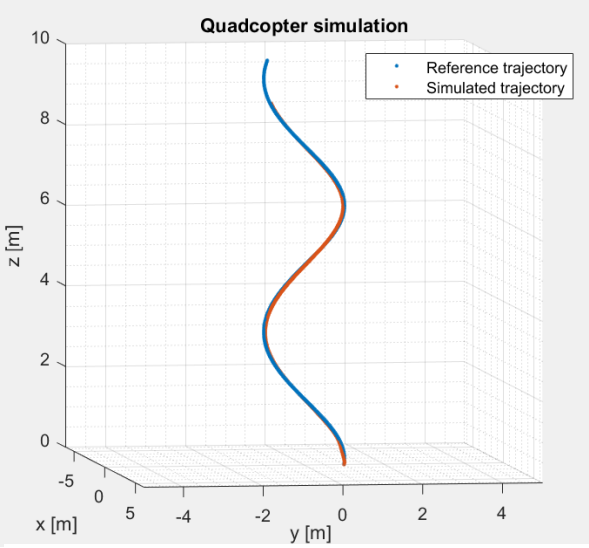

Figure 14: Second Trajectory 3D View for 3rd Function

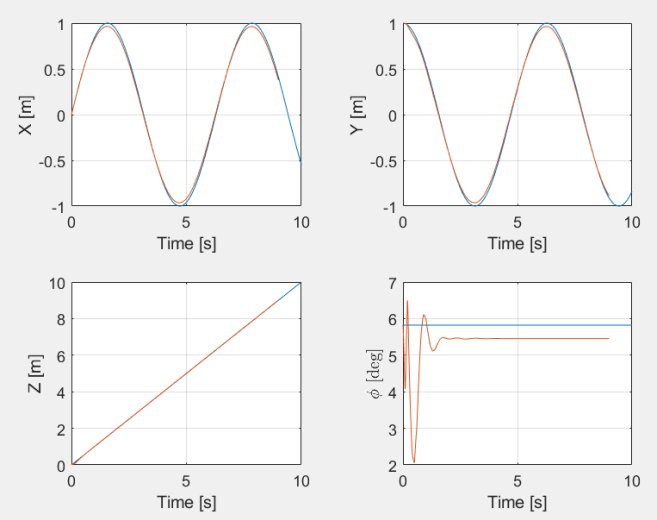

Figure 15: Position and Phi values

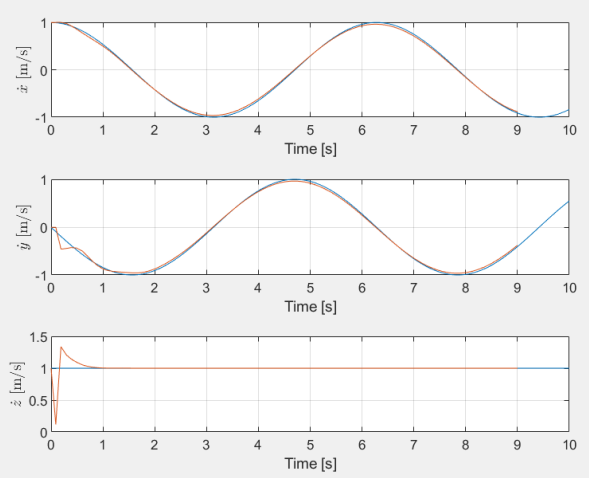

Figure 16: Orientation Values

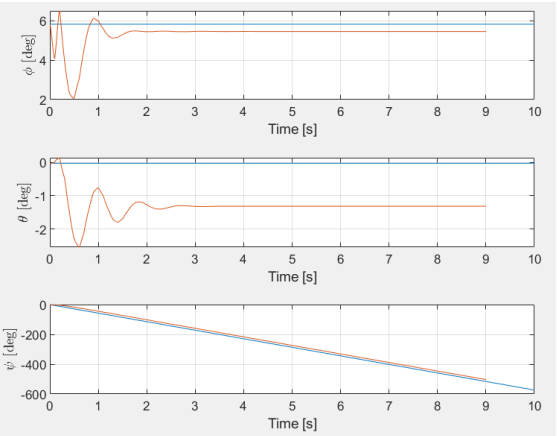

Figure 17: Velocity Values

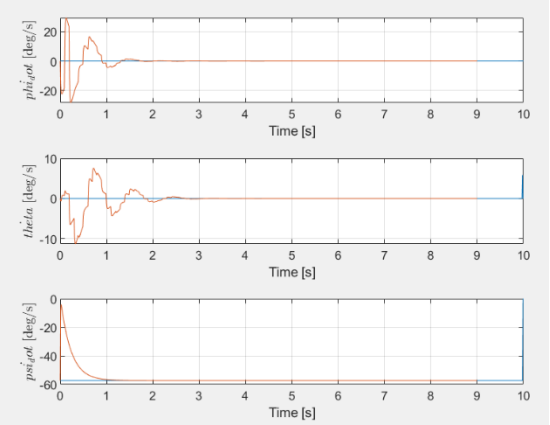

Figure 18: Angular Velocity Values

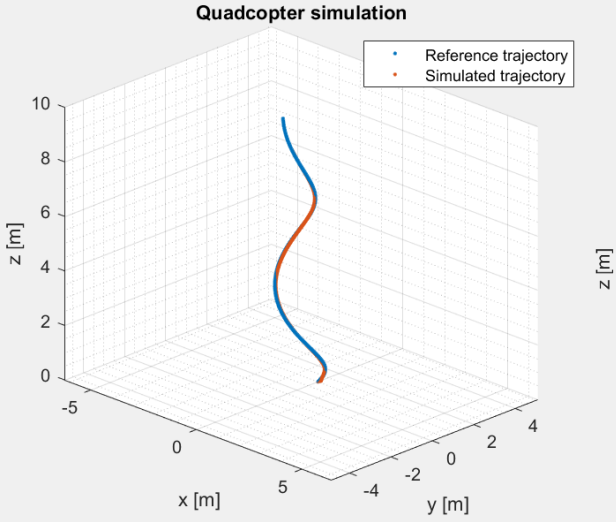

Figure 19: First Trajectory 3D View for 4th Function

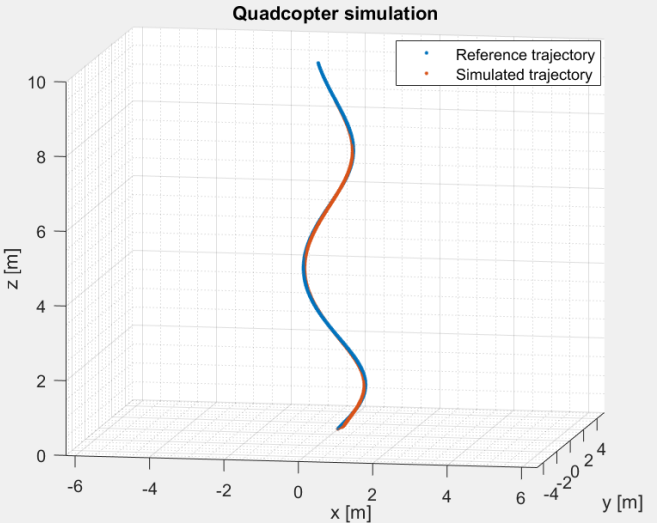

Figure 20: Second Trajectory 3D View for 4th Function

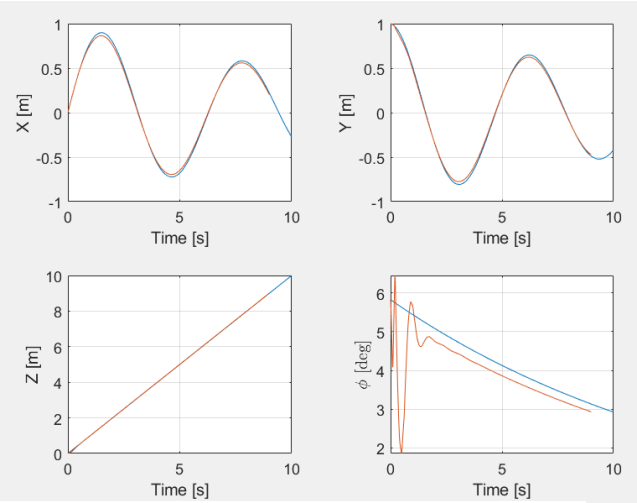

Figure 21: Position and Phi values

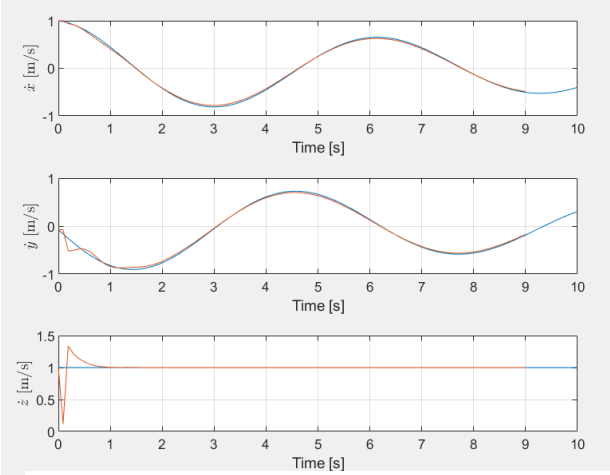

Figure 22: Orientation Values

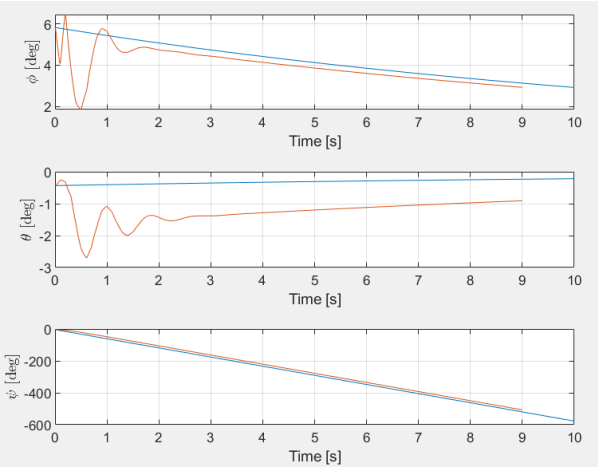

Figure 23: Velocity Values

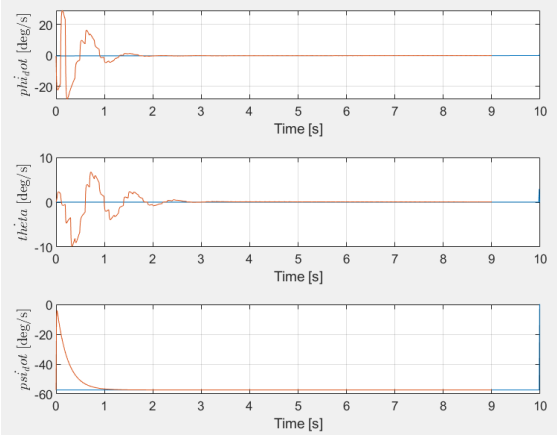

Figure 24: Angular Velocity Values

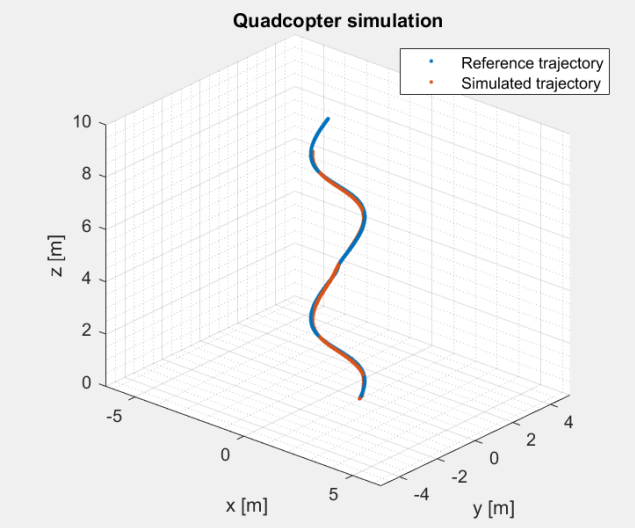

Figure 25: First Trajectory 3D View for 5th Function

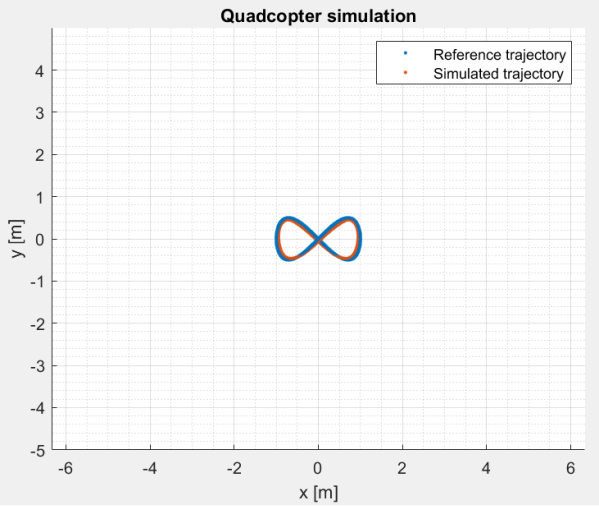

Figure 26: Second Trajectory 3D View for 5th Function

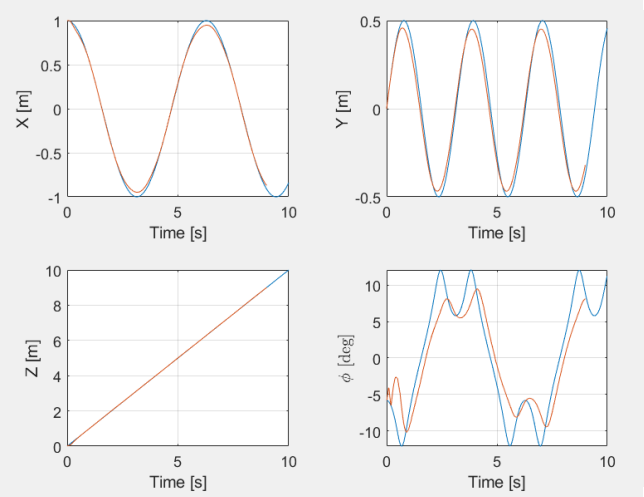

Figure 27: Position and Phi Values

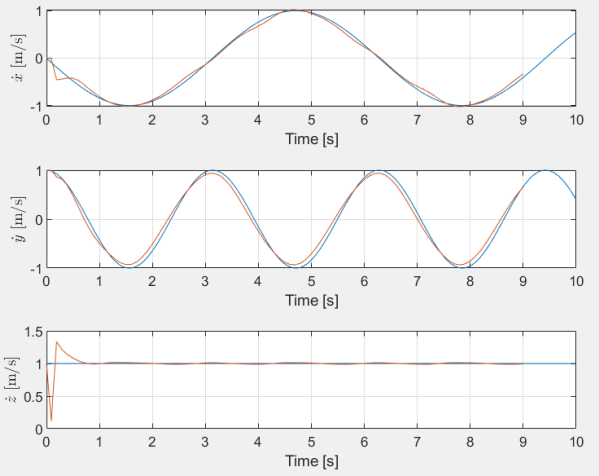

Figure 28: Orientation Values

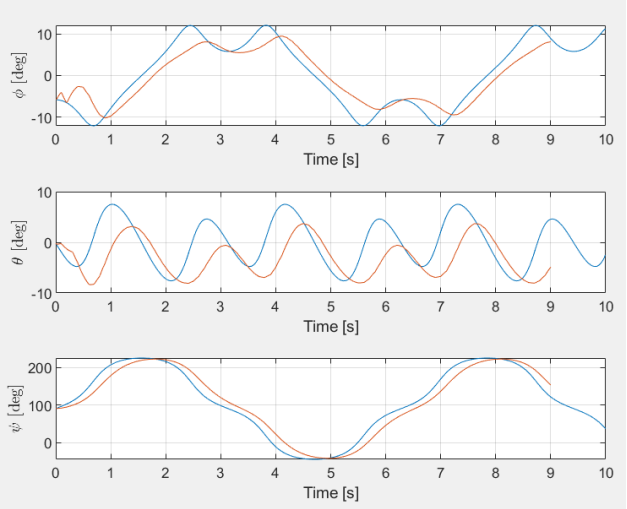

Figure 29: Velocity Values

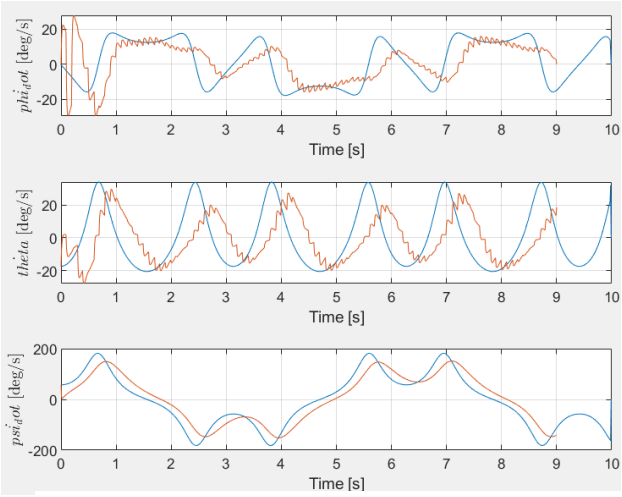

Figure 30: Angular Velocity Values

Next Steps

The results look promising, and I believe that the code works as intended. However, I think that there is still room for improvement in terms of simulation time and trajectory error. There are various functions and toolboxes that I use which might be significantly slowing the process down. I will also talk to other members of the Xu Research Group to see if the results are relatively consistent with the results of the algorithms that they have made in the past. For example, I expect the results to be slightly more accurate than the PID controllers. Once I have verified that the FMPC controller works as intended, I believe the next step is to program the code in a language, like C, such that the MPC could be uploaded to the quadrotor for real-time, on-board trajectory tracking.

Summary of Learnings

While taking part in this research experience, I was able to get insight into the growing field of MPC controllers, a brief understanding of drone dynamics, and a lot of MATLAB experience. In retrospect, I wasted too much time at the start of the semester trying to understand what is going on before getting started. In the future, I will just get started and research things that I do not understand as I go. I will also try to research existing work to have a reference rather than starting from scratch, realizing that my methodology is completely off, and having to start over.