`quadcopter` package

dinay-kingkiller

I. Introduction

The `quadcopter` package is filled with obtuse and often confusing code derived from various physics equations and mathematical formulas. This paper hopes to bridge the gap between quadcopter theory and the `src` files included. This paper can also be used as a reference for later expansion of this package.

II. The Model Node

While an actual quadcopter would be helpful, a simulation is the next best thing. The model node simulates the physics of a quadcopter. The physics in this section also provide insight on how motor input and sensor output behave.

A. Rotational Kinematics

Quaternions are integral in understanding the implementation of the model node. Derivations of quaternion operations can be found elsewhere, including on my GitHub. This paper will then focus on the implementation in the model node. There are multiple definitions for quaternion multiplication. The model node uses the same definition as the `tf2` ROS package.

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ w_1 \end{pmatrix} \otimes \begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ w_2 \end{pmatrix} = \begin{pmatrix} w_1x_2 + x_1w_2 + y_1z_2 - z_1y_2 \\ w_1y_2 - x_1z_2 + y_1w_2 + z_1x_2 \\ w_1z_2 + x_1y_2 - y_1x_2 + z_1w_2 \\ w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2 \end{pmatrix} \tag{1}$$

The unit quaternion q describes the orientation of the robot by describing a transform between two frames

$$\begin{pmatrix} \mathbf{v}^B \\ 0 \end{pmatrix} = \mathrm{q} \otimes \begin{pmatrix} \mathbf{v}^N \\ 0 \end{pmatrix} \mathrm{q}^* \tag{2}$$

where $\mathbf{v}^N$ and $\mathbf{v}^B$ are the same vector described in an inertial frame N and the robot body frame B respectively. A rotation matrix is another description of rotations:

$$\mathbf{v}^B = \mathrm{R}\left(\mathbf{v}^N\right) \tag{3}$$

Importantly for the model node, rotation matrices collect the dot product of basis vectors: $\mathrm{R}_{ij} = \hat{\mathbf{b}}_i \cdot \hat{\mathbf{n}}_j$. In terms of quaternion (x, y, z, w)

$$\mathrm{R} = \begin{pmatrix} x^2-y^2-z^2+w^2 & 2xy-2zw & 2xz+2yw \\ 2xy+2zw & -x^2+y^2-z^2+w^2 & -2xw+2yz \\ 2xz-2wy & 2xw+2yz & -x^2-y^2+z^2+w^2 \end{pmatrix} \tag{4}$$

That rate of change of the quaternion can be calculated from the angular velocity

$$\dot{\mathrm{q}} = \frac{1}{2}\begin{pmatrix} \boldsymbol{\omega} \\ 0 \end{pmatrix} \otimes \mathrm{q} \tag{5}$$

where $\boldsymbol{\omega}$ is the angular velocity of the robot body B in the inertial frame N described in the body basis:

$$\boldsymbol{\omega} = \omega_x\hat{\mathbf{b}}_x + \omega_y\hat{\mathbf{b}}_y + \omega_z\hat{\mathbf{b}}_z \tag{6}$$

B. Linear Equations of Motion

The linear equations of motion are pretty apparent upon starting our analysis. First, the motion of the quadcopter center of mass c from its origin point o is defined simply:

$$\mathbf{p}^N_{c/o} = p_x\hat{\mathbf{n}}_x + p_y\hat{\mathbf{n}}_y + p_z\hat{\mathbf{n}}_z \tag{7}$$

$$\mathbf{v}^N_c = \dot{p}_x\hat{\mathbf{n}}_x + \dot{p}_y\hat{\mathbf{n}}_y + \dot{p}_z\hat{\mathbf{n}}_z \tag{8}$$

$$\mathbf{a}^N_c = \ddot{p}_x\hat{\mathbf{n}}_x + \ddot{p}_y\hat{\mathbf{n}}_y + \ddot{p}_z\hat{\mathbf{n}}_z \tag{9}$$

For our model we assume each motor puts out a force proportional to its square. If we let m be the total mass of the robot and k be the proportionality constant:

$$m\mathbf{a}^N_c = k\left(\omega_f^2 + \omega_l^2 + \omega_b^2 + \omega_r^2\right)\hat{\mathbf{b}}_z - mg\hat{\mathbf{n}}_z \tag{10}$$

Where f, l, b, and r stand for the front, right, back and left motors respectively.

C. Rotational Equations of Motion

Supposing that most of the mass of the quadcopter is in the motors, the inertia dyadics (around the center of mass c) of the four motors are

$$\mathbf{I}_{f/c} = \frac{mr^2}{4}\hat{\mathbf{b}}_y\hat{\mathbf{b}}_y + \frac{mr^2}{4}\hat{\mathbf{b}}_z\hat{\mathbf{b}}_z \tag{11}$$

$$\mathbf{I}_{l/c} = \frac{mr^2}{4}\hat{\mathbf{b}}_x\hat{\mathbf{b}}_x + \frac{mr^2}{4}\hat{\mathbf{b}}_z\hat{\mathbf{b}}_z \tag{12}$$

$$\mathbf{I}_{b/c} = \frac{mr^2}{4}\hat{\mathbf{b}}_y\hat{\mathbf{b}}_y + \frac{mr^2}{4}\hat{\mathbf{b}}_z\hat{\mathbf{b}}_z \tag{13}$$

$$\mathbf{I}_{r/c} = \frac{mr^2}{4}\hat{\mathbf{b}}_x\hat{\mathbf{b}}_x + \frac{mr^2}{4}\hat{\mathbf{b}}_z\hat{\mathbf{b}}_z \tag{14}$$

where the mass of each motor is $m/4$ and r is the radius of the robot. The total inertial dyadic of the robot is then

$$\mathbf{I}_{B/c} = \frac{mr^2}{2}\left(\hat{\mathbf{b}}_x\hat{\mathbf{b}}_x + \hat{\mathbf{b}}_y\hat{\mathbf{b}}_y + 2\hat{\mathbf{b}}_z\hat{\mathbf{b}}_z\right) \tag{15}$$

The angular velocity was given in equation 6.

$$\boldsymbol{\omega}^N_B = \omega_x\hat{\mathbf{b}}_x + \omega_y\hat{\mathbf{b}}_y + \omega_c\hat{\mathbf{b}}_z$$

The angular acceleration comes out simple because of our choice of reference frame

$$\begin{aligned} \boldsymbol{\alpha}^N_B &= \frac{{}^B d}{dt}\boldsymbol{\omega}^N_B + \boldsymbol{\omega}^N_B \times \boldsymbol{\omega}^N_B \\ &= \dot{\omega}_x\hat{\mathbf{b}}_x + \dot{\omega}_y\hat{\mathbf{b}}_y + \dot{\omega}_z\hat{\mathbf{b}}_z \end{aligned} \tag{16}$$

And the angular momentum around the center of mass c is pretty nice too

$$\begin{aligned} \mathbf{H}^N_{B/c} &= \mathbf{I}_{B/c} \cdot \omega^N_B \\ &= \frac{mr^2}{2}\omega_x\hat{\mathbf{b}}_x + \frac{mr^2}{2}\omega_y\hat{\mathbf{b}}_y + mr^2\omega_z\hat{\mathbf{b}}_z \end{aligned} \tag{17}$$

And the derivative is

$$\begin{aligned}\frac{{}^N d}{dt}\mathbf{H}^N_{B/c} &= \mathbf{I}\cdot\boldsymbol{\alpha}^N_B + \boldsymbol{\omega}^N_B \times \mathbf{H}^N_{B/c} \\ &= \frac{mr^2}{2}\dot{\omega}_x\hat{\mathbf{b}}_x + \frac{mr^2}{2}\dot{\omega}_y\hat{\mathbf{b}}_y + mr^2\dot{\omega}_y\hat{\mathbf{b}}_z \\ &+ \frac{mr^2}{2}\omega_y\omega_z\hat{\mathbf{b}}_x - \frac{mr^2}{2}\omega_x\omega_z\hat{\mathbf{b}}_y \end{aligned} \tag{18}$$

Quadcopters move laterally by rolling or pitching (rotation around the x and y axes respectively) then increasing thrust to go "up". A simple rotation around a single axis is generated by increasing the speed on one propeller and decreasing its opposite. The moment of these forces provides a torque to the robot. All forces point up, so

$$\begin{aligned}\mathbf{M}_{B/c} &= r\hat{\mathbf{b}}_x \times k\omega_f^2\hat{\mathbf{b}}_z - r\hat{\mathbf{b}}_x \times k\omega_b^2\hat{\mathbf{b}}_z \\ &+ r\hat{\mathbf{b}}_y \times k\omega_l^2\hat{\mathbf{b}}_z - r\hat{\mathbf{b}}_y \times k\omega_r^2\hat{\mathbf{b}}_z & (19) \\ &= kr\left(\omega_l^2 - \omega_r^2\right)\hat{\mathbf{b}}_x + kr\left(\omega_b^2 - \omega_f^2\right)\hat{\mathbf{b}}_y & (20)\end{aligned}$$

For sanity sake, let's show that the other force acting on the motors, gravity, does not affect the rotation

$$\begin{aligned}-r\hat{\mathbf{b}}_x \times \frac{mg}{4}\hat{\mathbf{n}}_z + r\hat{\mathbf{b}}_x \times \frac{mg}{4}\hat{\mathbf{n}}_z = \mathbf{0} \\ -r\hat{\mathbf{b}}_y \times \frac{mg}{4}\hat{\mathbf{n}}_z + r\hat{\mathbf{b}}_y \times \frac{mg}{4}\hat{\mathbf{n}}_z = \mathbf{0}\end{aligned}$$

As long as the weight of the quadcopter is balanced along the axes, gravity should not affect the rotation. Adjacent quadcopter propellers have reverse pitch, so that together they can provide a yaw torque (around the z axis) in either direction; without it quadcopters would have to choose to fall or rotate. Choosing the front and rear motors to provide positive torque

$$\boldsymbol{\tau} = b\left(\omega_f^2 - \omega_r^2 + \omega_b^2 - \omega_l^2\right)\hat{\mathbf{b}}_z \tag{21}$$

The last couple equations combine to form the rotational equations of motion

$$\frac{{}^N d}{dt}\mathbf{H}^N_{B/c} = \mathbf{M}_{B/c} + \boldsymbol{\tau} \tag{22}$$

D. Numerical Integration

The first differential equations are the easiest differential equations. Define $v_i = \dot{r}_i$. Then $\dot{v}_i = \ddot{p}_i$ and

$$\begin{aligned}\dot{p}_x &= v_x & (23) \\ \dot{p}_y &= v_y & (24) \\ \dot{p}_z &= v_z & (25)\end{aligned}$$

Recall the linear equation of motion in equation 10.

$$m\mathbf{a}^N_c = k\left(\omega_f^2 + \omega_l^2 + \omega_b^2 + \omega_r^2\right)\hat{\mathbf{b}}_z - mg\hat{\mathbf{n}}_z$$

By taking the dot product with the inertial basis N via equation 4, we get our second set of equations of motion

$$\begin{aligned}\dot{v}_x &= 2Tq_xq_z - 2Tq_wq_y & (26) \\ \dot{v}_y &= 2Tq_xq_w + 2q_yq_z & (27) \\ \dot{v}_z &= -Tq_x^2 - Tq_y^2 + Tq_z^2 + Tq_w^2 & (28)\end{aligned}$$

where T is the specific motor thrust.

$$T = \frac{k}{m}\left(\omega_f^2 + \omega_l^2 + \omega_b^2 + \omega_r^2\right) \tag{29}$$

The rotational equations come out nicely since we chose to derive our equations in the robot frame

$$\begin{aligned}\dot{\omega}_x &= \omega_y\omega_z + \frac{2k}{mr}\left(\omega_l^2 - \omega_r^2\right) & (30) \\ \dot{\omega}_y &= \omega_x\omega_z + \frac{2k}{mr}\left(\omega_b^2 - \omega_f^2\right) & (31) \\ \dot{\omega}_z &= \frac{b}{mr^2}\left(\omega_f^2 - \omega_r^2 + \omega_b^2 - \omega_l^2\right) & (32)\end{aligned}$$

Expanding equation 5 gives the final four equations

$$\begin{aligned}\dot{q}_x &= -\omega_zq_y + \omega_yq_z + \omega_xq_w & (33) \\ \dot{q}_y &= \omega_zq_x - \omega_xq_z + \omega_yq_w & (34) \\ \dot{q}_z &= -\omega_yq_x + \omega_xq_y + \omega_zq_w & (35) \\ \dot{q}_w &= -\omega_xq_x - \omega_yq_y - \omega_zq_z & (36)\end{aligned}$$

This most simple integration method (Euler's Method) is fast to implement, but More accurate methods like Runge-Kutta or Crouch-Grossman could improve the accuracy. However they shouldn't be unneeded. Those algorithms can be more computationally intensive, and it's just a model.