Planar Quadrotor PID Controller Design Report

Samarth Barve

***Abstract*— This report details the design and implementation of a PID controller for a planar quadrotor restricted to motion in the YZ plane. The controller is designed to regulate the quadrotor's position and attitude to follow a specified trajectory accurately.**

I. INTRODUCTION

The objective of this task was initially to develop a PID controller for a UAV that could follow complex trajectories such as sine wave, diamond-like, and straight-line structures. However, due to the complexities involved, the focus was narrowed down to designing a PID controller for a planar quadrotor confined to motion in the YZ plane. The controller aims to regulate the quadrotor's position and attitude accurately to follow a specified trajectory within the YZ plane.

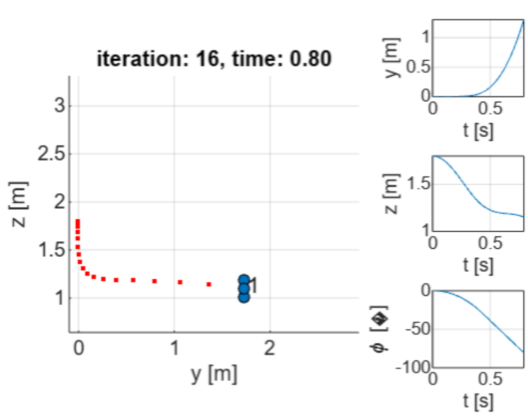
II. CONTROLLER DESIGN

The PID controller is implemented as a MATLAB function named `controller`. It takes the current state of the quadrotor (`state`), the desired state (`des_state`), and additional parameters as inputs and computes the desired controls (`F` and `M`) for position and attitude, respectively.

A. A. PID Controller Constants

The PID controller utilizes the following constants for position and attitude control:

- **Position Control Constants:**
  - Proportional Gain (Kp_{pos}): 4
  - Derivative Gain (Kd_{pos}): 0.1
  - Integral Gain (Ki_{pos}): 0
- **Attitude Control Constants:**
  - Proportional Gain (Kp_{att}): 0.001
  - Derivative Gain (Kd_{att}): 0



B. B. Implementation Details

- **Persistent Variables:** The controller maintains a persistent variable (`prev_error_integral`) for integral error in position control.
- **Initialization:** If the persistent variable is empty, it is initialized to zero.
- **Error Calculation:** Position and velocity errors are computed based on the current and desired states.
- **Integral Error Update:** The integral error is updated using trapezoidal integration.
- **Control Input Calculation:** Control inputs for position (`F`) and attitude (`M`) are computed using PID control formulas.
- **Range Limitation:** The attitude error is ensured to be within the $[-\pi, \pi]$ range.

III. CHALLENGES FACED

Writing the code for the `controller.m` file in MATLAB posed several challenges:

- **Complex Trajectories:** Designing a PID controller capable of accurately tracking complex trajectories such as sine wave, diamond-like, and straight-line structures proved to be highly challenging due to the nonlinearities and uncertainties in the system dynamics.
- **Algorithm Complexity:** Implementing algorithms to compute control inputs for position and attitude while ensuring stability and robustness in various scenarios required careful consideration of system dynamics and controller design principles.
- **Tuning PID Gains:** Tuning the PID gains to achieve desired performance while avoiding instability or overshoot presented a significant challenge, requiring iterative testing and adjustment.

IV. LEARNING OUTCOMES

Through this project, several key learning outcomes were achieved:

- Gain Understanding: Gain insights into the principles of PID control and its application in quadrotor control systems.
- System Modeling: Learn techniques for modeling complex dynamical systems and translating them into control algorithms.
- Algorithm Development: Enhance skills in algorithm development and implementation for real-time control applications.
- Problem Solving: Develop problem-solving skills to address challenges in controller design, tuning, and optimization.

V. Conclusion

The developed PID controller for the planar quadrotor effectively regulates position and attitude to follow a specified trajectory within the YZ plane. Despite the challenges faced, the controller demonstrates satisfactory performance and lays the groundwork for future enhancements and optimizations.

VI. Suggestions for Optimization

- **Parameter Tuning:** Experiment with different values of PID gains to achieve better performance and trajectory tracking.
- **Trajectory Planning:** Implement advanced trajectory planning algorithms to generate smooth and optimal trajectories for the quadrotor.
- **Real-time Control:** Optimize the controller for real-time performance to ensure rapid response to trajectory changes and external disturbances.