Dynamic Modeling and Simulation of A Quadcopter with Motor Dynamics

A Ram (Bella) Kim, Iowa State University

Available at: https://works.bepress.com/arambella-kim/4/
Dynamic Modeling and Simulation of A Quadcopter with Motor Dynamics

A Ram Kim*  Prasanth Vivekanandan†  Patrick McNamee‡  Ian Sheppard‡  
Aaron Blevins*  Alex Sizemore*

School of Engineering, The University of Kansas, Lawrence, KS, 66045, USA

Increased usage of quadcopters as an embedded system has raised the necessity of developing high quality dynamic models for designing controllers and simulations. In this work, a dynamic model was developed for a quadcopter platform by conducting experiments for estimating the moments of inertia and for determining motor dynamics. The dynamic model consisted of equations of motions derived from Newton and Euler’s approach. Moments of inertia were estimated by operating a bifilar pendulum test. The motor dynamics were developed using a custom built test bed. The thrust, torque, voltage, and current were recorded in order to develop an adaptive dynamic model with respect to battery voltage. Following the completion of the dynamic model, the state space model was developed in linear time invariant condition so that it could be utilized for designing a controller for simulation purposes. The system was flight tested and parameter estimation was performed on the thrust and torque motor coefficients.

Nomenclature

\[ \omega \] Angular speed of motors, rad/sec
\[ P, Q, R \] Angular velocity in body coordinate, rad/sec
\[ \vec{u} \] Control vector, -
\[ m \] Mass, kg
\[ \tau \] Moment generated by motors, N \cdot m
\[ p, q, r \] Perturbed Roll, Pitch, Yaw rate, rad/sec
\[ P_N, P_E, P_D \] Position in inertial coordinate, m
\[ \phi, \theta, \psi \] Roll, Pitch, Yaw angle, rad
\[ \vec{X} \] State vector, -
\[ I_{xx} \] X direction Moment of Inertia, kg \cdot m^2
\[ I_{yy} \] Y direction Moment of Inertia, kg \cdot m^2
\[ I_{zz} \] Z direction Moment of Inertia, kg \cdot m^2
\[ U, V, W \] Velocity component in body coordinate, m/sec
\[ T \] Thrust generated by motors

Subscript
1 Steady state
B Body coordinate system

I. Introduction

Unmanned aerial systems have high potential in civil and military applications such as mapping, reconnaissance, search and rescue, and agriculture. Multirotor systems also have the advantage of vertical take off

---

*Graduate student, Department of Aerospace Engineering, The University of Kansas, Lawrence, KS, 66045, USA.
†Graduate student, Department of Electrical Engineering and Computer Science, The University of Kansas, Lawrence, KS, 66045, USA.
‡Undergraduate student, Department of Aerospace Engineering, The University of Kansas, Lawrence, KS, 66045, USA.
and landing. Recently, many embedded systems have become available for multirotor vehicles at reasonable cost. Quadcopters are the most basic and common platform of multirotor systems and the necessity of modeling and simulation is increasing with their applications.

Al-Saedi and Sabar\textsuperscript{1} used equations of motion ruled by Newtonian mechanics. They modeled the quadcopter system using a plus(+) configuration, in which one motor is aligned along the x axis in the body coordinate system. A fuzzy logic controller was designed to simulate the quadcopter. Balasubramanian and Vasantha\textsuperscript{2} used the kinematic and Euler-Lagrange equation for the equations of motion. Compute torque control was designed for the simulation. This control method uses the torque as the input based on the model. This method is known as non-linearity cancellation technique. Elruby \textit{et. al}\textsuperscript{3} uses Newton-Euler approach to form the equations of motion. In this work, the moment of inertia was estimated using CAD software. The PID controller was designed to simulate the dynamic but the altitude tracking did not obtain a satisfactory quality. The biggest drawback was that their dynamic model did not include the motor dynamics. Carrillo \textit{et. al}\textsuperscript{4} proposed very detailed of equations of motion for Euler-Lagrange and Newton-Euler approach. In addition, the plus and cross (or X) configuration for quadcopter platform was discussed. Hoffman \textit{et. al}\textsuperscript{5} demonstrated the test bed for measuring thrust and torque using a load cell. In this work, the thrust model was mapped to angle of attack and flight speed. The aerodynamic and blade flapping effects were reflected in the dynamic model. The flight test results showed well-developed dynamic model and control. Ji and Trukoglu\textsuperscript{6} used Newton-Euler equations of motion. The drag due to the propeller was included using Blade Element Theory. The hardware configuration proved to be very cost effective through the use of an Arduino. The moment of inertia was calculated using the mathematical derivation of moment of inertia. Thrust and torque was modeled using Blade Element Analysis. Kader \textit{et. al}\textsuperscript{7} used Euler-Lagrange formalism and designed PID and fuzzy logic controllers to simulate. Pounds \textit{et. al}\textsuperscript{8} included blade flapping effect on Newton-Lagrange equations of motion. Moment of inertia was calculated based on each component. The controller was designed based on the pole-zero cancellation method in the frequency domain transfer function.

In this research, the dynamic model was developed for a quadcopter platform using an experimental approach of determining the moments of inertia and motor dynamics. The moments of inertia were estimated via a bifilar pendulum test\textsuperscript{9} and motor characteristic testing was conducted to measure thrust and torque. Two algorithms were also developed in order to make the dynamic model adaptive with respect to the voltage of the onboard battery. The state space model was developed to aid in controller design and simulation for future research.

\section{Methods}

\subsection*{A. Equations of Motion for Quadcopter}

The equations of motion are fundamental to dynamic modeling. Equations are the mathematical representation of the system and contains the physical constraints of the dynamic system. To initiate the dynamic modeling process of a quadcopter, it is imperative to investigate the equations of motion so that the proper procedure of modeling can be selected and considered. To begin with, the coordinate system should be defined, in this case see Figure 1.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{coordinate_system.png}
\caption{Coordinate system. Left figure shows the inertial coordinate system and right figure shows the body coordinate system and motor numbering.}
\end{figure}

The global coordinates are in the inertial coordinate system. The body coordinate system is established as
an "X" configuration in this work. However, the body coordinate can be defined freely with any configuration of quadcopters (e.g. "+" configuration). In addition, multicopters with more than four motors can also be modeled using proper free body diagrams. Figure 2 shows motor numbering and the direction of motor rotations in order to clarify the thrust and torque directions in the equations of motion. The following equations are the equations of motion for general quadcopters in the body coordinate system.

\[
\begin{align*}
\dot{U} &= R \dot{V} - Q \dot{W} - g \sin \theta \\
\dot{V} &= -RU + PW + g \sin \phi \cos \theta \\
\dot{W} &= QU - PV + g \cos \phi \cos \theta + \frac{(T_1 + T_2 + T_3 + T_4)}{m} \\
\dot{\phi} &= P + \tan \theta(Q \sin \phi + R \cos \phi) \\
\dot{\theta} &= Q \cos \phi - R \sin \phi \\
\dot{\psi} &= (Q \sin \phi + R \cos \phi) / \cos \theta \\
\dot{P} &= \frac{QR(I_{yy} - I_{zz})}{I_{xx}} - \frac{l_y(T_1 + T_2 - T_3 - T_4)}{I_{xx}} \\
\dot{Q} &= \frac{PR(I_{zz} - I_{xx})}{I_{yy}} - \frac{l_x(T_1 - T_2 - T_3 + T_4)}{I_{yy}} \\
\dot{R} &= \frac{PQ(I_{xx} - I_{yy})}{I_{zz}} + \frac{(-\tau_1 + \tau_2 - \tau_3 + \tau_4)}{I_{zz}}
\end{align*}
\]

\(P_N, P_E, \text{ and } P_D\) indicates north, east, and down position in the inertial coordinate system, respectively. \(l_x\) and \(l_y\) is the moment arm of the quadcopter from the center of gravity to the motors along the indicated axis in meters. \(T\) is the thrust generated by each motor. \(\tau\) is the torque of the motor. The subscript of thrust and torque indicate motor numbering. The thrust and torque were modeled as quadratic equations but they can also be determined using a higher order of polynomial for more accuracy. The following equation shows the thrust and torque model.

\[
\begin{align*}
T_i &= -b_i \omega_i^2 \\
\tau_i &= d_i \omega_i^2
\end{align*}
\]

To complete the thrust and torque model, the coefficient of Eq. 13 and 14 and the moments of inertia should be determined. In this work, the motors were tested with a custom test bed and analyzed to determine those coefficients.

B. Linear Time Invariant State Space Model

To design the controller for use in simulation, the state space model for the system should be developed. In this work, the linear time invariant state space model was developed and used for designing an LQR controller. To derive a linear time invariant state space model, the equations of motion should be linearized. The trim condition was set as the hovering condition, in which body velocity and angular rates were trimmed to zero as the system maintains an altitude. The state and control vectors were defined as follows.

\[
\begin{bmatrix}
\phi \\
\theta \\
p \\
q \\
r
\end{bmatrix}
\quad \text{and} \quad
\begin{bmatrix}
\Delta \omega_1 \\
\Delta \omega_2 \\
\Delta \omega_3 \\
\Delta \omega_4
\end{bmatrix}
\]

\(\Delta \omega\) indicates the perturbed motor angular rate. Subscripts shown indicate motor numbering. The linear time invariant state space model is defined as follows.

\[
\dot{x} = A x + B u
\]
For the linearization of equations of motion, the state space model was perturbed and the trim condition states were evaluated. Following this procedure, the following equations were obtained.

$$
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{p} \\
\dot{q} \\
\dot{r}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\phi \\
\theta \\
p \\
q \\
r
\end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
-2l_1 b_1 \omega_{i1} & -2l_2 b_2 \omega_{i2} & 2l_3 b_3 \omega_{i3} & 2l_4 b_4 \omega_{i4} & -2I_x \\
-2l_1 b_1 \omega_{i2} & -2l_2 b_2 \omega_{i2} & 2l_3 b_3 \omega_{i3} & 2l_4 b_4 \omega_{i4} & -2I_y \\
-2d_1 \omega_{i1} & 2d_2 \omega_{i2} & -2d_3 \omega_{i3} & 2d_4 \omega_{i4} & I_x \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\Delta \omega_1 \\
\Delta \omega_2 \\
\Delta \omega_3 \\
\Delta \omega_4
\end{bmatrix} 
$$

Subscripts on the motor angular rate, \(\omega_{i}\), defines the motor angular rate at the steady state condition for the respective motor.

C. Hardware configuration of Quadcopter

The quadcopter used in this dynamic modeling process consisted of a stock carbon fiber airframe, with 3D printed avionics enclosures. Four programmable electric speed controllers (ESCs) were used to drive four electric motors equipped with 12x5.5 inch carbon fiber propellers with blended winglet designs. Motors 1 and 3 were programmed to rotate clockwise, while motors 2 and 4 were programmed to rotate counterclockwise. A 14.8 Volt battery with 3200 mAH capacity was used to power the system. The system also contained a Tegra K1, IMU, RC receiver, GPS antenna, and a CC3D flight controller used for RC flight. The total mass of the quadcopter was 2.6 kg.

D. Moment of Inertia Estimation

To complete the dynamic modeling process, the moments of inertia need to be estimated. The bifilar pendulum was used to obtain the moments of inertia. Figure 3 shows the bifilar testing apparatus and configuration. \(L\) is the length of thread connecting the object to the stand. \(b\) is the distance from the center of the object to the location where the thread was attached. To perform a proper bifilar pendulum test, the simple pendulum test should be operated in advance. This is a very important procedure since for a proper bifilar pendulum test, the simple pendulum test should be long enough to perform an accurate test. With the simple pendulum test, the gravity can be back-computed so that the length of the thread, \(L\) can be determined properly. The gravity can be obtained with the following equation.

$$
\omega_{\text{simple pendulum}} = \sqrt{\frac{g}{L}}
$$

\(\omega_{\text{simple pendulum}}\) is the period of the simple pendulum motion. After the proper length of the thread was determined, the bifilar pendulum test can be performed. The time required for ten oscillations were recorded when the pendulum was perturbed. The test was repeated twenty times to have valid statistical analysis. The vehicle was hung in a different way for each axis so that each of the respective body axes could be measured. Once all the data was collected, the moments of inertia were estimated with following equation.

$$
I = \frac{mgT^2b^2}{4\pi^2L}
$$

\(m\) is the mass of the quadcopter. \(g\) is the gravity constant. \(T\) is the period, which was computed as the total time measured divided by ten, the number of oscillations.

1. Testing Setup for Moment of Inertia

Figure 4 shows the actual test setup for each body axis for the moments of inertia.

The mass of the quadcopter was measured to be 2.6 kg. The length of the thread, \(L\), was chosen as 1.96 m and the distance between the center of quadcopter to the engine, \(b\), was 0.2 m. From the simple pendulum test, the times for 10 oscillations were 28.19, 27.55, and 28.58 seconds for \(x_B\), \(y_B\), and \(z_B\) axis, respectively. The gravity constant was obtained as 9.46, 9.91, and 9.45 \(m/s^2\) for \(x_B\), \(y_B\), and \(z_B\) axis, respectively. The bifilar pendulum tests were conducted with this configuration and results were presented in Sec. III.A.
E. Thrust and Torque Modeling

To complete the thrust and torque modeling, the motors should be tested to record output RPM at various input PWM levels. From the hardware configuration, the motors will operate by receiving PWM signal from the electrical speed controller (ESC). To measure the thrust and torque, the test bed was developed, shown in Figure 5. In addition to PWM signals, the battery voltage, current coming from the ESC, force generated from each motor, and the torque acting on the whole airframe from each motor were recorded to develop an adaptive thrust and torque model for various battery voltage levels. For the quadcopter, the dynamics are solely based on the motor dynamics. Though the dependency of motor dynamics is very high, the importance of motor performance is often assumed as identical throughout the flight. The battery voltage drops as the flight time increases and this will effect the performance of different motors. Since the voltage of the battery is dropping, the thrust generated from each motor with identical PWM signal will be less than the optimal thrust. To compensate for the drop in battery voltage, the torque and thrust model was developed in the way of adjusting PWM command automatically. Figure 6 shows the installation of the quadcopter to the testing bed.
1. Testing Setup for Thrust and Torque

The test stand used in this experiment comprised of two computing platforms: A Raspberry PI 2 and an Arduino Mega board. The Raspberry PI 2 is a powerful computing device compared to the Arduino, and it controls the test bed. The Raspberry PI 2 board played the role of processing and recording data from the Arduino. The Arduino Mega acts as the data acquisition and was used to send the electric signals to the electric speed controllers. The sensors and motors are connected to the Arduino Mega and it has bidirectional serial communication with the PI. The pseudocode of the Arduino Mega and Raspberry PI are shown in Figure 7 and Figure 8, respectively. PI sends the PWM to the arduino. The arduino drives the motors based on the PWM and then reads all of the sensor outputs. The sensor data from the Arduino is then sent to the PI and the test stand data are logged into a file.

```c
1  void loop()
2  {
3    // get the pwm for motor from PI
4    motor_pwm = get_from_PI();
5    // send the pwm to motor
6    servo_write(motor_pwm);
7    // basic sensor input
8    current_sensor_data = i2c_read_current_sensor();
9    send_to_PI(current_sensor_data);
10   rpm_data = measure_rpm_motor();
11   send_to_PI(rpm_data);
12   voltage_data = measure_voltage();
13   send_to_PI(voltage_data);
14   thrust = measure_thrust();
15   send_to_PI(thrust);
16   torque = measure_torque();
17   send_to_PI(torque);
18    // real time property
19   sleep_until_next_period();
20  }
```

**Figure 7:** Arduino Logic for Thrust and Torque Testing

```c
1  void main()
2  {
3    // mode selection
4    /*
5    * mode 1 - High PWM to Low PWM
6    * mode 2 - Low PWM to High PWM
7    * mode 3 - High PWM to 50% PWM
8    * mode 4 - 50% PWM to Low PWM
9    */
10   // User selects the mode through the laptop connected to the PI
11   mode = get_from_user();
12   // entering into a loop
13   while (loop){
14     // send the PWM to arduino
15     switch (mode) {
16       case 1: send_to_arduino(High_to_Low); break;
17       case 2: send_to_arduino(Low_to_High); break;
18       case 3: send_to_arduino(High_to_50%); break;
19       case 4: send_to_arduino(50%_to_Low); break;
20     }
21    // get sensor values from arduino
22    current_sensor_data = get_from_arduino(current_sensor);
23    tachometer_data = get_from_arduino(rpm);
24    voltage_data = get_from_arduino(battery_voltage);
25    thrust_data = get_from_arduino(thrust);
26    torque_data = get_from_arduino(torque);
27    // log the data
28    write_to_log_file(time, iteration, current_sensor_data, tachometer_data, voltage_data,
29      thrust_data, torque_data, motorPWM);
30    // real time property
31    sleep_until_next_period();
32  }
```

**Figure 8:** PI 2 Logic
2. Adaptive Thrust and Torque Modeling

The thrust and torque generated from an electric motor is dependent upon the battery voltage. As battery voltage drops during flight, so too will the thrust and torque of the motors drop. In order to be adaptive to this characteristic, the motor test was conducted at different PWM levels and several battery voltage levels. This indicates that the coefficient of the thrust and torque model (Eq. 13 and 14) would be different with varying voltage. Data gathering from the test were processed to fit the quadratic curve between thrust, torque, and RPM at the different voltage level. With these curves, the following algorithms were developed.

a. Algorithm 1: Mapping trim thrust PWM command

This algorithm is used to find the trim thrust PWM command since a higher PWM command is required at a lower voltage level. This was necessary because the steady state of the quadcopter was chosen as the hovering condition. The procedure of this algorithm is as follows.

1. With given weight of the quadcopter, the required thrust was computed for each motor. In this case, the quadcopter has four motors so weight was divided by four.
2. With the current voltage level of battery, evaluate the curve between thrust and voltage. At this stage, the thrust data at different PWM will be obtained at the given battery voltage.
3. The required thrust can then be used to find the correct PWM command using linear interpolation between PWM level of each thrust curve.

b. Algorithm 2: Mapping thrust and torque PWM command with varying voltage

In this algorithm, the thrust and torque coefficient will be computed with the current voltage level.

1. Using the given battery voltage, evaluate the thrust and torque from curves between thrust, torque, and voltage.
2. With the current voltage level of battery, evaluate the curve between thrust and voltage. At this stage, the thrust data at different PWM will be obtained at the given battery voltage.
3. The required thrust can then be used to find the correct PWM command using linear interpolation between PWM level of each thrust curve.

These procedures were described in more detail with graphical demonstration in Sec. III.B.

F. Simulation of Dynamic Modeling using LQR controller

To observe the behavior of the quadcopter, the simulation was performed using an LQR controller. The LQR controller was designed to correct the attitude and level the quadcopter in a hovering condition. To design the LQR controller, the augmented state space was formed as the following.

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{p} \\
\dot{q} \\
e_\phi \\
e_\theta
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\phi \\
\theta \\
p \\
q \\
e_\phi \\
e_\theta
\end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-2l_x b_1 \omega_{d1} I_x & -2l_x b_2 \omega_{d2} I_x & 2l_x b_3 \omega_{d3} I_x & 2l_x b_4 \omega_{d4} I_x & 2l_y b_1 \omega_{d1} I_x & -2l_y b_2 \omega_{d2} I_x \\
-2l_x b_1 \omega_{d1} l_y & -2l_x b_2 \omega_{d2} l_y & 2l_x b_3 \omega_{d3} l_y & 2l_x b_4 \omega_{d4} l_y & -2l_y b_1 \omega_{d1} l_y & 2l_y b_2 \omega_{d2} l_y \\
-2l_x b_1 \omega_{d1} l_z & -2l_x b_2 \omega_{d2} l_z & 2l_x b_3 \omega_{d3} l_z & 2l_x b_4 \omega_{d4} l_z & -2l_y b_1 \omega_{d1} l_z & 2l_y b_2 \omega_{d2} l_z \\
-2l_x b_1 \omega_{d1} & -2l_x b_2 \omega_{d2} & 2l_x b_3 \omega_{d3} & 2l_x b_4 \omega_{d4} & -2l_y b_1 \omega_{d1} & 2l_y b_2 \omega_{d2}
\end{bmatrix}
\begin{bmatrix}
\Delta \omega_1 \\
\Delta \omega_2 \\
\Delta \omega_3 \\
\Delta \omega_4
\end{bmatrix}
\]

(20)

where \(e_\phi\) and \(e_\theta\) were error in roll and pitch angle. The LQR controller provided the correction of the angular speed of motors from the trim angular speed. From the guidance perspective, the roll and pitch angle was commanded to zero to level the quadcopter. To hover, the height tracking was also necessary. To satisfy the height tracking, a PID control approach was applied in the guidance.

\[
\Delta \omega_{\text{height}} = K_p \cdot e_{\text{height}} + K_i \cdot \int e_{\text{height}} dt
\]

(21)

where \(e_{\text{height}}\) was the error in height, \(K_p\) was the proportional gain which was selected as -80, \(K_i\) was the integral gain and was selected as -200.

Finally, the angular speed command was calculated as the following.

\[
\omega_{\text{cmd}} = \omega_{\text{trim}} + \Delta \omega_{\text{height}} + \Delta \omega_{\text{LQR}}
\]

(22)

The commanded angular speed was applied to equations of motion so the behavior of the quadcopter could be observed.
G. Parameter Estimation using System Identification

Flight testing of the quadcopter platform involved the use of a CC3D flight controller and an RC pilot. State measurements were obtained from an onboard IMU and an extended Kalman filter was used for in-flight data compatibility. Parameter estimation utilized System Identification Programs for AirCraft (SIDPAC), a collection of MATLAB programs for system identification by NASA, which has been used for system identification on numerous aircraft.

III. Result

A. Moment of Inertia Testing Result

Figure 9 shows the result of bifilar pendulum test. The time for ten oscillations of bifilar pendulum motion to occur was recorded and used to compute the moments of inertia using Eq. 19 for each sample. The test was repeated twenty time for valid statistical analysis.

![Figure 9: Result of Bifilar pendulum test. (a) Time measured during 10 oscillations of Bifilar pendulum motion. (b) Result of moment of inertia for each axis in Body coordinate system.](image)

Table 1: Result of Moment of Inertia

<table>
<thead>
<tr>
<th></th>
<th>$I_{xx}$ [kg \cdot m^2]</th>
<th>$I_{yy}$ [kg \cdot m^2]</th>
<th>$I_{zz}$ [kg \cdot m^2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>3.63</td>
<td>3.25</td>
<td>5.51</td>
</tr>
<tr>
<td>STD</td>
<td>0.06</td>
<td>0.14</td>
<td>0.31</td>
</tr>
<tr>
<td>Variance</td>
<td>0.1</td>
<td>0.0036</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 1 shows the numerical result of moment of inertia for each axis. The collected data appears reliable since the standard deviations were very low in all axis, as shown in table 1. Generally, the moments of inertia in the x and y axis were expected to be identical due to the symmetric geometry. However, in this work, the makeup of the internal hardware might have caused the asymmetric moment of inertia in those axis due to the asymmetric mass distribution. Though the moments of inertia in those axis are not identical, they are very close.

B. Thrust and Torque modeling result

In Figure 10, the collected thrust and torque test data were shown. The tests were operated within the PWM range from 1000 to 2000, in increments of 100. The entire PWM range was repeated four times in order to record the thrust and torque at the different levels of battery voltage. With the collected data, the algorithms mentioned in Sec. 2 was used. The algorithm that finds the trim angular speed of the motor requires the voltage versus thrust graphs, which are shown as (a) in Figure 10. Once the sensor reads the voltage level of the battery, the thrust force will be evaluated at each PWM value. The required force for the hover position, which is the weight of the system, can be used to determine what PWM value is required to generate the required thrust. For exact trim thrust PWM command, the linear interpolation was used to find the trim PWM command. The results are shown in Table 2.

![Figure 10: Result of Thrust and Torque Testing. (a) Graph showing the relationship between voltage and thrust. (b) Graph showing the relationship between battery voltage and torque.](image)

Higher trim angular speeds were commanded in order to compensate for lower battery voltages. With this algorithm, the hovering state can be maintained by adjusting the PWM command automatically with the voltage level. To utilize the algorithms developed, the collected data was estimated with the linear curve so that the thrust and torque can be determined given the PWM and battery voltage. For the algorithm of thrust and torque constant, see (a) and (c) graphs shown in Figure 10. With the current voltage, the thrust and torque can be evaluated with the respective data set. This data will be fitted to the quadratic curve which can be formed using Eq. 13 and 14. The result of these constants are shown in Table 3. As a result,
the coefficients were higher as the battery voltage increased. This proved similar with collected data trends. With this coefficient, the thrust and torque can be computed correctly with varying battery voltages. This provides an adaptive response from the system to counteract changing motor dynamics due to variances in battery voltage.

Table 3: Thrust and Torque Coefficient Numerical Results at Different Battery Voltages

<table>
<thead>
<tr>
<th>Motor</th>
<th>Thrust Coefficient</th>
<th>Torque Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor 1</td>
<td>16 V: 2.86 \cdot 10^{-5}</td>
<td>16 V: 8.89 \cdot 10^{-7}</td>
</tr>
<tr>
<td></td>
<td>15 V: 2.5 \cdot 10^{-5}</td>
<td>15 V: 7.7 \cdot 10^{-7}</td>
</tr>
<tr>
<td></td>
<td>14 V: 2.13 \cdot 10^{-5}</td>
<td>14 V: 6.5 \cdot 10^{-7}</td>
</tr>
<tr>
<td></td>
<td>13 V: 1.77 \cdot 10^{-5}</td>
<td>13 V: 5.3 \cdot 10^{-7}</td>
</tr>
</tbody>
</table>

C. LTI State space model

Using the processed data, all coefficients were determined in order to complete the linear time invariant state space model in Eq. 17. The following equation shows the completed LTI state space model for the system utilizing a battery at 16V. The state space model can be derived at any battery voltages practical for the system by using the results of the motor testing.

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{p} \\
\dot{q} \\
\dot{r}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\phi \\
\theta \\
p \\
q \\
r
\end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\Delta \omega_1 \\
\Delta \omega_2 \\
\Delta \omega_3 \\
\Delta \omega_4
\end{bmatrix}
\]

Table 4: Eigenvalue and Eigenvector of A matrix of LTI state space model

| Eigenvalue | 0 | 0 | 0 | 0 | 0 |
| State | \phi | \theta | p | q | r |

American Institute of Aeronautics and Astronautics
Figure 10: Motors 1, 2, 3 and 4 Thrust and Torque Test Result (a) Voltage vs. Thrust, (b) Thrust vs. Voltage, (c) Voltage vs. Torque
D. Dynamic Model Simulation using LQR Controller

Using the proposed dynamic model, an LQR controller was designed to control the altitude, roll, and pitch angle of the quadcopter. The Q and R weighting matrices were selected as the following.

\[
Q = \begin{bmatrix}
1.4 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.4 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 10^{-3} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 10^{-3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9 \cdot 10^5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.9 \cdot 10^5
\end{bmatrix},
\]

\[
R = \begin{bmatrix}
1.33 & 0 & 0 & 0 \\
0 & 1.33 & 0 & 0 \\
0 & 0 & 1.33 & 0 \\
0 & 0 & 0 & 1.33
\end{bmatrix}
\]  

The LQR gain was computed in MATLAB and is shown below.

\[
k_{LQR} = \begin{bmatrix}
-1717 & -1708.2 & -672.3 & 634 & -0.3 & 133.4 & 132.9 \\
-1613.2 & 1616.2 & -613.1 & 581.1 & 0.5 & 125.5 & -125.9 \\
1631.4 & 1623.1 & 638.8 & 602.4 & -0.3 & -126.8 & -126.3 \\
1719.9 & -1723 & 653.6 & -619.5 & 0.5 & -133.8 & 134.2
\end{bmatrix}
\]  

To demonstrate the attitude correction of the controller, initial conditions for states and positions were selected at non-trim or non-hovering point. Table 5 shows the numerical values of initial conditions.

Table 5: Initial Condition of States and Position

<table>
<thead>
<tr>
<th></th>
<th>North[m]</th>
<th>(\phi[\text{deg}])</th>
<th>20</th>
<th>U[m/s]</th>
<th>0.1523</th>
<th>(P[\text{deg/s}])</th>
<th>-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>East[m]</td>
<td>0</td>
<td>(\theta[\text{deg}])</td>
<td>20</td>
<td>V[m/s]</td>
<td>0.035</td>
<td>(Q[\text{deg/s}])</td>
<td>-4</td>
</tr>
<tr>
<td>Down[m]</td>
<td>0.3045</td>
<td>(\psi[\text{deg}])</td>
<td>0</td>
<td>W[m/s]</td>
<td>0</td>
<td>(R[\text{deg/s}])</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 12 shows the result of the simulation of the dynamic model using the LQR controller. In the simulation, the controller corrects the roll and pitch angle to zero and the height displaces to the desired altitude at 0 meters. The LQR does not incorporate any yaw control on the system, so the steady state yawing rate is non-zero in the simulation.

![Figure 12: Result of Simulation using LQR Controller](image-url)
E. Parameter Estimation using SIDPAC

1. Thrust Coefficient Estimation

During the flight test, engine pairs are coupled to control body angular rates. A unique motor sign combination for decoupling can be obtained by combining pitch and roll rates with vertical velocity with Eq. 3, 7, and 8. From this, the thrust coefficients were determined using altitude, pitch rate, and roll rate output error matching to decouple the engine rotations and minimize overall parameter error. A caveat of Eq. 3 is that it requires XY-planar velocities to be near-zero to remove translational-angular velocity coupling. To account for this, the test flight was constrained to small pitch and roll Euler angles to prevent the excessive buildup of forward and side velocities. The altitude, roll angle, and pitch angle from the flight test are shown in Fig. 13.

Since the engine model uses the same four parameters as inputs for the three equations of motion, a least square estimation was not used. Instead an output error analysis is used to match propagated altitude, pitch, and roll rates, shown in Fig. 14. The parameters estimated were then used to simulate another portion of the flight test for model verification and graphically show model inaccuracies, shown in Fig. 15. Resulting engine thrust coefficients are shown below in the table below. The overall altitude shows good tracking in the output error matching and drift error in the simulation, however, the model only follows the average roll and pitch accelerations. The in-flight thrust coefficients were found to be similar to the test stand coefficients, shown in Table. 6.

![Flight Test States](image)

**Figure 13:** Flight Test States

<table>
<thead>
<tr>
<th>Engine</th>
<th>Test Stand</th>
<th>Flight Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2.86 \cdot 10^{-5}$</td>
<td>$1.84 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>2</td>
<td>$2.82 \cdot 10^{-5}$</td>
<td>$3.16 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>3</td>
<td>$2.64 \cdot 10^{-5}$</td>
<td>$2.12 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>4</td>
<td>$2.64 \cdot 10^{-5}$</td>
<td>$2.91 \cdot 10^{-5}$</td>
</tr>
</tbody>
</table>
2. Torque Coefficient Estimation

The results of the torque parameter estimation are shown in Fig. 16. Due to high coupling of motor rotations, individual torque coefficients could not be isolated given the flight data, as seen in Fig. 17. Instead, the average torque coefficient was determined using the summation of engine rotations squared with signs dictated by Eq. 9. Since there is only one equation, a least squares estimation was used for simplicity. The resulting average torque coefficient was found to be approximately four times higher than the value determined experimentally, as shown in Table. 7. Please note that this is not the result of neglecting to divide by the number of engines used while determining the average.
Figure 16: Torque Coefficient Simulated States

Figure 17: Engine Coupling during Yaw Rate Derivative Test

Table 7: Comparison of Average Engine Torque Coefficients

<table>
<thead>
<tr>
<th>Test Stand</th>
<th>Flight Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.992 · 10^{-6}</td>
<td>4.17 · 10^{-6}</td>
</tr>
</tbody>
</table>
3. Sources of Error

The dynamics of a quadcopter are heavily dependent on the position of the center of gravity, moments of inertia, and the rotational speeds of the motor. The quadcopter tested underwent design changes involving the landing gear before flight-testing. The new design slightly alters the quadcopter moments of inertia but is not enough to explain the lack of tracking in the angular velocities. Another error source is that the center of gravity location was found to be slightly below the intersection of the quadcopter arms. This would potentially create a pendulum effect that would need to be incorporated into the dynamic model. However, when accounting for the pendulum dynamics in the modeling and simulation, testing resulted in no noticeably differences than in the proposed point mass model. A third source of error comes from the lack of explicit motor RPM feedback. This could lead to errors concerning parameter estimation in post flight analysis. The motor rotations are not measured directly but are instead derived from the pulse width modulations of the electric speed controller and modeled as an instantaneous change. While the relation between pulse width modulation and motor rotational speed is determined before flight through the ESC calibration, the actual motor rotations depend largely on variables such as current supplied to the motor and the resistance torque on the motor.

IV. Conclusion

A quadcopter dynamic model was developed that utilized an adaptive thrust and torque model. The moments of inertia were estimated by conducting a bifilar pendulum test. The resulting standard deviations were very low and indicates the reliability of the bifilar pendulum test. To obtain the thrust and torque data, a custom built test bed was used. This test bed could conduct the experiment to record thrust, torque, voltage, RPM and the current. With the resulting data, the thrust and torque coefficients were estimated for the measured voltage level. The trim thrust PWM command value was evaluated with a developed algorithm corresponding to the battery voltage. The parameters determined from the inertia and motor testing were used to develop the state space model, which was used to develop an LQR controller for the system. The controller was used in a simulation to demonstrate the response of the system to changes in desired altitude and attitude. The quadcopter was flight tested and parameter estimation was used to compare with the results from motor testing. Similar thrust coefficients were determined, though discrepancies were found in the torque coefficients.

Acknowledgments

The authors would like to acknowledge colleagues (Katie Constant-Coup, Riley Sprunger, Shiriniwas Kolpule, and Daksh Shukla) for supporting this work. The authors would also like to express their appreciation for the immense support and guidance from Dr. Shawn Keshmiri and Dr. Hee Chul Yun.

References