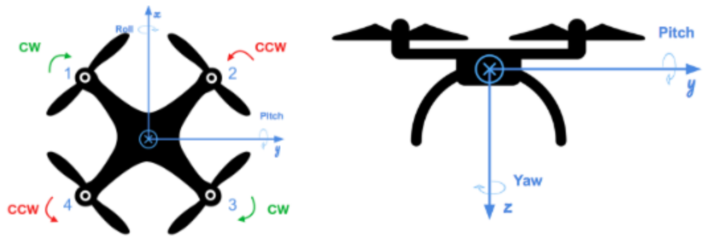
Author: Ambika Prasad Dahal

Kinematics



Let "N" be the Newtonian Frame. We are going to assume a Roll-Pitch-Yaw Euler Sequence. First, the body will Roll around X-axis of some intermediate frame (B2) with angle phi (ϕ).Then, the body will Pitch around Y-axis of another intermediate frame (B1) with angle theta (θ). Finally, the body will Yaw around Z-axis of the Body Frame (B) with angle psi (ψ).

Defining Rotations and Forumulating Rotation Matrix for Body to Newtonian Transformation.

```
% Defining some symbols
syms phi theta psi

% Defining Rotation around x-axis (Roll) of B2 Frame.
NtoB2 = [1 0 0; 0 cos(phi) -sin(phi); 0 sin(phi) cos(phi)]
```

NtoB2 =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

```
% Defining Rotation around y-axis (Pitch) of B1 Frame.
B2toB1 = [cos(theta) 0 sin(theta); 0 1 0; -sin(theta) 0 cos(theta)]
```

B2toB1 =

$$\begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix}$$

```
% Defining Rotation around z-axis (Yaw) of B Frame.
B1toB = [cos(psi) -sin(psi) 0; sin(psi) cos(psi) 0; 0 0 1]
```

B1toB =

$$\begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
% To transform from Body Frame B to Newtonian Frame N, we need Rotation Matrix (NtoB).
NtoB = B1toB * B2toB1 * NtoB2 % yaw * pitch *roll
```

NtoB =

$$\begin{pmatrix} \cos(\psi)\cos(\theta) & \cos(\psi)\sin(\varphi)\sin(\theta)-\cos(\varphi)\sin(\psi) & \sin(\varphi)\sin(\psi)+\cos(\varphi)\cos(\psi)\sin(\theta) \\ \cos(\theta)\sin(\psi) & \cos(\varphi)\cos(\psi)+\sin(\varphi)\sin(\psi)\sin(\theta) & \cos(\varphi)\sin(\psi)\sin(\theta)-\cos(\psi)\sin(\varphi) \\ -\sin(\theta) & \cos(\theta)\sin(\varphi) & \cos(\varphi)\cos(\theta) \end{pmatrix}$$

Formulating the Rate of Change of Euler Angles from the Body Angular Velocities.

phi_dot, theta_dot, psi_dot are Euler Rates.

p,q,r are Body Rates.

```
syms phi_dot theta_dot psi_dot p q r
Eul_dot = [p;q;r] == NtoB2*B2toB1*[0;0;psi_dot]+ NtoB2 * [0; theta_dot; 0]+ [phi_dot;0
expand(Eul_dot)
```

ans =

$$\begin{pmatrix} p = \varphi_{\mathrm{dot}} + \psi_{\mathrm{dot}}\sin(\theta) \\ q = \theta_{\mathrm{dot}}\cos(\varphi) - \psi_{\mathrm{dot}}\cos(\theta)\sin(\varphi) \\ r = \theta_{\mathrm{dot}}\sin(\varphi) + \psi_{\mathrm{dot}}\cos(\varphi)\cos(\theta) \end{pmatrix}$$

Clearly there is a transformation matrix such that Euler_Rates = Body2Euler_Rates * Body_Rates.

```
Euler2Body_Rates=[1 0 sin(theta);0 cos(phi) -cos(theta)*sin(phi); 0 sin(phi) cos(phi)*c
```

Test if we can get same R.H.S. equations as above.

```
Body_Rates=Euler2Body_Rates*[phi_dot;theta_dot;psi_dot]
```

Body_Rates =

$$\begin{pmatrix} \varphi_{\mathrm{dot}} + \psi_{\mathrm{dot}}\sin(\theta) \\ \theta_{\mathrm{dot}}\cos(\varphi) - \psi_{\mathrm{dot}}\cos(\theta)\sin(\varphi) \\ \theta_{\mathrm{dot}}\sin(\varphi) + \psi_{\mathrm{dot}}\cos(\varphi)\cos(\theta) \end{pmatrix}$$

```
Body2Euler_Rates=simplify(inv(Euler2Body_Rates))
```

Body2Euler_Rates =

$$\begin{pmatrix} 1 & \frac{\sin(\varphi)\sin(\theta)}{\cos(\theta)} & -\frac{\cos(\varphi)\sin(\theta)}{\cos(\theta)} \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\frac{\sin(\varphi)}{\cos(\theta)} & \frac{\cos(\varphi)}{\cos(\theta)} \end{pmatrix}$$

Dynamics

Forces

The Forces produced by Propellers always act alongs z-axis of Body Frame (B). So, the net thrust on the body is the summation of thrusts produced by individual propellers.

```
syms kf omega1 omega2 omega3 omega4
F1 = kf*omega1^2
```

F1 = $\mathrm{kf}\,\omega_1^2$

```
F2 = kf*omega2^2
```

F2 = $\mathrm{kf}\,\omega_2^2$

```
F3 = kf*omega3^2
```

F3 = $\mathrm{kf}\,\omega_3^2$

```
F4 = kf*omega4^2
```

F4 = $\mathrm{kf}\,\omega_4^2$

```
Fx_body = 0;
Fy_body = 0;
Fz_body = (F1+F2+F3+F4); % Total Thrust on Drone as seen from Body Frame.
F_Body = [Fx_body; Fy_body; -Fz_body]
```

F_Body =

$$\begin{pmatrix} 0 \\ 0 \\ -\mathrm{kf}\,\omega_1^2 - \mathrm{kf}\,\omega_2^2 - \mathrm{kf}\,\omega_3^2 - \mathrm{kf}\,\omega_4^2 \end{pmatrix}$$

The gravity vector "g" is always pointing downwards (i.e. along z-axis of Newtonian Frame) and Fz_body points upward (-ve z-axis of Body Frame according to our schematic).

Linear Accelerations as seen from Newtonian Frame Derived From Forces as seen from Newtonian Frame.

From Newton's Equations, Fnet = m*acceleration.

We need to transform "acceleration seen from Body Frame" to "acceleration seen from Newtonian Frame".

```
syms g m x_ddot y_ddot z_ddot
accl_body = (F_Body)/m
```

accl_body =

$$\begin{pmatrix} 0 \\ 0 \\ -\frac{\mathrm{kf}\,\omega_1^2+\mathrm{kf}\,\omega_2^2+\mathrm{kf}\,\omega_3^2+\mathrm{kf}\,\omega_4^2}{m} \end{pmatrix}$$

```
linacc_Newtonian = [x_ddot;y_ddot;z_ddot] == NtoB * accl_body + [0;0;g];
linacc_sol = solve(linacc_Newtonian,[x_ddot,y_ddot,z_ddot]);
x_ddot = linacc_sol.x_ddot
```

x_ddot =

$$-\frac{(\sin(\varphi)\sin(\psi)+\cos(\varphi)\cos(\psi)\sin(\theta))\ \left(\mathrm{kf}\,\omega_1^2+\mathrm{kf}\,\omega_2^2+\mathrm{kf}\,\omega_3^2+\mathrm{kf}\,\omega_4^2\right)}{m}$$

```
y_ddot = linacc_sol.y_ddot
```

y_ddot =

$$\frac{(\cos(\psi)\sin(\varphi)-\cos(\varphi)\sin(\psi)\sin(\theta))\ \left(\mathrm{kf}\,\omega_1^2+\mathrm{kf}\,\omega_2^2+\mathrm{kf}\,\omega_3^2+\mathrm{kf}\,\omega_4^2\right)}{m}$$

```
z_ddot = linacc_sol.z_ddot
```

z_ddot =

$$g-\frac{\cos(\varphi)\cos(\theta)\ \left(\mathrm{kf}\,\omega_1^2+\mathrm{kf}\,\omega_2^2+\mathrm{kf}\,\omega_3^2+\mathrm{kf}\,\omega_4^2\right)}{m}$$

Torques/Moments on Body

Note: Propeller 1 and 3 rotate clockwise producing counterclockwise reaction moment on body. Similarly, Propeller 2 and 4 rotate counterclockwise producing clockwise reaction moment on body.

Let 'L' be rotor to rotor distance and 'l' be the perpendicular distances from axes to the rotors.

The moments around x and y axes are generated by Thrust.

The moment around z axis is generated by rotational speed of propellers.

```
syms L l km
l = L/(2*sqrt(2)); % l^2 + l^2 = (L/2)^2
T1 = -km*omega1^2;
```

```
T2 = km*omega2^2;
T3 = -km*omega3^2;
T4 = km*omega4^2;
% Moment Generation around x,y,z-axes of body
M_x = (F1+F4-F2-F3)*l
```

M_x =

$$\frac{\sqrt{2}\ L\ \left(\mathrm{kf}\,\omega_1^2 - \mathrm{kf}\,\omega_2^2 - \mathrm{kf}\,\omega_3^2 + \mathrm{kf}\,\omega_4^2\right)}{4}$$

```
M_y = (F1+F2-F3-F4)*l
```

M_y =

$$\frac{\sqrt{2}\ L\ \left(\mathrm{kf}\,\omega_1^2 + \mathrm{kf}\,\omega_2^2 - \mathrm{kf}\,\omega_3^2 - \mathrm{kf}\,\omega_4^2\right)}{4}$$

```
M_z = T1 + T2 + T3 + T4
```

M_z = $-\mathrm{km}\,\omega_1^2 + \mathrm{km}\,\omega_2^2 - \mathrm{km}\,\omega_3^2 + \mathrm{km}\,\omega_4^2$

Setting the propeller Angular velocities

Based on the Inputs from the controller, we set up angular velocities of propellers. To do so, we derive following linear equations from vertical acceleration (g_ddot), roll(M_x), pitch(M_y), and yaw (M_z) equations.

```
syms c_bar p_bar q_bar r_bar Ix Iy Iz
set_omegaEqn = [c_bar; p_bar; q_bar; r_bar]== [1 1 1 1; 1 -1 -1 1; 1 1 -1 -1; 1 -1 1 -
```

set_omegaEqn =

$$\begin{pmatrix} c_{\mathrm{bar}} = \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 \\ p_{\mathrm{bar}} = \omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2 \\ q_{\mathrm{bar}} = \omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2 \\ r_{\mathrm{bar}} = \omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 \end{pmatrix}$$

The above Dimensionless Forms are derived as follow;

```
c_bar = simplify(Fz_body/kf)
```

c_bar = $\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2$

```
p_bar = simplify(Ix*(M_x/Ix)/(kf*l))
```

p_bar = $\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2$

```
q_bar = simplify(Iy*(M_y/Iy)/(kf*l))
```

q_bar = $\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2$

```
r_bar = simplify(Iz*(M_z/Iz)/(km))
```

r_bar = $-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2$

Deriving rates of p,q,r using Euler's Rotational Equation

M = **I** omega_dot + omega x (**I** omega)

```
syms p_dot q_dot r_dot
eqn1 = M_x == Ix*p_dot + (Iz-Iy)*q*r;
eqn2 = M_y == Iy*q_dot + (Ix-Iz)*p*r;
eqn3 = M_z == Iz*r_dot + (Iy-Ix)*q*p;
p_dot = solve(eqn1,p_dot)
```

p_dot =

$$\frac{q\,r\,(\mathrm{Iy} - \mathrm{Iz}) + \frac{\sqrt{2}\,L\,\left(\mathrm{kf}\,\omega_1^2 - \mathrm{kf}\,\omega_2^2 - \mathrm{kf}\,\omega_3^2 + \mathrm{kf}\,\omega_4^2\right)}{4}}{\mathrm{Ix}}$$

```
q_dot = solve(eqn2,q_dot)
```

q_dot =

$$-\frac{p\,r\,(\mathrm{Ix} - \mathrm{Iz}) - \frac{\sqrt{2}\,L\,\left(\mathrm{kf}\,\omega_1^2 + \mathrm{kf}\,\omega_2^2 - \mathrm{kf}\,\omega_3^2 - \mathrm{kf}\,\omega_4^2\right)}{4}}{\mathrm{Iy}}$$

```
r_dot = solve(eqn3,r_dot)
```

r_dot =

$$\frac{-\mathrm{km}\,\omega_1^2 + \mathrm{km}\,\omega_2^2 - \mathrm{km}\,\omega_3^2 + \mathrm{km}\,\omega_4^2 + p\,q\,(\mathrm{Ix} - \mathrm{Iy})}{\mathrm{Iz}}$$