Stabilisation & Hover Algorithm

MATLAB Code Documentation

List of m-files:

1. **main.m**
   Sets up the physical parameters, initial conditions, and desired conditions.
   Calls the simulation-function.
   Plots the results of the simulation against time. (Linear position & velocity, angular position & velocity, rotor velocities).

2. **simuilation.m**
   Unloads the structures.
   Initialises variables for storing data. Variables labelled _data are used to store data for each increment.
   The while-loop iterates to run calculations in this sequence:
   a. Update time-vector.
   b. Update iteration count.
   c. Call PID function.
   d. Call motor-mixing algorithm.
   e. Calculate forces for each rotor.
   f. Calculate torques for each rotor.
   g. Calculate the total thrust.
   h. Calculate the total torque vector.
   i. Call Linear Acceleration function.
   j. Call Angular Acceleration function.
   k. Use Euler's method to estimate velocity.
   l. Use Euler's method to estimate position.
   m. Update data-saving matrices.
   n. Check termination-conditions.

3. **PID.m**
   PID Controller

4. **motormix.m**
   Converting PID output to rotor velocities.

5. **lin_acc.m**
   Newtonian linear acceleration calculation.

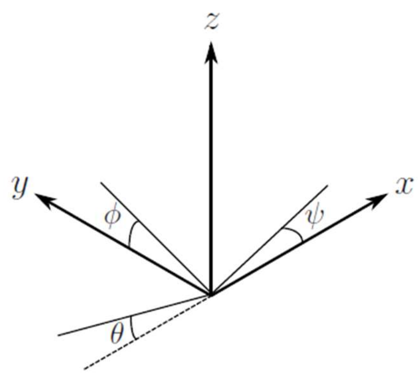
6. **ang_acc.m**
   Newtonian angular acceleration calculation.

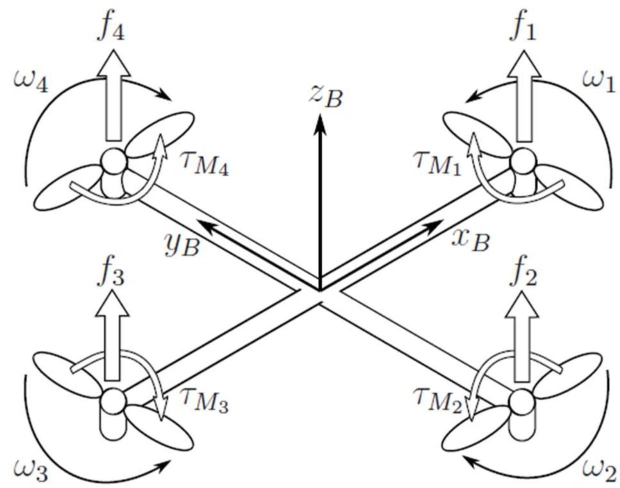
7. **rotation.m**
   Rotation matrix: converts body frame to global frame.
   Note: $R^{-1} = R^T$ if the rotation matrix for global frame to body frame.

8. **trans_mat.m**
   Transformation matrix for angular velocities from the global frame to the body frame.

■■■

Equations





main.m

$$x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$n = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \rightarrow Angles\ in\ Global\ Frame$$

$$v = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \rightarrow Angles\ in\ Body\ Frame$$

simulation.m

Time & Iteration

$$t_{i+1} = i + dt \tag{1}$$

$$i = i + 1 \tag{2}$$

where, i is the iteration count.

Physical Equations

assuming thrust and torque is proportional to the square of the rotor velocity.

$$f_i = k.\omega_i^2 \tag{3}$$

$$\tau_{Mi} = b.\omega_i^2 + I_M.\omega_i \approx b.\omega_i^2 \tag{4}$$

$$T = \sum_{i=1}^{4} f_i \tag{5}$$

$$T_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{6}$$

$$\tau_B = \begin{bmatrix} l.k.(-\omega_2^2 + \omega_4^2) \\ l.k.(-\omega_1^2 + \omega_3^2) \\ \tau_{M1} - \tau_{M2} + \tau_{M3} - \tau_{M4} \end{bmatrix} \tag{7}$$

where, i is the number of rotor, per figure.

Euler's Increment

$$\dot{n} = \dot{n} + \ddot{n}.dt \tag{8}$$

$$n = n + \dot{n}.dt \tag{9}$$

$$\dot{x} = \dot{x} + \ddot{x}.dt \tag{10}$$

$$x = x + \dot{x}.dt \tag{11}$$

PID.m

Error Calculation

$$L(t) = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}$$

$$L_e = L_c - L_d \tag{12}$$

Numerical Integration using Trapezoid Method

$$L_eI = \sum_{k=0}^{t-dt} \frac{L_e(k) + L_e(k+dt)}{2}.dt \tag{13}$$

$$L_eD = \frac{L_e(t_i) - L_e(t_{i-1})}{dt} \tag{14}$$

Control Law

$$u = K_p.L_e + K_d.L_eD + K_i.L_eI \tag{15}$$

where, u is a 6×1 *vector of the forces and torques to be applied to the aircraft in the global frame.*

motormix.m

Rotation Matrix: Body to Global Frame,

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{16}$$

where, $C_x = \cos(x)$ *and* $S_x = \sin(x)$.

Rotation Matrix: Global to Body Frame,

$$R = R' \tag{16a}$$

Accounting for Gravitational Force,

$$u = u - \begin{bmatrix} 0 \\ 0 \\ m.g \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ m.g \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

Converting u to local frame,

$$u_L = \begin{bmatrix} R & z \\ z & R \end{bmatrix} . u \tag{18}$$

$$u_L = \begin{bmatrix} 0 \\ 0 \\ L_f \\ t_\phi \\ t_\theta \\ t_\psi \end{bmatrix}$$

Solving for rotor velocity squares using linear systems,

from equations (6) & (7),

$$\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 = \frac{L_f}{k}$$

$$\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 = \frac{t_z}{b}$$

$$-\omega_2^2 + \omega_4^2 = \frac{t_x}{l.k}$$

$$-\omega_1^2 + \omega_3^2 = \frac{t_y}{l.k}$$

therefore,

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \end{bmatrix} . \vec{\omega}^2 = \begin{bmatrix} \frac{L_f}{k} \\ \frac{t_z}{b} \\ \frac{t_x}{l.k} \\ \frac{t_y}{l.k} \end{bmatrix} \tag{19}$$

where,

$$\vec{\omega}^2 = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

lin_acc.m

Linear Acceleration in Global Frame,

$$m\ddot{x} = mg + \boldsymbol{R}T_B + \boldsymbol{A}\dot{x}$$

$$\ddot{x} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{1}{m} . \boldsymbol{R} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} - \frac{1}{m} \begin{bmatrix} Ax & 0 & 0 \\ 0 & Ay & 0 \\ 0 & 0 & Az \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{20}$$

where, R is the rotation matrix: body frame to global frame.
Ax, Ay, and Az are drag coefficients of drag for velocity.

ang_ang.m

Transformation matrix for angular velocities from the global frame to the body frame,

$$W_n = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta S_\phi \end{bmatrix} \tag{21}$$

where, $C_x = \cos(x)$ *and* $S_x = \sin(x)$.
such that,

$$\nu = W_n . \dot{n} \tag{22}$$

Gyroscopic effects,

$$\omega_\Gamma = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{23}$$

External torque,

$$\tau = \boldsymbol{I}\dot{\nu} + \nu \times (\boldsymbol{I}\nu) + \Gamma$$

$$\begin{aligned} \textit{where, } \boldsymbol{I}\dot{\nu} &\rightarrow acceleration\ of\ the\ inertia \\ \nu \times (\boldsymbol{I}\nu) &\rightarrow centripetal\ forces \\ \Gamma &\rightarrow gyroscopic\ forces \\ \boldsymbol{I} &\rightarrow moment\ of\ inertia\ matrix \end{aligned}$$

Angular velocity in body frame,

$$\dot{\nu} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boldsymbol{I}^{-1} \left(-\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}.\,p \\ I_{yy}.\,q \\ I_{zz}.\,r \end{bmatrix} - I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_\Gamma + \tau \right) \tag{24}$$

Angular acceleration in global frame,

$$\ddot{n} = \frac{d}{dt}(\boldsymbol{W}_n^{-1}\,.\,\nu) = \frac{d}{dt}(\boldsymbol{W}_n^{-1})\,\nu + \boldsymbol{W}_n^{-1}\,\dot{\nu}$$

$$\ddot{n} = \begin{bmatrix} 0 & \dot{\phi}C_\phi T_\theta + \dfrac{\dot{\theta}S_\phi}{C_\theta^2} & -\dot{\phi}S_\phi C_\theta + \dfrac{\dot{\theta}C_\phi}{C_\theta^2} \\ 0 & -\dot{\phi}S_\phi & -\dot{\phi}C_\phi \\ 0 & \dfrac{\dot{\phi}C_\phi}{C_\theta} + \dfrac{\dot{\phi}S_\phi T_\theta}{C_\theta} & -\dfrac{\dot{\phi}S_\phi}{C_\theta} + \dfrac{\dot{\theta}C_\phi T_\theta}{C_\theta} \end{bmatrix} \nu + \boldsymbol{W}_n^{-1}\,\dot{\nu} \tag{25}$$

■■■