Robotics Studio 4: Quadrotor three – Proposal

|  |  |
| --- | --- |
| Tanach Chinbutarnont ID 62340500026 | Nattasit Phaisalrittiwong ID 60340500063 |
| Pakapaka Silpapinun ID 62340500042 |  |
| **Advisor** – Pi Thanacha Choopojcharoen |  |

**Preface**

This project report is a part of FRA361 Studio 4: Open Topic. In this report, we utilize four studies from our 4-year curriculum course: FRA131 Basic Mechanics for Robotics and Automation Engineering, FRA231 Robotics Modeling and Experimentation, FRA233 Control Engineering for Robotics, and FRA333 Kinematics for Robotics System. Apart from the lesson we have learnt, controlling system of quadrotor and sensor data preparation.

This report consists of 5 chapters. The first chapter, introduction, gives an overview of the project including the scope. The second chapter, physical quadrotor model, provides the mathematical explanation and its derivation. The third chapter, sensor model, deals with sensor’ state estimation and filter. The fourth chapter, control design & implementation, shows control algorithm used to stabilize model and simulation for testing the controller’s performance. The simulation was done in MATLAB. The last chapter, conclusion, summarizes the goal of this report and evaluates the performance and result of simulation.

**I. Introduction**

The quadrotor is a 4-rotor Unmanned Aerial Vehicle (UAV), which contains the maneuvering ability of traditional helicopters with lower mechanical complexity [1]. Because of the maneuvering ability, the quadrotor becomes a tool helping users completing desired tasks in a dangerous or inaccessible environment. Furthermore, the quadrotor is usually used [1] in research of developing control laws. To have a better understanding of control laws and its application, our group decided to research this topic.

Regarding to the COVID-19 situation and limited time, we limit the scope of study as follows:

1. The action of a quadrotor is limited to hovering, which means translation in the x-y plane and orientation do not occur.
2. The input of the closed system is height.
3. The quadrotor’s behavior is controlled by torque in the x, y, and z-axis; and force in the z-axis.
4. The study is limited to the simulation process.
5. All physical parameters of the quadrotor are constant.
6. At least two sensor types are used: 6-axis IMU (3-axis accelerometer; and 3-axis gyroscope) and range sensor.
7. The simulation result is visualized as the movement in a 3D plane; graphs of position signal and rotation signal are plotted against time.

**III. Working of Individual System**

*2.1 Environment Setup*

The simulation will be created via MATLAB and Simulink by using UAV toolbox which is capable for designing, simulating, testing, and deploying. To simulate the modeled quadrotor, Simulation 3D UAV Vehicle will used which its input are translation vector and rotation vector.

*2.2 Controlling System*

The controlling system of quadrotor compose of 3 subsystems: attitude control, altitude control, and lateral flight. ระบบความคุม quadrotor ที่ใช้ในการเคลื่อนที่ขึ้น และ ลง จะถูกแบ่งออกเป็น 2 ระบบ นั่นคือ altitude control และ attitude control โดยในระยะนี้จะศึกษาระบบ altitude control ที่ถูกออกแบบโดยใช้ LQR ที่มี PI controller [2] สำหรับ attitude control จะถูกควบคุมโดยใช้ PD controller [3]

*2.3 Sensor Modeling*

ในทางปฏิบัติตัวแปรที่จะส่งผลต่อการทำงานของระบบควบคุมจะได้รับผ่านการส่งข้อมูลจาก sensor ซึ่งในโปรเจคนี้ sensor ที่จะทำการจำลองขึ้นมานั่นก็คือ Inertial Measurement Unit (IMU) โดยจะประกอบไปด้วย 3-axis gyroscope และ 3-axis accelerometer

*2.4 Sensor Estimation*

**IV. Testing case**

ในการทดสอบจะทำการทดสอบการเคลื่อนที่ขึ้นลงที่ระยะต่างๆ ตามที่ต้องการโดยจะทำการจำลองผ่าน Simulink และการ plot กราฟจาก MATLAB

# **References**

|  |  |
| --- | --- |
| [2] | S. Akyurek, U. Kaynak and C. Kasnakoglu, "Altitude Control for Small Fixed-Wing Aircraft Using H∞ Loop-Shaping Method", *IFAC-PapersOnLine*, vol. 49, no. 9, pp. 111-116, 2016. Available: 10.1016/j.ifacol.2016.07.507. |
| [3] | T. Choopojcharoen, "IMPLEMENTATION OF CONTROL & ESTIMATION OF QUADROTOR IN MATLAB", 2016. |