Robotics Studio 4: Quadrotor three – Progress Update I

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

**I. Introduction**

The quadrotor is a four-rotor Unmanned Aerial Vehicle (UAV), which contains of four rigidly attached rotors on the vehicle. Due to its unique rotor’s configuration, a control scheme for quadrotor can be synthesized to maneuver with four degrees of freedom. This maneuverability helps users accomplishing desired tasks in dangerous or inaccessible environment. To gain better understanding of control scheme and its application, our team proposes a project on simulating the physics and behavior of the quadrotor.

The scope of this project consists of the following:

1. The entire system will be developed only in computer-simulated environment.
2. The goal of the quadrotor is to vary its altitude and yaw angle, while maintaining minimum change in lateral position.
3. The input from the user interface is the desired altitude and forward direction of the quadrotor.
4. The control scheme provides 3-dimentional resultant torque and force along the rotational axis of the quadrotor.
5. All physical parameters of the quadrotor are constant and known.
6. At least two sensor types are equipped on the quadrotor: 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.

The model of the quadrotor consists of two main parts: dynamics and kinematics model. The kinematics model equation was constructed in the body frame, while state-space representation is in the global coordinate frame. The controlling system of the quadrotor divides into three subsystems: attitude control, altitude control, and lateral flight control [1]. Sensor models are added to the model to create the closed-loop system. In the physical system, many uncertainties occur, such as model error and disturbance of surroundings. Therefore, the state estimation is required[1].

**II. Quadrotor Model & Control System**

To understand the essence of the quadrotor, we firstly created the quadrotor model and identified its control scheme.

1. *Kinematic Model of Quadrotor*

We utilized the knowledge of FRA333, kinematics, to find the pose and rotation matrix of the quadrotor.

* Rotation (Euler’s Angle)

To control the orientation of the quadrotor, we used Z-Y-X Euler angles to model its rotation in the global frame. In this report, the global frame, G, is defined by axes    with pointing upward [1]. The Body frame, B, is attached to the center of the mass of the quadrotor with pointing in the quadrotor forward direction and perpendicular to the plane of rotors. Therefore, the rotation matrix transforming the global frame to the body frame is given by equation 3,

(3)

where is yaw angle, is pitch angle, is roll angle, and and denote .

Nevertheless, describing orientation in Z-Y-X Euler angles creates singularity issues and complexity while changing the frame reference. Therefore, besides the rotation matrix, all angular variables are in quaternion in this report.

* Orientation Kinematics (Quaternion)

From the research [1], the derivative of quaternion is given by the equation 1.

(1)

The quaternion of the quadrotor can be found by integrating the derivative of quaternion as in figure 1.

Diagram

Description automatically generated

**Figure 1:** block orientation kinematic

Then, the integration was applied to find the quaternion.

* Position Kinematics

Likewise in angular position, the linear position can be found by integrating the input linear velocity.

1. *Dynamics Model of Quadrotor*

From the kinematics equation, the velocity must identify to find the pose of the quadrotor. The inputs of our quadrotor model are torque and force, respecting to global and body frame, respectively. Therefore, before applying the dynamics equation, the input force is transformed into the body frame by a rotation matrix in equation 3.

Diagram

Description automatically generated

**Figure 2:** the overview of the quadrotor model

* Orientation Dynamics

The orientation dynamics of quadrotor can be described as equation 1.

(2)

Diagram

Description automatically generated

**Figure 3:** the Orientation Dynamics block

Then, the integration was applied to find the angular velocity.

* Position Dynamics

For the position dynamics of quadrotor, the linear acceleration, is calculated by using Newton’s second Law of motion.

(3)

Diagram

Description automatically generated

**Figure 4:**  the Position Dynamics block

Then, the integration was applied to find the linear velocity.

1. *Control System*

To control the created model, applied torque and force are calculated.

Diagram

Description automatically generated

**Figure 5:** the overview of the hover control

* Lateral Flight

In this report, the quadrotor flight was controlled by a quintic polynomial trajectory leading to slight changes in position in the earliest stage. These insignificant changes will allow attitude control to perform synchronously with altitude control.

where, is the initial position

is the initial velocity

is the initial acceleration

is the desired position

is starting position

is the maximum acceleration

* Altitude Control

To control the applied force, we control it through acceleration. PID controller was applied to control the desired acceleration from the position error, .

(4)

where,  is a desired position vector.

is a current position vector.

is a current velocity vector.

Diagram

Description automatically generated

**Figure 6:** the overview of Altitude Control block

* Attitude Control

The result of attitude control is desired torque with the orientation as the input. According to the project’s scope, yaw value has already inputted through the desired forward direction. Linearization applies to equation 5 to find roll and pitch.

(5)

**Linearization**

*Diagram, schematic

Description automatically generated*

**Figure 7:** the overview of Desire Angle block

The quaternion-based attitude control is used, where and are the quaternion error and angular velocity error, respectively.

(6)

**Figure 8:**  the overview of Attitude Control Block

To observe the result, we connect all blocks and visualize the movement in a 3D plane via MATLAB.Diagram

Description automatically generated

**Figure 9:** quadrotor model

Chart

Description automatically generated

**Figure 10:** the visualization of quadrotor movement in a 3D plane

**III. Sensor Model & State-Space Estimation**

In the real physical system, all dependent variables are inputted from sensor, which leads to the occurrence of uncertainties.

1. *Sensor Model*

In this report, two sensor types are equipped on the quadrotor: 6-axis IMU (3-axis accelerometer; and 3-axis gyroscope) and range sensor (ultrasonic).

* Ultrasonic Sensor

Due to the possibilities of rotation in roll and pitch, the sensor equation can be described as follow,

*=*

Ultrasonic measures the length along its z axis. Therefore, the ultrasonic measured distance, , can be calculated as in equation 7.

(7)

where is the actual position vector along z axis

is a measured position vector

is a measured position vector along z axis

* Accelerometer

The accelerometer is equipped on the quadrotor, such that the measured acceleration, , is calculated in the body frame as in the following equation.

(8)

To find the quadrotor acceleration, , the Newton’s second laws is used.

Therefore,

(9)

where, is measured acceleration

is a quadrotor generated acceleration vector

is a propellers’ generated force

is a propellers’ generated thrust

is a measured acceleration along z axis

* Gyroscope Sensor

Due to the quadrotor model, the torque will only apply around z axis, therefore the measured angular velocity around z axis, , is quadrotor angular velocity around z axis, .

Thus,

(10)

1. *Estimator*

The estimator constructs with 2 main parts: position estimator and orientation estimator. In position estimation, the Kalman Filter is used to estimate the quadrotor’s position, while the Extended Kalman Filter is used to estimate the orientation.

* Position Estimator

The estimation of position, velocity, and acceleration equation is in the discrete-time domain. The Kalman Filter can be described in equation 11.

(11)

where, is a state of the quadrotor

is a input of the quadrotor  
 is a output of the quadrotor

is a process noise of the quadrotor

is a measurement noise of the quadrotor

and are parameters

* Orientation Estimator

- Euler’s rate

find the angular velocity of each axis,

From

The state-space model of the quadrotor orientation can be written as these following equations.

From those equations, they are a non-linear equation. Therefore, the Extended Kalman Filter is used to estimate the quadrotor orientation

1. *Implementation*

* Position Estimation

In the model, we create the Kalman filter by inserting the Kalman filter block with the calculated parameters and set the sample rate to 100 Hz. To test the performance of the Kalman filter, the noise is added into the system.

Chart, line chart

Description automatically generated

**Figure 11:** the graph of position along z axis after adding noise and position estimator into the system.

* Orientation Estimation

The orientation Estimation is in the validating process. Therefore, the Orientation Estimation block is not yet created.

# **References**

[1] T. Choopojcharoen, “IMPLEMENTATION OF CONTROL & ESTIMATION OF QUADROTOR IN MATLAB”, 2016.

**Appendix A: Position State-Space Identification**

From

where, is the acceleration

is the discrete-time acceleration

is mass

is the rotation matrix from Body-frame to Global-frame

is the discrete-time rotation matrix from Body-frame to Global-frame

is thrust

is gravity

Because of

*d*=  *d*

at

at

(1)

Replace in equation (1)

+

Because of

=  *d*

=

at

at

(2)

Replace in equation (2)

Therefore

Because of position estimator interest just 2 sensors which are ultrasonic sensor and accelerometer (, respectively. The equation can be rewritten it in form of state

From

Rearrange them in form of state variables

where, is the next step of i-axis position where i = x, y, z

is the next step of i-axis velocity where i = x, y, z

is the next step of i-axis acceleration where i = x, y, z

is the discrete-time of i-axis position where i = x, y, z

is the discrete-time of i-axis velocity where i = x, y, z

is the discrete-time of i-axis acceleration where i = x, y, z

**Appendix B: Estimation of Kalman Filter’s Parameter**

Rearrange the equation of state and output equation in form of Kalman filter

From (11)

(12)

Define and

From Output equation, which is sensor equation from Appendix A, it cannot rearrange into the equation which its variables are thrust and gravity. Because sensor equation doesn’t have thrust variables, we must combine them in matrix form,

Where is

Rearrange the above equation

Then, substitute =

And can be calculate from

Because of gravity is one of inputs of this system

Therefore,

All Kalman filter variables are ready to write in their form

Therefore

A in the equation 11 is

B in the equation 11 is

C in the equation 11 is

D in the equation 11 is

G in the equation 12 is

where, = and =

**Appendix C: Orientation State-Space Identification**

Prove the equation for Orientation

From

Define that and

Re-write in term of discrete time

Assume that is (rate of change of Angular acceleration) is a gaussian distribution with 0 mean

*d* =  *d*

at

at = + ;

*d d*

at

at

Therefore,

where, is the next step of Angular velocity

is the next step of Angular acceleration

is the next step of Euler’s orientation

is the discrete time of Angular velocity

is the discrete time of Angular acceleration

is the discrete time of rate of change of Angular acceleration

is the discrete time of Euler’s orientation

is the Jacobian of orientation and its rate of change

The result of equation will lead to do Extended Kalman Filter because equation is non-linear type

Planning Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TASK | Description | RESPONSIBLE | START | FINISH | STATUS |
| 1 | Planning and Listing all tasks |  | everyone | 24/Aug/21 | 30/Aug/21 | 100% |
| 2 | Modelling | | | | |  |
| Controller's Model | Altitude Control | Pakapak | 31/Aug/21 | 7/Sep/21 | 100% |
| Attitude Control | 100% |
| Lateral Control | 100% |
| Dynamic Model | Motion | 8/Sep/21 | 14/Sep/21 | 100% |
| Integrate Modelling |  | 15/Sep/21 | 21/Sep/21 | 100% |
| 3 | Estimation | | | | |  |
| Range Sensor | State Estimation | Tanach&Nattasit | 31/Aug/21 | 7/Sep/21 | 100% |
| IMU Sensor | State Estimation | 8/Sep/21 | 14/Sep/21 | 100% |
| Integrate estimation of IMU and Range sensor |  | 15/Sep/21 | 21/Sep/21 | 100% |
| Kalman filter | Do kalman filter of state estimation of IMU and Range sensor | 25/Sep/21 | 75% |
| 4 | Simulation and Visualization |  |  |  |  |  |
| Do Plant |  | Tanach&Nattasit | 22/Sep/21 | 28/Sep/21 | 100% |
| Controller | PID Tuning | 100% |
| 3D-plot graph |  | 29/Sep/21 | 5/Oct/21 | 100% |
| 5 | Further Develop | Visualization Develop | everyone | 6/Oct/21 | 19/Oct/21 | 25% |
| Adding sensor | 25% |
| 6 | Presentation |  |  |  |  |  |
| Rechecking |  | everyone | 20/Oct/21 | 26/Oct/21 | 0% |
| report and presentation |  | 27/Oct/21 | 24/Nov/21 | 0% |
| Proposal Presentation |  | 9/Sep/21 | 9/Sep/21 | 100% |
| Progress Presentation |  | 19/Oct/21 | 19/Oct/21 | 100% |
| Final Presentation |  | 25/Nov/21 | 25/Nov/21 | 0% |