Cerebri Multirotor

James Goppert

January 22, 2023

Contents

Notation

We will use a notation where $\hat{x}_e^b$ is the x unit vector of frame b expressed in component of frame e, $\vec{v}_b$ is a vector v expressed in components of frame b, and $\vec{\omega}_b^{eb}$ is the angular velocity of frame b with respect to frame, expressed in components of frame b. If not coordinate frame is indicated, it indicates the general vector that can be expressed in any frame. The rotation matrix C_b^e operates on vectors such that $\vec{v}_b = C_b^e \vec{v}_e$. This convention let's us be very specific about the coordinate frame each vector is expressed in when necessary.

1 Trajectory Generation

1.1 Problem Statement

To simplify planning, we will leverage the differential flatness of the multirotor to solve for the states and inputs as a fucntion of the flat outputs and their derivatives. We will closely follow Mellinger 2012[1].

Flat Outputs

- $\vec{p}_e$: the position of the multirotor expressed in the e frame
- ψ : the desired yaw angle

Flat Output Derivatives

- $\vec{v}_e$: the velocity of the multirotor expressed in the e frame
- $\vec{a}_e$: the acceleration of the multirotor expressed in the e frame
- $\vec{j}_e$: the jerk of the multirotor expressed in the e frame
- $\vec{s}_e$: the snap of the multirotor expressed in the e frame
- $\dot{\psi}$, $\ddot{\psi}$, ... : derivatives of the yaw angle
- higher derivatives if necessary

States

- $\vec{p}_e$: the position of the multirotor expressed in the e frame
- $\vec{v}_b$: the velocity of the multirotor wrt the e frame expressed in the b frame
- C_e^b : the direction cosine matrix, element of $SO(3)$, of the states wrt to the b frame, can be parameterized with Euler angles, quaternions, etc.
- $\vec{\omega}_b^{eb}$: the angular velocity of body frame wrt the e frame expressed in the b frame

Inputs

- T : The thrust
- $\vec{M}_b$: the control moment of the quadrotor due to the motors expressed in the b (body) frame

Frames

Note we deviate from Mellginer's convention and use the North-East-Down instead of East-North-Up frame to follow aeronautical convention.

- e : earth/world frame, inertial frame, (North-East-Down)
- c : camera frame, rotated from earth frame by: $\psi \hat{e}^z$
- b : body frame, fixed in multirotor, (Forward-Right-Down)

Constants

- m : mass of multirotor
- g : acceleration of gravity

1.2 Solve for Orientation

From Newton's 2nd Law:

$$m\vec{a} = -T\hat{z}^b + mg\hat{z}^e$$

Note $\hat{z}^e$ is the down direction in the earth frame and $\hat{z}^b$ is the down direction in the body-fixed frame. Solving for Thrust (T):

$$T\hat{z}^b = m(g\hat{z}^e - \vec{a})$$

We can now solve for $\hat{z}^b$ and T:

$$\vec{t}_e \equiv m(g\hat{z}^e_e - \vec{a}_e)$$
$$T = ||\vec{t}_e||$$
$$\hat{z}^b_e = \frac{\vec{t}_e}{T}$$

This will result in a singularity if $T = 0$.

We wish to specify the yaw angle ψ, and can do so by computing the body unit vector $\hat{y}^b$ as follows:

$$\vec{x}^c_e = \begin{bmatrix}\cos\psi & \sin\psi & 0\end{bmatrix}^T$$

$$\hat{y}^b_e = \frac{\hat{z}^b_e \times \hat{x}^c_e}{||\hat{z}^b_e \times \hat{x}^c_e||}$$

This will result in a singularity if $\hat{z}^b_e$ is aligned with $\hat{x}^c_e$. We can solve for $\hat{x}^b_e$ using the cross product of unit vectors:

$$\hat{x}^b_e = \hat{y}^b_e \times \hat{z}^b_e$$

Finally, we can construct C^b_e using the unit vectors:

$$C^b_e = \begin{bmatrix}\hat{x}^b_e & \hat{y}^b_e & \hat{z}^b_e\end{bmatrix}$$

1.3 Solve for Angular Velocity

We take the derivative of the translational equation of motion with respect to frame e:

$$\vec{\omega}^{eb} \equiv p\hat{x}^b + q\hat{y}^b + r\hat{z}^b$$

$$\frac{{}^e d}{dt}\left(-m\vec{a} = T\hat{z}^b - mg\hat{z}^e\right)$$

$$-m\vec{j} = \dot{T}\hat{z}^b + \vec{\omega}^{eb} \times T\hat{z}^b$$

Taking the dot product with $\hat{z}^b$:

$$\dot{T} = -m\vec{j}_e \cdot \hat{z}^b_e$$

Taking the dot product with $\hat{x}^b$ and $\hat{y}^b$:

$$\left((p\hat{x}^b + q\hat{y}^b + r\hat{z}^b) \times \hat{z}^b\right) \cdot (\hat{x}^b + \hat{y}^b) = -\frac{m}{T}\vec{j} \cdot (\hat{x}^b + \hat{y}^b)$$

$$(-p\hat{y}^b + q\hat{x}^b) \cdot (\hat{x}^b + \hat{y}^b) = -\frac{m}{T}\vec{j} \cdot (\hat{x}^b + \hat{y}^b)$$

$$p = \frac{m}{T}\vec{j} \cdot \hat{y}^b = \frac{m}{T}\vec{j}_e \cdot \hat{y}^b_e$$

$$q = -\frac{m}{T}\vec{j} \cdot \hat{x}^b = -\frac{m}{T}\vec{j}_e \cdot \hat{x}^b_e$$

To find r we can note that the relationships between the body angular velocities and euler angle rates. Note the deviation here from Mellinger, which uses a B312 rotation. For standard Body 321 Euler angles the roll rate should be about the body x unit vector, not the c frame x unit vector. Similarly the pitch rate should be about the c frame y unit vector, not the x frame y unit vector.

$$p\hat{x}^b + q\hat{y}^b + r\hat{z}^b = \dot{\phi}\hat{x}^b + \dot{\theta}\hat{y}^c + \dot{\psi}\hat{z}^e$$

$$\hat{y}^c_e = \hat{z}^c_e \times \hat{x}^c_e$$

$$C^b_e \begin{bmatrix} p \\ q \\ r \end{bmatrix}_b = \begin{bmatrix} \hat{x}^b_e & \hat{y}^c_e & \hat{z}^e_e \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} \hat{x}^b_e & \hat{y}^c_e & \hat{z}^e_e \end{bmatrix}^{-1} C^b_e \begin{bmatrix} p \\ q \\ r \end{bmatrix}_b = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$A \equiv \begin{bmatrix} \hat{x}^b_e & \hat{y}^c_e & \hat{z}^e_e \end{bmatrix}^{-1} C^b_e$$

$$A[2,0]p + A[2,1]q + A[2,2]r = \dot{\psi}$$

$$r = (\dot{\psi} - A[2,0]p - A[2,1]q)/A[2,2]$$

Alternatively, we can use a symbolic solution in terms of the Euler angles, and compute the Euler angles from C^b_e. In this form, it is more clear to see where A becomes singular and cannot be inverted, where the pitch is 90 degrees.

$$C^b_e = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\phi)\sin(\theta)\cos(\psi) - \sin(\psi)\cos(\phi) & \sin(\phi)\sin(\psi) + \sin(\theta)\cos(\phi)\cos(\psi) \\ \sin(\psi)\cos(\theta) & \sin(\phi)\sin(\psi)\sin(\theta) + \cos(\phi)\cos(\psi) & -\sin(\phi)\cos(\psi) + \sin(\psi)\sin(\theta)\cos(\phi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix}$$

$$\tan\phi = \frac{C^b_e[2,1]}{C^b_e[2,2]}$$

$$\sin\theta = -C^b_e[2,0]$$

$$r = -q\tan\phi + \frac{\cos\theta}{\cos\phi}\dot{\psi}$$

1.4 Solve for Angular Acceleration

$$\dot{\vec{\omega}}^{eb} \equiv \dot{p}\hat{x}^b + \dot{q}\hat{y}^b + \dot{r}\hat{z}^b$$

Taking the 2nd derivative of the translational equation of motion we arrive at:

$$\frac{{}^e d}{dt}\left(-m\vec{j} = \dot{T}\hat{z}^b + \vec{\omega}^{eb} \times T\hat{z}^b\right)$$

$$-m\vec{s} = \ddot{T}\hat{z}^b + 2\vec{\omega}^{eb} \times \dot{T}\hat{z}^b + \dot{\vec{\omega}}^{eb} \times T\hat{z}^b + \vec{\omega}^{eb} \times \vec{\omega}^{eb} \times T\hat{z}^b$$

Taking the dot product with the $\hat{z}^b$ vector we arrive at:

$$\ddot{T} = -m\vec{s} \cdot \hat{z}^b - \left(\vec{\omega}^{eb} \times \vec{\omega}^{eb} \times T\hat{z}^b\right) \cdot \hat{z}^b$$

1.5 Solve for Inputs

We have previously solved for the thrust:

$$\vec{t}_e \equiv m(g\hat{z}_e^e - \vec{a}_e)$$
$$T = ||\vec{t}_e||$$

We can now solve for the body torques produced by the motors:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = J \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times J \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

References

[1] Daniel Warren Mellinger. *Trajectory Generation and Control for Quadrotors.* Publicly Accessible Penn Dissertations, 2012.