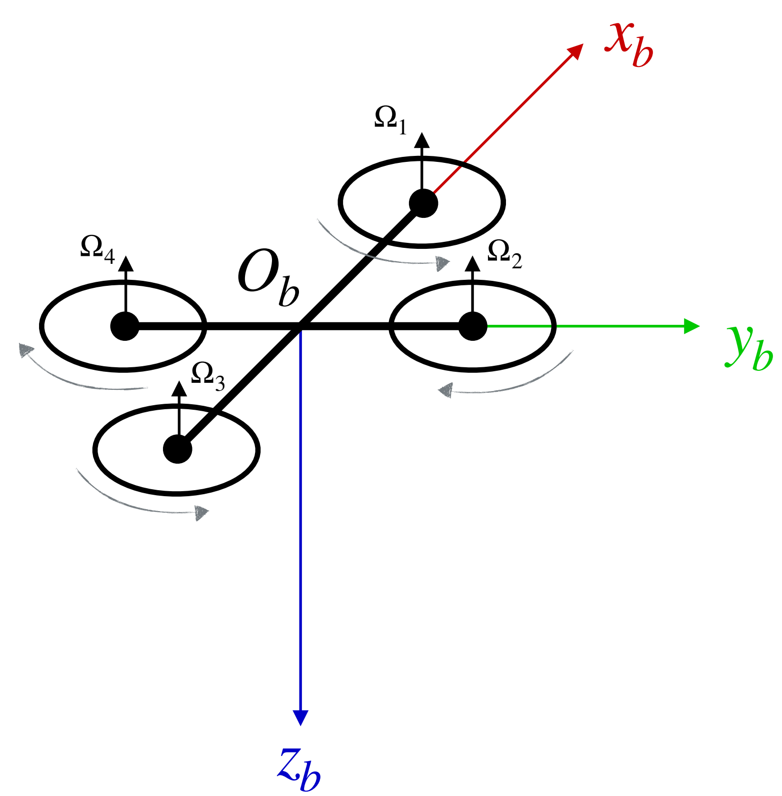
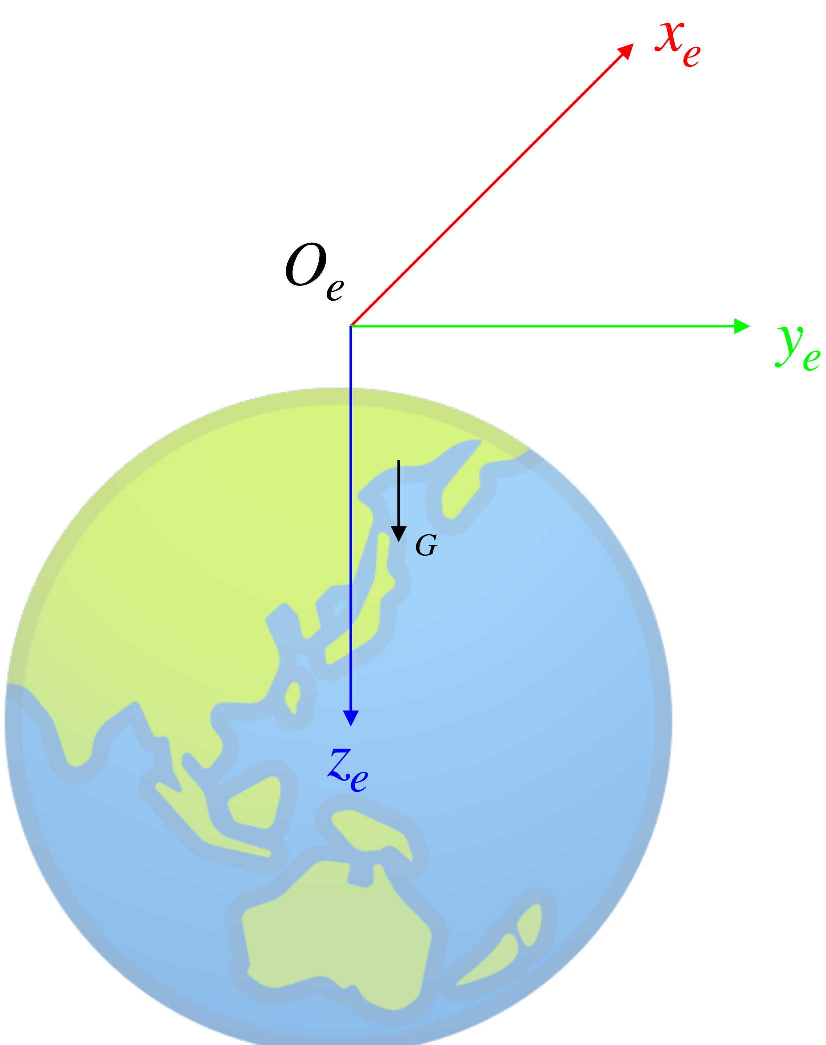
Fundamental Assumption

- The quadcopter is an uniform and symmetrical rigid body.
- The mass and inertia do not change
- The geometrical center of quadcopter is same as the center of mass
- Only two forces: gravity force and propeller thrust
- Propellers 1 and 3 rotate counterclockwise, and propellers 2 and 4 rotate clockwise.



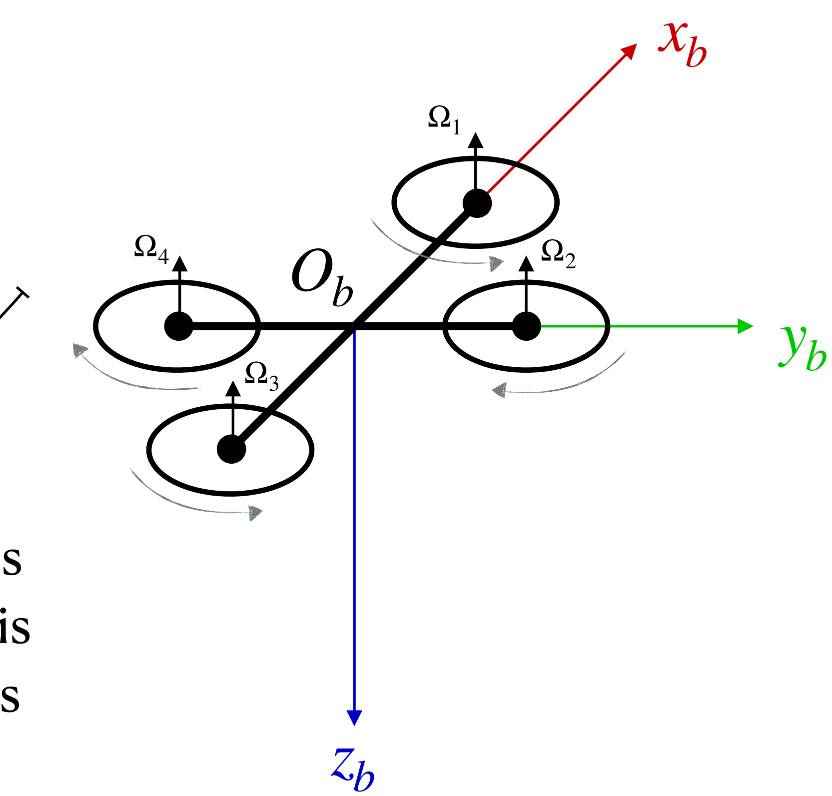
Coordinates

The relation between inertia coordinate O_e and quadcopter coordinate O_b



The rotation matrix used to transfer from quadcopter coordinate to world coordinate:

$$O_e = R_b^e O_b.$$

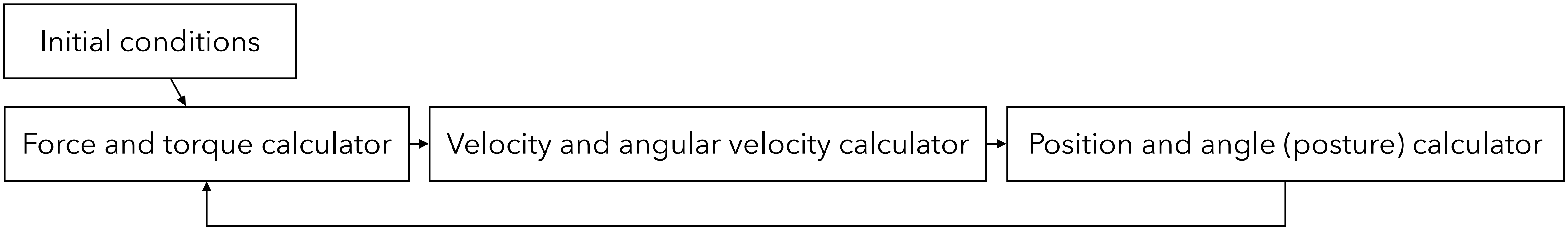


Roll: rotate ϕ around x-axis
Pitch: rotate θ around y-axis
Yaw: rotate ψ around z-axis

$$R_b^e = Z(\psi)Y(\theta)X(\phi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\psi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}$$

Rigid Body Simulation



Quadcopter Motion Theory

Initial conditions

Force and torque calculator → Velocity and angular velocity calculator → Position and angle (posture) calculator

+-Configuration

Force

$$f = c_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right)$$

Torque

$$\tau_x = dc_T\left(-\Omega_2^2 + \Omega_4^2\right)$$

$$\tau_y = dc_T\left(\Omega_1^2 - \Omega_3^2\right)$$

$$\tau_z = c_M\left(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2\right)$$

X-Configuration

Force

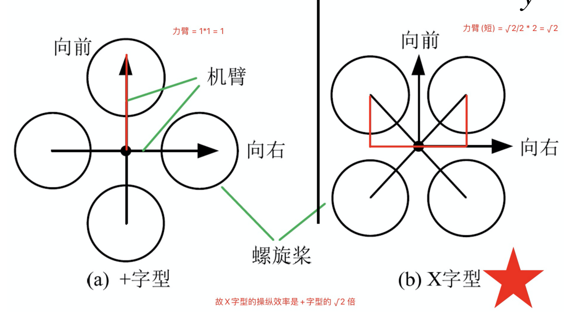
$$f = c_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right)$$

Torque

$$\tau_x = dc_T\left(\frac{\sqrt{2}}{2}\Omega_1^2 - \frac{\sqrt{2}}{2}\Omega_2^2 - \frac{\sqrt{2}}{2}\Omega_3^2 + \frac{\sqrt{2}}{2}\Omega_4^2\right)$$

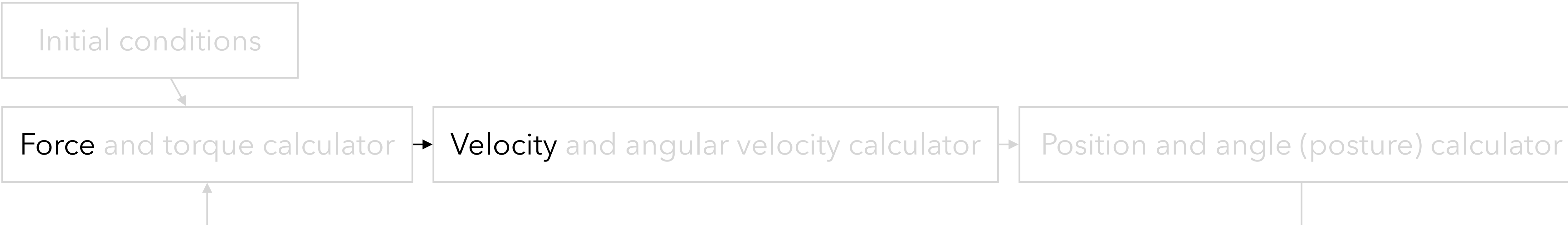
$$\tau_y = dc_T\left(\frac{\sqrt{2}}{2}\Omega_1^2 + \frac{\sqrt{2}}{2}\Omega_2^2 - \frac{\sqrt{2}}{2}\Omega_3^2 - \frac{\sqrt{2}}{2}\Omega_4^2\right)$$

$$\tau_z = c_M\left(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2\right)$$



c_T is the propeller thrust coefficient; c_M is the propeller torque coefficient; d is the distance from the center of the quadcopter to any motor.

Quadcopter Motion Theory



By Newton's second law of motion $\mathbf{F} = m\mathbf{a}$, the equation of motion is $\mathbf{a}_e = \mathbf{g}_e - \frac{\mathbf{f}_b}{m}$, where $\mathbf{g}_e = (0,\ 0,\ g)$ is gravity that defined on the world coordinate points at $+z$ direction and $\mathbf{f}_b = (0,\ 0,\ f)$ is propeller thrust that only applies z direction force on quadcopter at quadcopter coordinate.

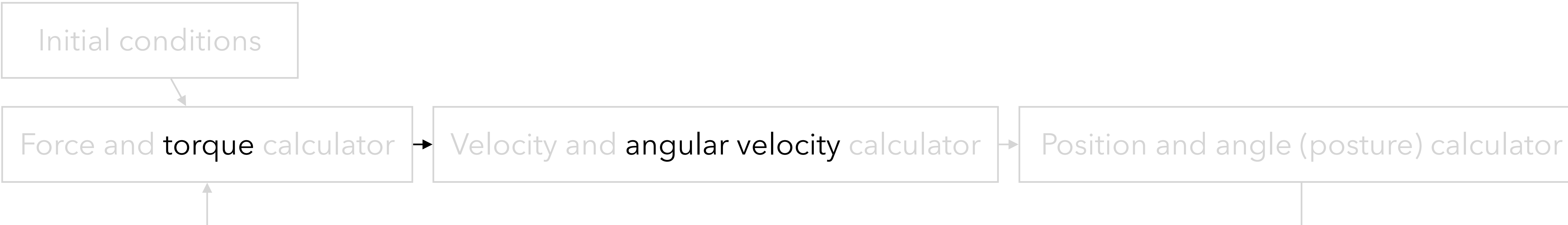
In order to calculate the quadcopter's position in world coordinate, as well as people observe it standing on earth, the thrust force should be transferred by an rotation matrix and expressed by a vector that is based on world coordinate. Therefore, $\mathbf{f}_e = R_b^e \mathbf{f}_b$ (appendix A).

Finally, $\mathbf{a}_e = \mathbf{g}_e - \frac{\mathbf{f}_e}{m} = \mathbf{g}_e - \frac{R_b^e \mathbf{f}_b}{m}$ and each component is

$$
\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\psi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{f}{m} \end{bmatrix}
$$

$$
= \begin{bmatrix} -\frac{f}{m}(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi) \\ -\frac{f}{m}(\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi) \\ g - \frac{f}{m}\cos\phi\cos\theta \end{bmatrix}
$$

Quadcopter Motion Theory



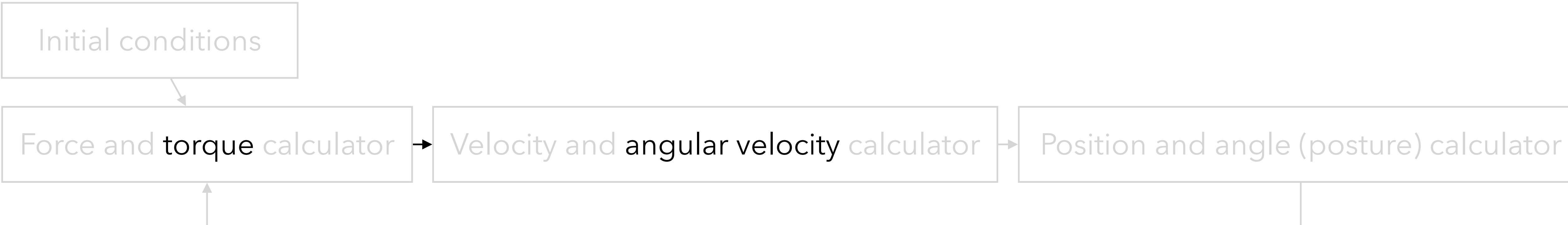
By Euler's equation $\tau_{total} = \mathbf{I}\alpha + \omega \times \mathbf{I}\omega$ (appendix B), where τ_{total} is the total applied torque, $\mathbf{I}$ is the inertia matrix, α and ω are angular acceleration and velocity. All quantities are defined in the rotating reference frame, the quadcopter coordinate here. Therefore, the rotation equation of quadcopter is $\mathbf{I}\alpha_b + \omega_b \times \mathbf{I}\omega_b = \mathbf{G}_a + \tau$, where τ is the torque generated by propellers and $\mathbf{G}_a$ is gyroscopic torque (appendix C) $\begin{bmatrix} G_{a,\phi} \\ G_{a,\theta} \\ G_{a,\psi} \end{bmatrix} = \begin{bmatrix} -J_{RP}\omega_{by}\Omega \\ J_{RP}\omega_{bx}\Omega \\ 0 \end{bmatrix}$. J_{RP} is the total moment of inertia of the quadcopter and propeller around the axis and Ω is the total angular velocity of all propellers.

According to assumption 1 and 2, the inertia matrix is $\mathbf{I} = \begin{bmatrix} I_{xx} & & \\ & I_{yy} & \\ & & I_{zz} \end{bmatrix}$.

Substitute into the rotation equation, it becomes

$$\begin{bmatrix} I_{xx} & & \\ & I_{yy} & \\ & & I_{zz} \end{bmatrix} \begin{bmatrix} \alpha_{bx} \\ \alpha_{by} \\ \alpha_{bz} \end{bmatrix} + \omega_b \times \begin{bmatrix} I_{xx} & & \\ & I_{yy} & \\ & & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = \begin{bmatrix} -J_{RP}\omega_{by}\Omega \\ J_{RP}\omega_{bx}\Omega \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

Quadcopter Motion Theory



Derivation process……

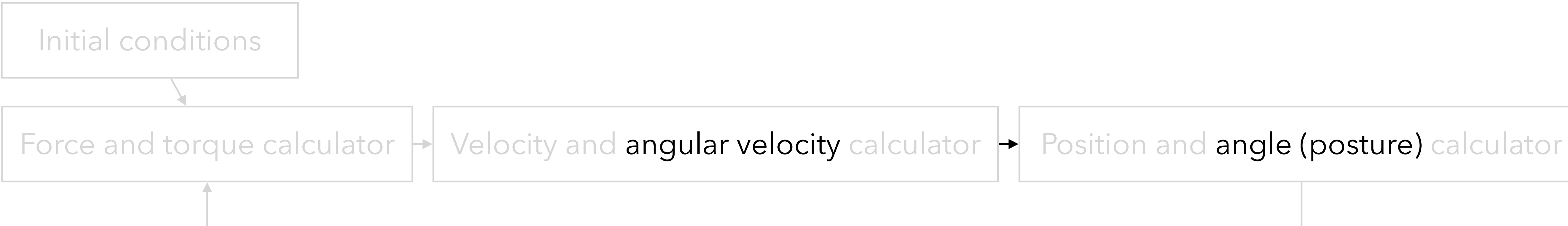
$$\begin{bmatrix} I_{xx} & & \\ & I_{yy} & \\ & & I_{zz} \end{bmatrix} \begin{bmatrix} \alpha_{bx} \\ \alpha_{by} \\ \alpha_{bz} \end{bmatrix} + \omega_b \times \begin{bmatrix} I_{xx} & & \\ & I_{yy} & \\ & & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = \begin{bmatrix} -J_{RP}\omega_{by}\Omega \\ J_{RP}\omega_{bx}\Omega \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} I_{xx}\alpha_{bx} \\ I_{yy}\alpha_{by} \\ I_{zz}\alpha_{bz} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_{bz} & \omega_{by} \\ \omega_{bz} & 0 & -\omega_{bx} \\ -\omega_{by} & \omega_{bx} & 0 \end{bmatrix} \times \begin{bmatrix} I_{xx}\omega_{bx} \\ I_{yy}\omega_{by} \\ I_{zz}\omega_{bz} \end{bmatrix} = \begin{bmatrix} -J_{RP}\omega_{by}\Omega \\ J_{RP}\omega_{bx}\Omega \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} I_{xx}\alpha_{bx} \\ I_{yy}\alpha_{by} \\ I_{zz}\alpha_{bz} \end{bmatrix} + \begin{bmatrix} \omega_{by}\omega_{bz}(I_{zz} - I_{yy}) \\ \omega_{bz}\omega_{bx}(I_{xx} - I_{zz}) \\ \omega_{bx}\omega_{by}(I_{yy} - I_{xx}) \end{bmatrix} = \begin{bmatrix} \tau_x - J_{RP}\omega_{by}\Omega \\ \tau_y + J_{RP}\omega_{bx}\Omega \\ \tau_z \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} I_{xx}\alpha_{bx} \\ I_{yy}\alpha_{by} \\ I_{zz}\alpha_{bz} \end{bmatrix} = \begin{bmatrix} \tau_x + \omega_{by}\omega_{bz}(I_{yy} - I_{zz}) - J_{RP}\omega_{by}\Omega \\ \tau_y + \omega_{bz}\omega_{bx}(I_{zz} - I_{xx}) + J_{RP}\omega_{bx}\Omega \\ \tau_z + \omega_{bx}\omega_{by}(I_{xx} - I_{yy} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha_{bx} \\ \alpha_{by} \\ \alpha_{bz} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}}\left(\tau_x + \omega_{by}\omega_{bz}(I_{yy} - I_{zz}) - J_{RP}\omega_{by}\Omega\right) \\ \frac{1}{I_{yy}}\left(\tau_y + \omega_{bz}\omega_{bx}(I_{zz} - I_{xx}) + J_{RP}\omega_{bx}\Omega\right) \\ \frac{1}{I_{zz}}\left(\tau_z + \omega_{bx}\omega_{by}(I_{xx} - I_{yy})\right) \end{bmatrix}$$

Quadcopter Motion Theory



The relation between angular velocity and the change of posture angle (appendix D) is

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

If all of the change of posture angles are small, it can be simplified as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

Quadcopter Motion Theory

Summary

Initial conditions → Force and torque calculator → Velocity and angular velocity calculator → Position and angle (posture) calculator →

Velocity (on world coordinate)

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \frac{v_{x,n+1} - v_{x,n}}{\Delta t} \\ \frac{v_{y,n+1} - v_{y,n}}{\Delta t} \\ \frac{v_{z,n+1} - v_{z,n}}{\Delta t} \end{bmatrix} = \begin{bmatrix} -\frac{f}{m}(\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi) \\ -\frac{f}{m}(\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi) \\ g - \frac{f}{m}\cos\phi \cos\theta \end{bmatrix}$$

Position (on world coordinate)

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \frac{x_{n+1} - x_n}{\Delta t} \\ \frac{y_{n+1} - y_n}{\Delta t} \\ \frac{z_{n+1} - z_n}{\Delta t} \end{bmatrix}$$

Angular velocity (on quadcopter coordinate)

$$\begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} = \begin{bmatrix} \frac{\omega_{x,n+1} - \omega_{x,n}}{\Delta t} \\ \frac{\omega_{y,n+1} - \omega_{y,n}}{\Delta t} \\ \frac{\omega_{z,n+1} - \omega_{z,n}}{\Delta t} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}}\left(\tau_x + \omega_y\omega_z(I_{yy} - I_{zz}) - J_{RP}\omega_y\Omega\right) \\ \frac{1}{I_{yy}}\left(\tau_y + \omega_z\omega_x(I_{zz} - I_{xx}) + J_{RP}\omega_x\Omega\right) \\ \frac{1}{I_{zz}}\left(\tau_z + \omega_x\omega_y(I_{xx} - I_{yy})\right) \end{bmatrix}$$

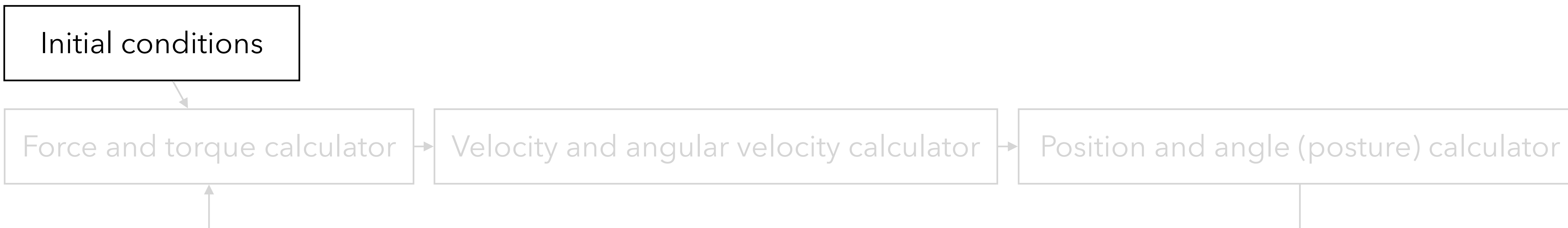
(the subscript bi of α and ω simplified as i)

Euler angles (modified from angular velocity on world coordinate)

$$\begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} = \begin{bmatrix} \frac{\phi_{n+1} - \phi_n}{\Delta t} \\ \frac{\theta_{n+1} - \theta_n}{\Delta t} \\ \frac{\psi_{n+1} - \psi_n}{\Delta t} \end{bmatrix} = \begin{bmatrix} \omega_x + \omega_y \tan\theta \sin\phi + \omega_z \tan\theta \cos\phi \\ \omega_y \cos\phi - \omega_z \sin\phi \\ \omega_y \frac{\sin\phi}{\cos\theta} + \omega_z \frac{\cos\phi}{\cos\theta} \end{bmatrix}$$

The angle should be modified by a matrix!!

Quadcopter Simulation



Constant Parameters:

$$c_T = 1.105 \times 10^{-5}$$

$$c_M = 2 \times 1.779 \times 10^{-7}$$

$$d = 0.225 \text{ m}$$

$$m = 1.4 \text{ kg}$$

$$g = 9.8 \text{ m/s}^2$$

$$I_{xx} = 0.0211 \text{ kg} \cdot \text{m}^2$$

$$I_{yy} = 0.0219 \text{ kg} \cdot \text{m}^2$$

$$I_{zz} = 0.0366 \text{ kg} \cdot \text{m}^2$$

$$J_{RP} = 0.0001287 \text{ kg} \cdot \text{m}^2$$

Physical Quantities:

Angular velocity $\left(\omega_{x,0},\ \omega_{y,0},\ \omega_{z,0}\right) = (0.0,\ 0.0,\ 0.5)$

Velocity $\left(v_{x,0},\ v_{y,0},\ v_{z,0}\right) = (0.0,\ 0.0,\ 0.0)$

Angle $\left(\phi_0,\ \theta_0,\ \psi_0\right) = (0.0,\ 0.0,\ 0.0)$

Position $\left(x,\ y,\ z\right) = (0.0,\ 0.0,\ -1.0)$

$$\Omega_k = \sqrt{\frac{mg}{4c_T}} = \sqrt{\frac{1.4 \times 9.8}{4 \times 1.105 \times 10^{-5}}} = 557.142 \text{ rad/s}$$

Quadcopter Simulation

Initial conditions

Force and torque calculator → Velocity and angular velocity calculator → Position and angle (posture) calculator

Input data

$$[\Omega_k]$$

Force and torque of a X-configuration quadcopter

(In my code, Ω_k are constant)

$$f_{n+1} = c_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right)$$

$$\tau_{x,n+1} = dc_T\left(\frac{\sqrt{2}}{2}\Omega_1^2 - \frac{\sqrt{2}}{2}\Omega_2^2 - \frac{\sqrt{2}}{2}\Omega_3^2 + \frac{\sqrt{2}}{2}\Omega_4^2\right)$$

$$\tau_{y,n+1} = dc_T\left(\frac{\sqrt{2}}{2}\Omega_1^2 + \frac{\sqrt{2}}{2}\Omega_2^2 - \frac{\sqrt{2}}{2}\Omega_3^2 - \frac{\sqrt{2}}{2}\Omega_4^2\right)$$

$$\tau_{z,n+1} = c_M\left(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2\right)$$

Output data

$$\begin{bmatrix} f_{n+1} \\ \tau_{x,n+1} \\ \tau_{y,n+1} \\ \tau_{z,n+1} \end{bmatrix}$$

Quadcopter Simulation

Implementation by programming

Initial conditions

Force and torque calculator → Velocity and angular velocity calculator → Position and angle (posture) calculator

Input data at time n

$$\begin{bmatrix} f_n \\ \tau_{x,n} \\ \tau_{y,n} \\ \tau_{z,n} \end{bmatrix}$$

Difference equation of velocity (on world coordinate) and angular velocity (on quadcopter coordinate)

$$v_{x,n+1} = v_{x,n} - \Delta t \frac{f_n}{m}(\cos\psi_n \sin\theta_n \cos\phi_n + \sin\psi_n \sin\phi_n)$$

$$v_{y,n+1} = v_{y,n} - \Delta t \frac{f_n}{m}(\sin\psi_n \sin\theta_n \cos\phi_n - \cos\psi_n \sin\phi_n)$$

$$v_{z,n+1} = v_{z,n} + \Delta t g - \Delta t \frac{f_n}{m} \cos\phi_n \cos\theta_n$$

$$\omega_{x,n+1} = \omega_{x,n} + \frac{\Delta t}{I_{xx}} \left(\tau_{x,n} + \omega_{y,n}\omega_{z,n}(I_{yy} - I_{zz}) - J_{RP}\omega_{y,n}\Omega_n \right)$$

$$\omega_{y,n+1} = \omega_{y,n} + \frac{\Delta t}{I_{yy}} \left(\tau_{y,n} + \omega_{z,n}\omega_{x,n}(I_{zz} - I_{xx}) + J_{RP}\omega_{x,n}\Omega_n \right)$$

$$\omega_{z,n+1} = \omega_{z,n} + \frac{\Delta t}{I_{zz}} \left(\tau_{z,n} + \omega_{x,n}\omega_{y,n}(I_{xx} - I_{yy}) \right)$$

Output data at time $n+1$

$$\begin{bmatrix} v_{x,n+1} \\ v_{y,n+1} \\ v_{z,n+1} \\ \omega_{x,n+1} \\ \omega_{y,n+1} \\ \omega_{z,n+1} \end{bmatrix}$$

Quadcopter Simulation

Implementation by programming

Initial conditions

Force and torque calculator → Velocity and angular velocity calculator → Position and angle (posture) calculator

Input data

$$\begin{bmatrix} v_{x,n+1} \\ v_{y,n+1} \\ v_{z,n+1} \\ \omega_{x,n+1} \\ \omega_{y,n+1} \\ \omega_{z,n+1} \end{bmatrix}$$

Difference equation of position (on world coordinate) and angle (posture) of quadcopter

$$x_{n+1} = x_n + \Delta t \, v_{x,n+1}$$

$$y_{n+1} = y_n + \Delta t \, v_{y,n+1}$$

$$z_{n+1} = z_n + \Delta t \, v_{z,n+1}$$

$$\phi_{n+1} = \phi_n + \Delta t \left(\omega_{x,n+1} + \omega_{y,n+1} \tan\theta_n \sin\phi_n + \omega_{z,n+1} \tan\theta_n \cos\phi_n \right)$$

$$\theta_{n+1} = \theta_n + \Delta t \left(\omega_{y,n+1} \cos\phi_n - \omega_{z,n+1} \sin\phi_n \right)$$

$$\psi_{n+1} = \psi_n + \Delta t \left(\omega_{y,n+1} \frac{\sin\phi_n}{\cos\theta_n} + \omega_{z,n+1} \frac{\cos\phi_n}{\cos\theta_n} \right)$$

Output data

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \\ \phi_{n+1} \\ \theta_{n+1} \\ \psi_{n+1} \end{bmatrix}$$

Quadcopter Simulation

Implementation by programming

Initial conditions

Force and torque calculator → Velocity and angular velocity calculator → Position and angle (posture) calculator

Input data

$$[\Omega_k]$$

Force and torque of a X-configuration quadcopter

(In my code, Ω_k are constant)

$$f_{n+1} = c_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right)$$

$$\tau_{x,n+1} = dc_T\left(\frac{\sqrt{2}}{2}\Omega_1^2 - \frac{\sqrt{2}}{2}\Omega_2^2 - \frac{\sqrt{2}}{2}\Omega_3^2 + \frac{\sqrt{2}}{2}\Omega_4^2\right)$$

$$\tau_{y,n+1} = dc_T\left(\frac{\sqrt{2}}{2}\Omega_1^2 + \frac{\sqrt{2}}{2}\Omega_2^2 - \frac{\sqrt{2}}{2}\Omega_3^2 - \frac{\sqrt{2}}{2}\Omega_4^2\right)$$

$$\tau_{z,n+1} = c_M\left(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2\right)$$

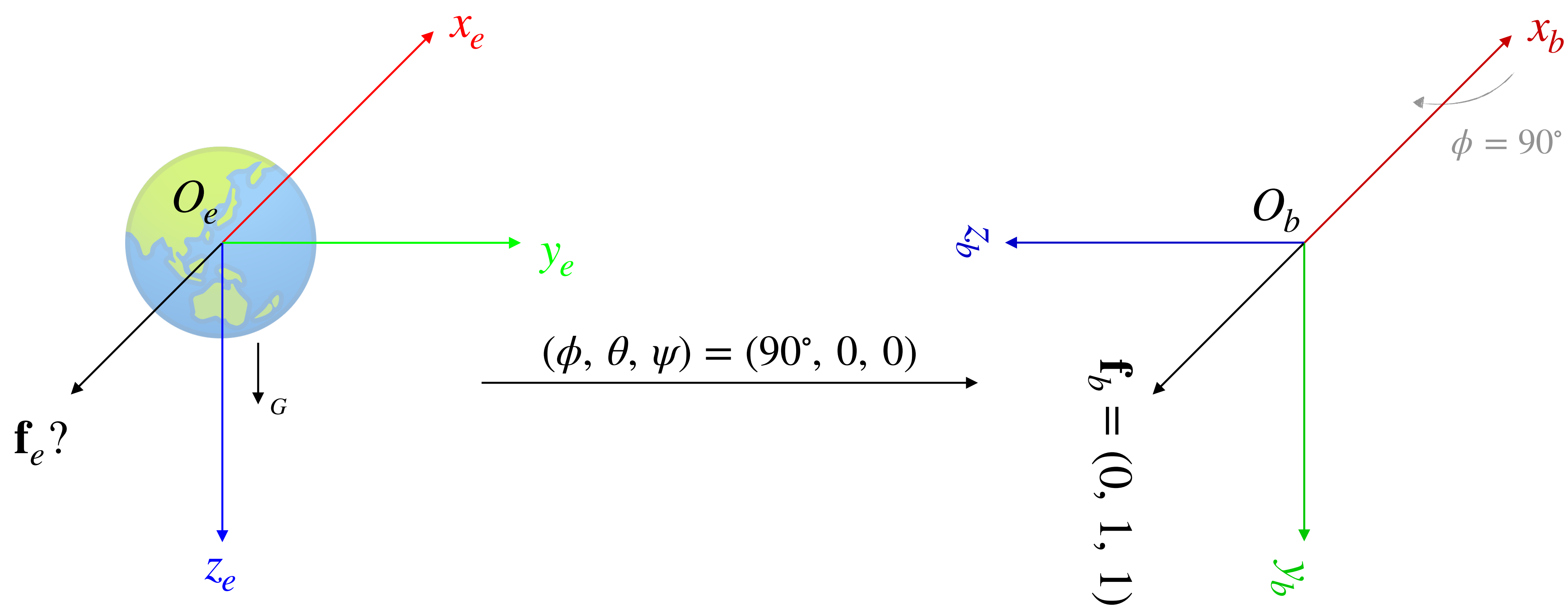
Output data

$$\begin{bmatrix} f_{n+1} \\ \tau_{x,n+1} \\ \tau_{y,n+1} \\ \tau_{z,n+1} \end{bmatrix}$$

Appendix A

Rotation matrix example

The thrust force $\mathbf{f_b}$ is defined on quadcopter coordinates. In order to express on world coordinates, the rotation matrix R_b^e should be applied on it. Here is the example.



$$\mathbf{f}_e = R_b^e \mathbf{f}_b = \begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\psi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

Appendix B

Euler's equations (rigid body dynamics)

Euler's equation of motion is a vectorial quasilinear first-order ODE which describes the rotation of a rigid body in world coordinate. I want to compare the concept with the Lagrangian equation in fluid dynamics.

Lagrangian equation in fluid dynamics

$$\frac{D\mathbf{F}}{Dt} = \frac{\partial \mathbf{F}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{F}$$

First term: change observed at a fixed point (world coordinate)

Second term: caused by the movement of fluid particles, which carry them to other positions

Euler's equation of motion

$$\left(\frac{d\mathbf{L}}{dt}\right)_e = \left(\frac{d\mathbf{L}}{dt}\right)_b + \omega \times \mathbf{L}$$

First term: change observed in the object's own rotation coordinate

Second term: caused by the rotation of the rotation coordinate itself

$\mathbf{L} = \mathbf{I}_b \omega$, where $\mathbf{I}_b$ is the inertia tenser expressed in the rotation coordinate, it is a constant matrix for a rigid body. Therefore, the net external torque is

$$\tau_{total} = \omega \cancel{\frac{d\mathbf{I}_b}{dt}} + \mathbf{I}_b \frac{d\omega}{dt} + \omega \times \mathbf{I}_b \omega = \mathbf{I}_b \alpha + \omega \times \mathbf{I}_b \omega$$

$$\rightarrow \tau_{total} = \mathbf{I}_b \alpha + \omega \times \mathbf{I}_b \omega$$

Appendix C

gyroscopic torque (陀螺力矩)

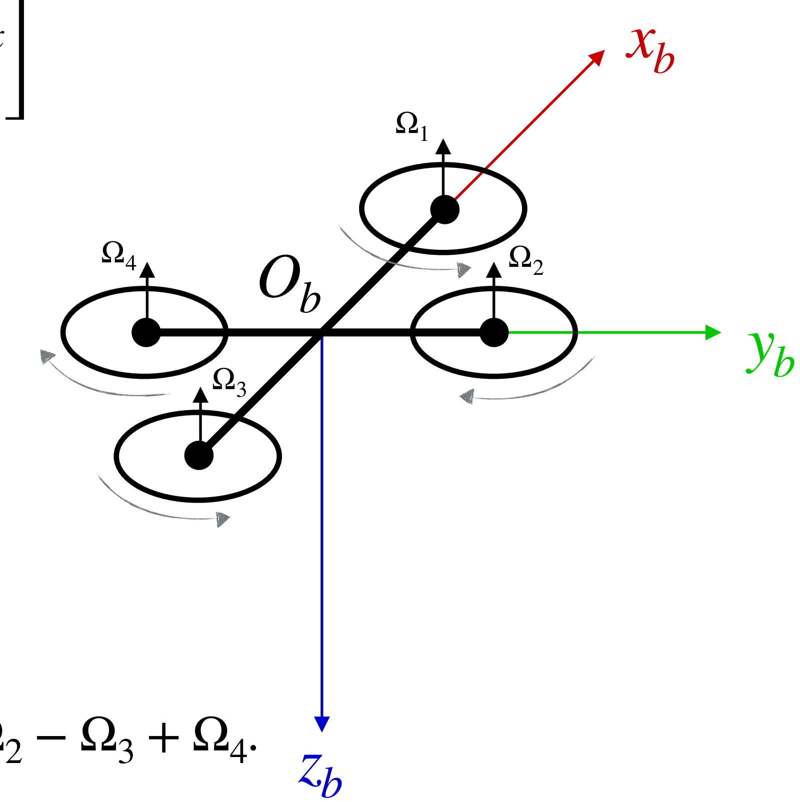
In the rotation equation of quadcopter, $\mathbf{I}\alpha_b + \omega_b \times \mathbf{I}\omega_b = \mathbf{G}_a + \tau$. $\mathbf{G}_a$ is the total gyroscopic torque of all the propellers. The gyroscopic torque of the propeller k is defined as

$$\mathbf{G}_{a,k} = J_{RP}\mathbf{\Omega}_k \times \omega$$

, where J_{RP} is the total moment of inertia of the quadcopter and propeller around the axis, $\mathbf{\Omega}_k$ is the angular velocity of propeller pointing at z direction, and ω is the angular velocity of the quadcopter's rotation axis. By expand the equation

$$\mathbf{G}_{a,k} = J_{RP}\mathbf{\Omega}_k \times \omega = J_{RP}\Omega_k \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \omega = -J_{RP}\Omega_k \begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix}$$

, where $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \omega = -\omega \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = -\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = -\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix}$.



In the figure, the angular velocity is $\mathbf{\Omega}_k = (-1)^k\Omega$ according to right-hand rule.

Finally, each component of gyroscopic torque shown as

$$\mathbf{G}_a = \begin{bmatrix} G_{a,\phi} \\ G_{a,\theta} \\ G_{a,\psi} \end{bmatrix} = \begin{bmatrix} J_{RP}\omega_y(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) \\ J_{RP}\omega_x(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) \\ 0 \end{bmatrix} = \begin{bmatrix} -J_{RP}\omega_y\Omega \\ J_{RP}\omega_x\Omega \\ 0 \end{bmatrix}$$

, where $\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$.

Appendix D

Relation between angular velocity and Euler angles

The angular velocity ω_b we calculated is defined in the quadcopter coordinate and the Euler angles (ϕ, θ, ψ) are defined in world coordinate, the rotation matrix can modified time derivation of the Euler angles by angular velocity. Since the rotation order is X, Y, then Z, we have

$$\begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = R(-\phi)R(-\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R(-\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{M} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}.$$

By calculating the inverse matrix of $\mathbf{M}$,

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}.$$

Therefore, the time derivative of the Euler angles should be modified by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$