Quadrotor Mathematics for Simpletons

Cody Lewis
Anzhelka Project
srlm@anzhelka.com

March 8, 2012

And when he [Herod] had apprehended him [Peter], he put him in prison, and delivered him to four quaternions of soldiers to keep him; intending after Easter to bring him forth to the people.
–Acts 12:14, King James Bible, Cambridge Edition

Abstract

This paper describes in detail the dynamics of a quadrotor platform. Most current research papers simply explain the equations without explaining derivations or reasoning behind the numbers. This paper will explain and derive the equations for a quadrotor aircraft based an the system found in [1].

1 Introduction

A quadrotor is a simple aerial vehicle much like a helicopter, but instead of a single main rotor a quadrotor has four rotors. This allows for a much simpler mechanical system, since each rotor can be directly mounted to it's motor without any linkages to change the pitch of a rotor.

Quadrotor dynamics are very complex. It's difficult because you have four motors, all pointed in the same direction, and that's it. But yet the vehicle can move on three axes in translational motion, and can rotate about another three axes. A quadrotor must be able to convert the four motor inputs into a these six different (and often conflicting) motions.

The simplest case is hovering in a single location, and will be dealt with in this paper. Initially it is assumed that the quadrotor will have negligable translational motion, although eventually the translational equations may be added to this paper.

The quadrotor used in this paper is assumed to be fairly simple. The platform must be able to measure the following components:

* rotation (direction and velocity)

* translational (velocity and accleration)

* motor rotational speed

* air density

Note that if air density is assumed to be constant than performance may suffer at different altitudes or weather. In addition, the following constants must be emperically measured for each quadrotor:

* mass (kg)

* moment of inertia tensor ($kg \cdot m^2$)

* offset of each rotor from the center of mass (m)

* thrust coefficient (see –CITE)

* torque coefficient (see –CITE)

* rotor diameter (m)

The system is complicated by the fact that a quadrotor will need to be able to balance itself in flight. If a gust of wind blows from the side and rolls the quadrotor, it will no longer be a stable hover. The vehicle will need to increase the speed of some motors and decrease the speed of others so that the vehicle is rotated back into a vertical hovering state.

There is still a problem: each motor is producing torque on the body as well as thrust. This is manifested by the motor "attempting" to rotate the body due to the spinning mass of the motor and propeller, and the air resistance the propeller encounters. On a quadrotor frame we have two motors spinning clockwise and two spinning counter clockwise, and the torques cancel each other out. But if a motor speed in increased to account for a tilt, then it's torque on the frame will increase as well, causing the whole body to rotate.

And so now, when a gust of wind blows on the frame, we have to somehow correct the tilt without allowing the torque to yaw the frame. At other times we may want to yaw without tilting, and sometimes we want to do both simultaneously. But there is so much more beyond just the basic thrust, tilt, and yaw: we have to account for many factors such as free stream velocity (or air, in layman's terms) through the propellers, blade flapping, sensor failure, and the many small equations that support the main equations.

2 Symbols Used

2.1 Variables

Variables with the subscript i denote association with motor i.

$\boldsymbol{\omega}_b$	angular velocity in the body frame (rad/s)
$\mathbf{q}$	attitude quaternion
$\tilde{\mathbf{q}}$	error quaternion
$\mathbf{q}_d$	desired quaternion
$\boldsymbol{V}_b$	velocity in the body frame (m/s)
$\boldsymbol{R}_e$	position in the inertial frame (m)
F_z	total force of the rotors on the z axis (N)
$M_x - M_z$	total rotor moments along each axis ($N \cdot m$)
$F_1 - F_4$	thrust force of each rotor (N)
Q	rotor torque ($N \cdot m$)
T	rotor thrust force (N)
n_i	rotation frequency of the rotors (Hz)
Ω_i	rotation speed of the rotors (rad/s)
Ω_{di}	desired rotation speed of the rotors (rad/s)
Ω_{imeas}	measured rotation speed of the rotors (rad/s)
$\hat{\Omega}_i$	expected speed (rad/s)
u_i	motor command (???)
a_i	motor gain (no units)
ψ	yaw angle (???)
γ_H	tilt direction in the x-y plane (???)
α_H	tilt amount around γ_H (???)
$\boldsymbol{q}_V$	yaw error
$\boldsymbol{q}_H$	tilt error
$\begin{bmatrix} r_x \\ r_y \end{bmatrix}$	tilt axis in the $x - y$ plane (???)
β_H	direction of the tilt axis in the $x - y$ plane (???)
K_{tilt}	amount of tilt in the system? (from altitude controller)

2.2 Constants

$\boldsymbol{I}_{nb}$	moment of inertia tensor ($kg \cdot m^2$)
D	rotor diameter (m)
d	offset of each rotor from the center of mass (m)
ρ	air density (???)
K_t	thrust coefficient (???)
K_q	torque coefficient (???)
K_g	the speed adaptable gain for a_i (???)
$f_{mot}(\cdot)$	motor response function (???)
τ	time constant, step (???)
K_{PH}	tilt moment proportional constant
K_{DH}	tilt moment derivative constant
K_{Pz}	yaw moment proportional constant
K_{Dz}	yaw moment derivative constant
F_{PIDz}	proportional altitude controller (N ?)
m	quadrotor mass (kg ?)

3 Introduction to Quaternions

This paper represents orientation using quaternions. There are other methods of representing rotation, but a quaternion has several advantages that lend it well to use on the quadrotor:

* Quaternions avoid gimbal lock (where two rotation axes align on each other during the course of a rotation, producing unnatural results)

* Quaternions have smaller memory requirements (4 floats) than other representations

* Quaternions have much more efficient mathematical operations, and don't require orthonormalization (making the axes orthogonal after an imprecise computation) like a rotation matrix does.

* Quaternions allow for smooth interpolation between two orientations.

* Orientation updates (updating a refernce orientation with ω, for example) do not require trigonometric functions.

A quaternion is a way of representing a rotation. This is similar to Euler angles, but has the advantage of a more concise notation (and math), and not having singularities like Euler angles have. Quaternions are difficult to visualize from the numbers, but are easy enough to understand and use.

A quaternion is defined as follows:

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2) \cdot \boldsymbol{r} \end{bmatrix}$$

This means that the rotation is around the axis $\boldsymbol{r}$ and by amount α. Note that in the second representation the second element is q_1, q_2, and q_3 (it's three because of the $\cdot\boldsymbol{r}$, which is a three element vector). To add some terminology, the first term q_0 is called the scalar part, while q_1 through q_3 is called the vector part.

A quaternion can be thought of a number, just like any other. So, we can do several operations such as add, multiply, invert (divide), take the norm, and of course test for equality. We'll be focusing on rotation quaternions for the rest of this paper. A rotation quaternion is the specific form of quaternion used to represent rotation, and must have a norm equal to one. We define the norm of quaternion as

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$$

The norm can be thought of as the "length" of a quaternion (or vector, etc). A $\|q\|$ value of 1 is called a unit quaternion. In quaternion literature there seems to be a discrepency: some authors define the quaternion norm as above, with a square root (—Citations—), while others omit the square root ([3] —CITE—). In this paper we will use the norm with a square root. This is called the L2-norm, or more commonly the Euclidean norm. I couldn't find anything on a norm that squared it's components but had no root, so we'll ignore it (for more information on norms, see [2]).

We also define the conjugate and inverse of a rotation quaternion as

$$q' = (q_0, -q_1, -q_2, -q_3) = q^{-1}$$

Note this equality is only true for unit quaternions (rotation quaternions), and does not hold for other quaternions.

The only binary operation that we will be using for rotation quaternions is multiplication, denoted by $\otimes$. Note that quaternion multiplication is not commutative. Multiplying $q_1 \otimes q_2 \neq q_2 \otimes q_1$. Multiplying rotation quaternions produces a third rotation quaternion that is the result of rotating each quaternion in reverse order starting with the right hand side quaternion. To rotate a standard vector from one coordinate system to another simply place it into the vector part of the quaternion and set the scalar part to zero. Then, for transforming the vector r_e to the vector r_b:

$$\begin{bmatrix} 0 \\ r_b \end{bmatrix} = \boldsymbol{q}^* \otimes \begin{bmatrix} 0 \\ r_e \end{bmatrix} \otimes \boldsymbol{q} \tag{3.1}$$

$$\begin{bmatrix} 0 \\ r_e \end{bmatrix} = \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ r_b \end{bmatrix} \otimes \boldsymbol{q}^* \tag{3.2}$$

$$\tag{3.3}$$

For the rest of this paper, if a vector is rotated by a quaternion it will be written as a vector but assumed to be in the vector part of a quaternion. For example,

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \omega_b, \qquad \text{where} \quad \omega_b = \begin{bmatrix} 0 \\ \omega_b \end{bmatrix} \tag{3.4}$$

Does the above actually make sense?... Need to clarify

The product of two quaternions is another quaternion and is defined as:

$$\begin{aligned} p \otimes q = (&p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3, \\ &p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2, \\ &p_0q_2 + p_2q_0 + p_3q_1 - p_1q_3, \\ &p_0q_3 + p_3q_0 + p_1q_2 - p_2q_1) \end{aligned} \tag{3.5}$$

Question: Order of the component multiplication doesn't matter, right? Also, double check the correct multiplication...

One thing to note is that a quaternion multiplication has 12 scalar additions and 16 scalar multiplies. Since the components are floating point, each quaternion multiplication consumes 28 floating point operations.

—-A bunch more stuff to define——
—-Need to define the fact that two quaternions can represent the same space
—-Need to define quaternion derivatives —-
—-What is q^*?—-

4 Basic Equations

4.1 Quadrotor Model

$$\dot{\boldsymbol{V}}_b = \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} + \boldsymbol{q}^* \otimes \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \otimes \boldsymbol{q} - \boldsymbol{\omega}_b \times \boldsymbol{V}_b \tag{4.1}$$

$$\dot{\boldsymbol{\omega}}_b = \boldsymbol{I}_{nb}^{-1}\left(\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - \boldsymbol{\omega_b} \times (\boldsymbol{I}_{nb}\boldsymbol{\omega}_b)\right) \tag{4.2}$$

$$\dot{\boldsymbol{R}}_e = \boldsymbol{q} \otimes \boldsymbol{V}_b \otimes \boldsymbol{q}^* \tag{4.3}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega}_b \tag{4.4}$$

Question: Is the above equation for quaternion derivatives?...

Next, in equation (4.5) we have a standard conversion from rotational frequency (in RPM – CHECK!) to angular frequency (in radians/second – CHECK!).

$$\Omega_i = 2\pi n_i \tag{4.5}$$

Each motor generates a thrust in the direction of the motor shaft axis, and a torque around the motor shaft axis. Equations (4.6) and (4.7) are derived from –WHAT???

$$K_T = \frac{T}{\rho n^2 D^4} \tag{4.6}$$

$$K_Q = \frac{Q}{\rho n^2 D^5} \tag{4.7}$$

These two equations are used to define K_T and K_Q, which are determined experimentally. A motor test stand must be able to measure the components on the right hand side and generate the constants for a particular motor and propeller combination. For use onboard the quadrotor, equation (4.6) will be used with equation (4.5) to connect thrust to rotational speed in equation (4.8). By this combination we will be able to determine what rotational speed we need to run the motors at to achieve a specified thrust. From equations (4.6) and (4.5) we rewrite and combine:

$$\begin{aligned}
n^2 &= \frac{T}{\rho K_T D^4} \\
n &= \frac{1}{D^2}\sqrt{\frac{T}{\rho K_T}} \\
\Omega_{di} &= 2\pi n \\
\Omega_{di} &= \frac{2\pi}{D^2}\sqrt{\frac{T}{\rho K_T}} \qquad (4.8)
\end{aligned}$$

Now, with equation (4.8) we have an equation that will tie the thrust required to how fast the rotors must spin. This can later be fed into a PI controller for each motor, described in —–SECTION CITATION—-.

$$\begin{bmatrix} F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ -d & 0 & d & 0 \\ c & -c & c & -c \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{4.9}$$

where

$$c = \frac{Q}{T} \tag{4.10}$$

4.2 Yaw and Tilt Errors

$$\boldsymbol{q}_d = \tilde{\boldsymbol{q}} \otimes \boldsymbol{q} \tag{4.11}$$

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_d \otimes \boldsymbol{q}^* \tag{4.12}$$

$$\tilde{\boldsymbol{q}} = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2) \cdot \boldsymbol{r}_e \end{bmatrix} \tag{4.13}$$

Question: should α in the equations above and below have the subscript α_h like later in this section?

$$\boldsymbol{r}_b = q^* \otimes \boldsymbol{r}_e \otimes \boldsymbol{q} \tag{4.14}$$

$$\tilde{\boldsymbol{q}}_b = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2) \cdot \boldsymbol{r}_b \end{bmatrix} \tag{4.15}$$

Question: are β_H and Γ_H perpendicular or somehow constant relative to one another?

$$\tilde{\boldsymbol{q}}_b = \boldsymbol{q}_H \otimes \boldsymbol{q}_V \tag{4.16}$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\alpha_H/2) \\ \sin(\alpha_H/2) \cdot r_x \\ \sin(\alpha_H/2) \cdot r_y \\ \sin(\alpha_H/2) \cdot 0 \end{bmatrix} \otimes \begin{bmatrix} \cos(\psi/2) \\ \sin(\psi/2) \cdot 0 \\ \sin(\psi/2) \cdot 0 \\ \sin(\psi/2) \cdot 1 \end{bmatrix} \tag{4.17}$$

$$\alpha_H = \arccos\left[1 - 2\left(q_1^2 + q_2^2\right)\right] \qquad 0 \leq \alpha_H < \pi \tag{4.18}$$

$$\psi = 2\arctan_2(q_3, q_0) \qquad -\pi \leq \psi < \pi \tag{4.19}$$

$$r_x = \frac{\cos(\psi/2)\, q_1 - \sin(\psi/2)\, q_2}{\sin(\alpha_H/2)} \tag{4.20}$$

$$r_y = \frac{\sin(\psi/2)\, q_1 - \cos(\psi/2)\, q_2}{\sin(\alpha_H/2)} \tag{4.21}$$

$$\beta_H = \arctan_2(r_y, r_x) \tag{4.22}$$

$$\psi_H = \arctan_2(r_x, r_y) \tag{4.23}$$

4.3 Motor control

This subsection describes the equations used to set the rotational speed of the motors. Note that this differs from [1]: in that paper the authors used an empirically measured function (f_{mot}) to determine the rotational speed based on the voltage of the motor and the command given. This required a dynamically adjustable constant a_i to calibrate each motor for small differences in manufacturing. In this paper it is assumed that actual motor rotational speed is able to be measured. By measuring the rotational speed of the motors we are able to ignore the manufacturing differences between motors. If a motor is not spinning at the correct speed the the PI control loop described next will be able to correct until the desired rotation is achieved.

The following two equations are explicitly defined a discrete time t. For clarity, the subscript i denoting a specific motor is not written but is implied:

$$\Omega\left(t\right)_{Ierror} = \Omega\left(t-1\right)_{Ierror} + \left(\Omega_d - \Omega_i\right) \tag{4.24}$$

$$u_i = K_P\left(\Omega_d - \Omega_i\right) + K_I \Omega\left(t\right)_{Ierror} \tag{4.25}$$

4.4 Attitude stabilization

$$M_x = K_{PH}\alpha_H \cos\beta_H - K_{DH}\omega_x \tag{4.26}$$
$$M_y = K_{PH}\alpha_H \sin\beta_H - K_{DH}\omega_y \tag{4.27}$$
$$M_z = K_{Pz}\psi - K_{Dz}\omega_z \tag{4.28}$$
$$\tag{4.29}$$

4.5 Altitude Control

$$F_z = \frac{mg + F_{PIDz}}{K_{tilt}} \tag{4.30}$$

$$K_{tilt} = \begin{bmatrix}0 & 0 & 1\end{bmatrix} \cdot \left(q \otimes \begin{bmatrix}0\\0\\-1\end{bmatrix} \otimes q^*\right) \tag{4.31}$$

5 Basic Equations from [1]

The follow equations are exactly as written in [1]. They are here for reference only.

5.1 Quadrotor Model

$$\dot{\boldsymbol{V}}_b = \frac{1}{m}\begin{bmatrix}0\\0\\F_z\end{bmatrix} + \boldsymbol{q}^* \otimes \begin{bmatrix}0\\0\\g\end{bmatrix} \otimes \boldsymbol{q} - \boldsymbol{\omega}_b \times \boldsymbol{V}_b \tag{5.1}$$

$$\dot{\boldsymbol{\omega}}_b = \boldsymbol{I}_{nb}^{-1}\left(\begin{bmatrix}M_x\\M_y\\M_z\end{bmatrix} - \boldsymbol{\omega_b} \times \left(\boldsymbol{I}_{nb}\boldsymbol{\omega}_b\right)\right) \tag{5.2}$$

$$\dot{\boldsymbol{R}}_e = \boldsymbol{q} \otimes \boldsymbol{V}_b \otimes \boldsymbol{q}^* \tag{5.3}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega}_b \tag{5.4}$$

$$K_T = \frac{T}{\rho n^2 D^4} \tag{5.5}$$

$$K_Q = \frac{Q}{\rho n^2 D^5} \tag{5.6}$$

$$\Omega_i = 2\pi n_i \tag{5.7}$$

$$\tau\dot{\Omega}_i + \Omega_i = a_i f_{mot}\left(u_i, V_{batt}\right) \tag{5.8}$$

$$\begin{bmatrix} F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ -d & 0 & d & 0 \\ c & -c & c & -c \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{5.9}$$

where

$$c = \frac{Q}{T} \tag{5.10}$$

$$\Omega_{di} = \frac{2\pi}{D^2}\sqrt{\frac{-F_{di}}{K_T \rho}} \tag{5.11}$$

$$u_i = f_{mot}^{-1}\left(\frac{1}{a_i}\Omega_{di}, V_{batt}\right) \tag{5.12}$$

5.2 Motor Gain Estimation

$$\dot{a}_i = K_g\left(\Omega_{imeas} - \hat{\Omega}_i\right) \tag{5.13}$$

5.3 Yaw and Tilt Errors

$$\boldsymbol{q}_d = \tilde{\boldsymbol{q}} \otimes \boldsymbol{q} \tag{5.14}$$

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_d \otimes \boldsymbol{q}^* \tag{5.15}$$

$$\tilde{\boldsymbol{q}} = \begin{bmatrix} \cos\left(\alpha/2\right) \\ \sin\left(\alpha/2\right) \cdot \boldsymbol{r}_e \end{bmatrix} \tag{5.16}$$

Question: should α in the equations above and below have the subscript α_h like later in this section?

$$\boldsymbol{r}_b = q^* \otimes \boldsymbol{r}_e \otimes \boldsymbol{q} \tag{5.17}$$

$$\tilde{\boldsymbol{q}}_b = \begin{bmatrix} \cos\left(\alpha/2\right) \\ \sin\left(\alpha/2\right) \cdot \boldsymbol{r}_b \end{bmatrix} \tag{5.18}$$

Question: are β_H and Γ_H perpendicular or somehow constant relative to one another?

$$\tilde{\boldsymbol{q}}_b = \boldsymbol{q}_H \otimes \boldsymbol{q}_V \tag{5.19}$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\alpha_H/2) \\ \sin(\alpha_H/2) \cdot r_x \\ \sin(\alpha_H/2) \cdot r_y \\ \sin(\alpha_H/2) \cdot 0 \end{bmatrix} \otimes \begin{bmatrix} \cos(\psi/2) \\ \sin(\psi/2) \cdot 0 \\ \sin(\psi/2) \cdot 0 \\ \sin(\psi/2) \cdot 1 \end{bmatrix} \tag{5.20}$$

$$\alpha_H = \arccos\left[1 - 2\left(q_1^2 + q_2^2\right)\right] \qquad 0 \leq \alpha_H < \pi \tag{5.21}$$

$$\psi = 2\arctan_2(q_3, q_0) \qquad -\pi \leq \psi < \pi \tag{5.22}$$

$$r_x = \frac{\cos(\psi/2)\, q_1 - \sin(\psi/2)\, q_2}{\sin(\alpha_H/2)} \tag{5.23}$$

$$r_y = \frac{\sin(\psi/2)\, q_1 - \cos(\psi/2)\, q_2}{\sin(\alpha_H/2)} \tag{5.24}$$

$$\beta_H = \arctan_2(r_y, r_x) \tag{5.25}$$

$$\psi_H = \arctan_2(r_x, r_y) \tag{5.26}$$

5.4 Attitude stabilization

$$M_x = K_{PH}\alpha_H \cos\beta_H - K_{DH}\omega_x \tag{5.27}$$
$$M_y = K_{PH}\alpha_H \sin\beta_H - K_{DH}\omega_y \tag{5.28}$$
$$M_z = K_{Pz}\psi - K_{Dz}\omega_z \tag{5.29}$$
$$\tag{5.30}$$

5.5 Altitude Control

$$F_z = \frac{mg + F_{PIDz}}{K_{tilt}} \tag{5.31}$$

$$K_{tilt} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \left(q \otimes \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \otimes q^* \right) \tag{5.32}$$

$$F_{PIDz} = ??? \tag{5.33}$$

References

[1] E. Stingu, F. Lewis, "Design and Implementation of a Structured Flight Controller for a 6DoF Quadrotor Using Quaternions," Mediterranean Conference on Control & Automation, pp. 1233-1238, June 2009.

[2] M. Malek, "Vector and Matrix Norms", California State University, East Bay, May 2009.

[3] K. Shoemake, "Quaternions", Department of Computer and Information Science, University of Pennsylvania.