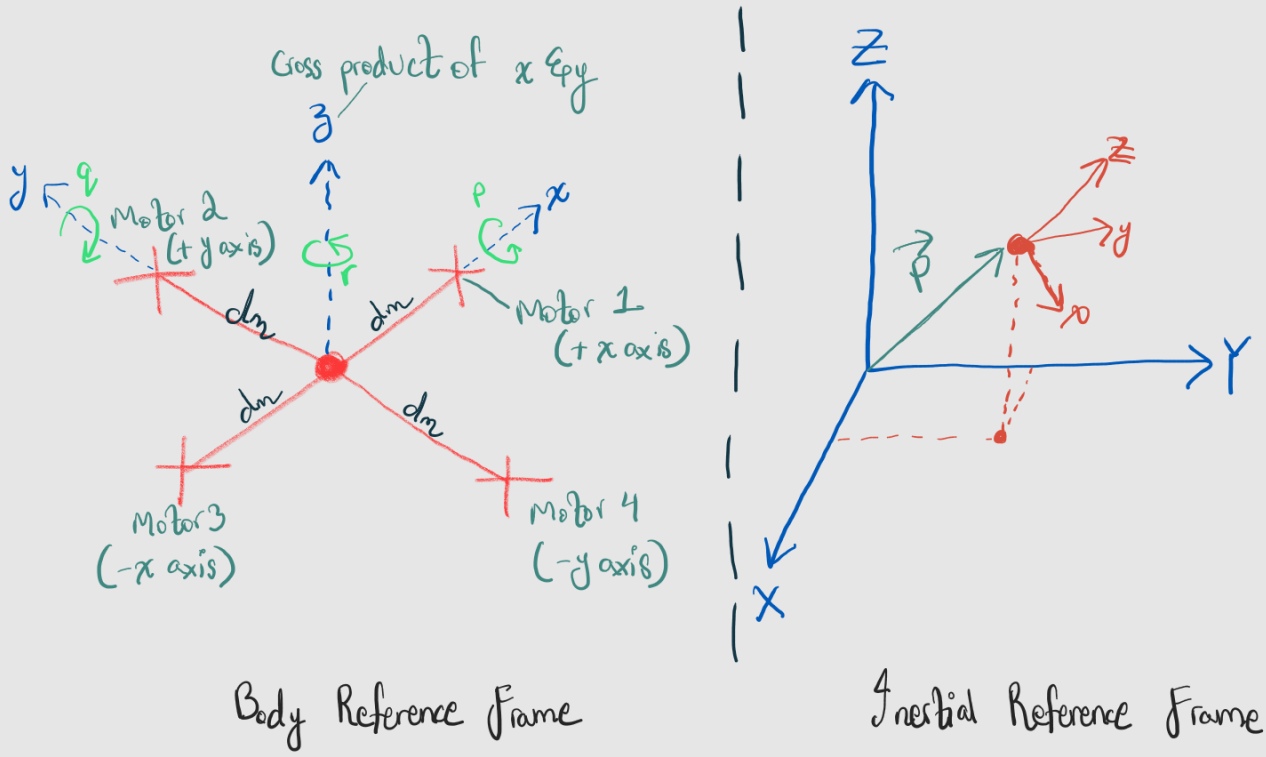
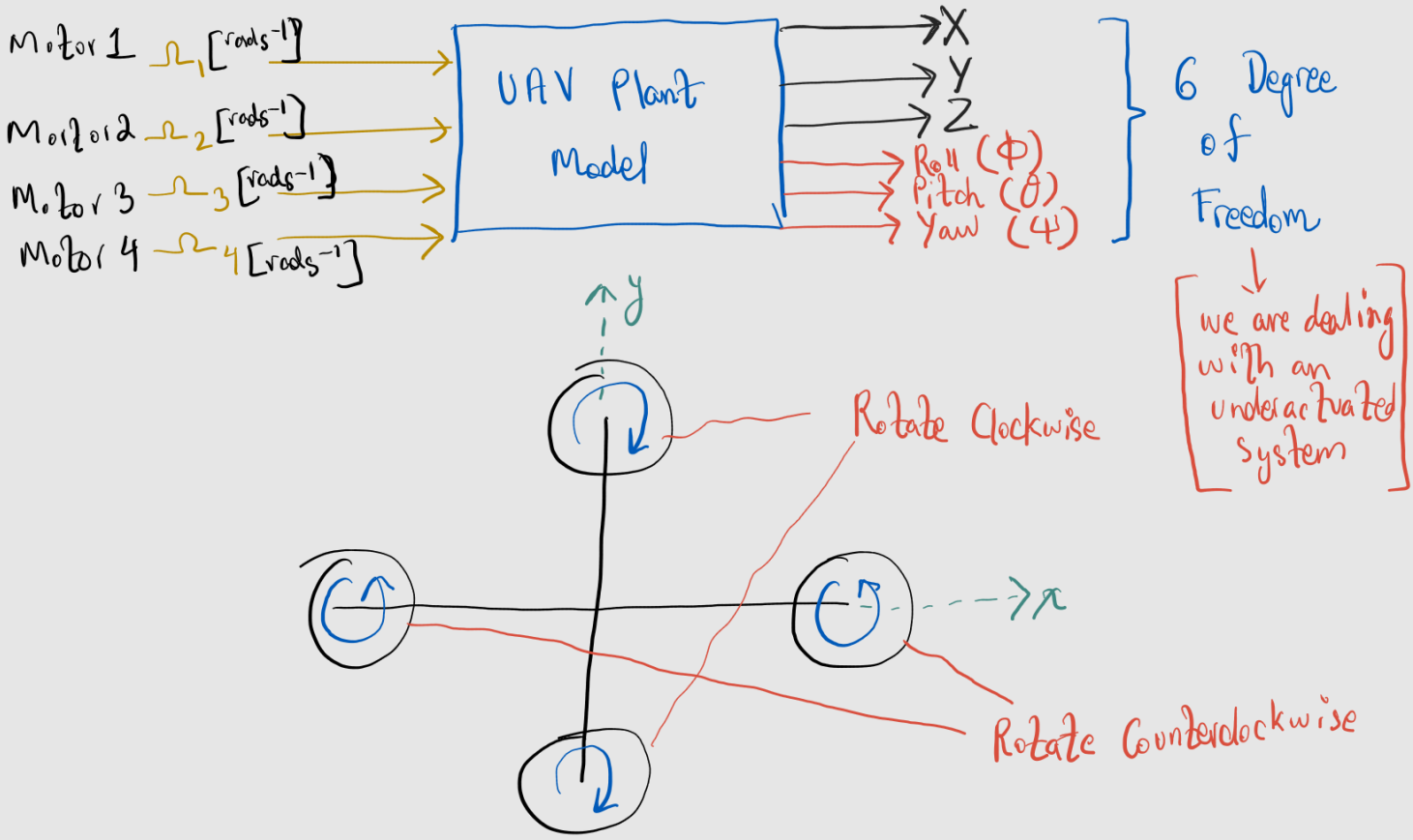
Drones Architecture from Control POV



Body Reference Frame

Inertial Reference Frame



Direction of Rotation of Motors

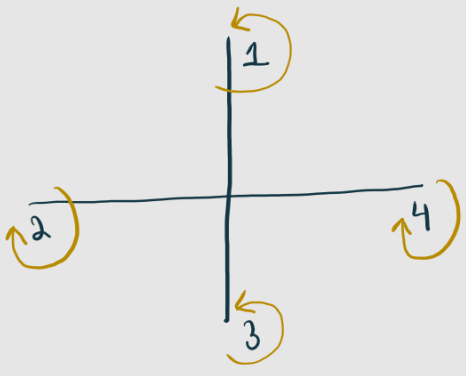
- The motor rotational directions depicted on the previous page are necessary to:

  (1) Maintain a zero net torque on the quadcopter
  (2) To allow to command independently the thrust, roll, pitch and yaw. (Atleast to a first degree analysis i.e no complex fluid dynamics)

Explaining Using Yaw as An Example

* Assume we want to yaw in ACW direction while maintaing altitude

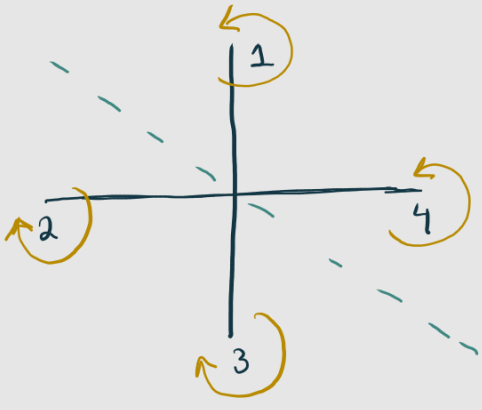
Option 01



- Speed up motor 1 & 3
- Slow down motor 2 & 4
- Thrust is maintained
- No force imbalance and thus no pitch or roll
- Yaw is independently commanded

✓

Option 02

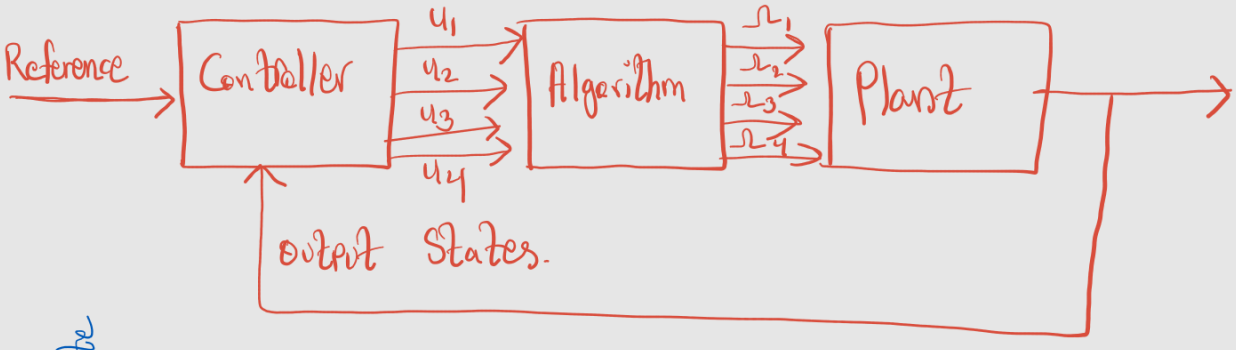


- Speed up motor 1 & 4
- Slow down motor 2 & 3
- Thrust is maintained
- Force imbalance causes rotation about the dotted line
- Drone pitches and rolls

✗

System Control Actions

- The plant as shown previously shown, needs to be fed with rotational speeds of all the 4 motors. However, it isn't convenient to directly generate them through a state space representation.
- Instead, we will do the following:



Each control input is meant to control its state INDEPENDETLY

$u_1 \longrightarrow$ Thrust Control Command

$u_1 - F_g = m\ddot{z}$ — only when z & Z are aligned. otherwise F_g will have components in other directions too

$u_2 \longrightarrow$ Roll Control Command

$u_2 = I_x \ddot{\phi}$

$u_3 \longrightarrow$ Pitch Control Command

$u_3 = I_y \ddot{\theta}$

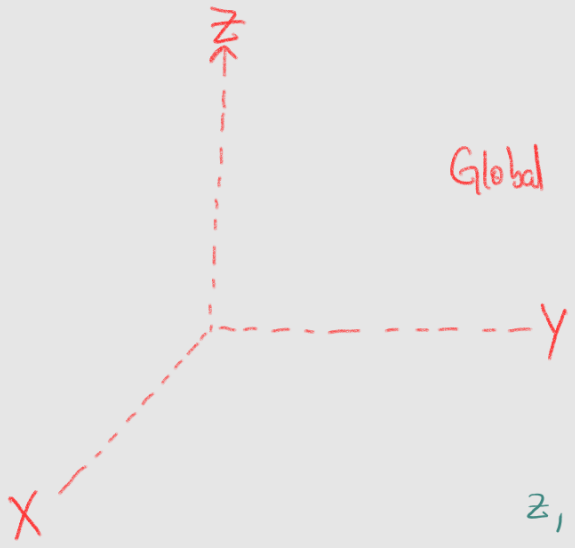
$u_4 \longrightarrow$ Yaw Control Command

$u_3 = I_z \ddot{\psi}$

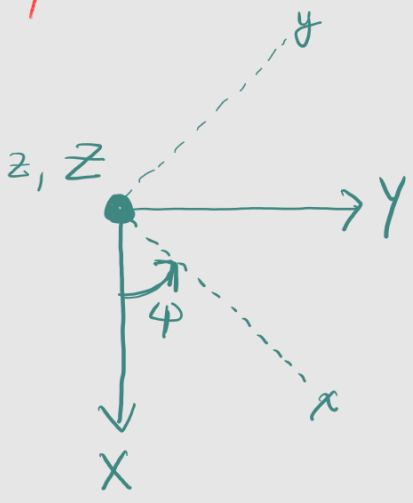
Fundamental Kinematics And Dynamics

3D Rotational Matrices

- Recall from 'Stability and Controls' course the concept of rotational matrices which can be leveraged to go from body frame to inertial frame or vice versa.



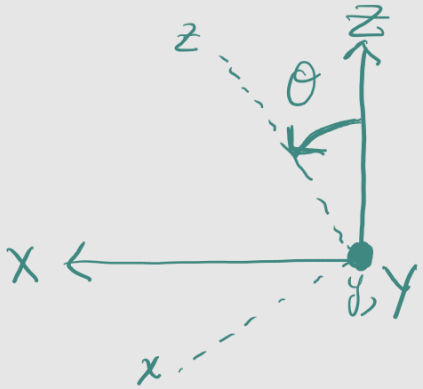
Global Reference Frame



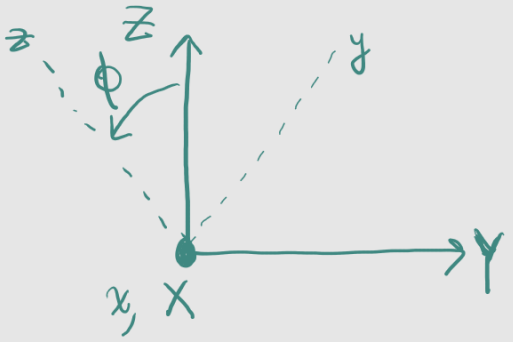
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R_z} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$R_z \rightarrow$ Rotational Matrix about Z axis

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}}_{R_y} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}}_{R_x} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



- R_x, R_y, R_z are all what are called **Orthonormal** Matrices. That is:

$$\begin{bmatrix} R_x^{-1} = R_x^T \\ R_y^{-1} = R_y^T \\ R_z^{-1} = R_z^T \end{bmatrix}$$

- Note that these rotational matrices can't be applied in any arbitrary order to go from body axis to inertial axis or vice versa. This can be mathematically proved through the multiplication property of matrices.

Example

$$\begin{bmatrix} R_x R_y R_z \neq R_x R_z R_y \neq R_y R_x R_z \neq R_y R_z R_x \\ \neq R_z R_x R_y \neq R_z R_y R_x \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_x R_y R_z \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Thus, when specifying rotation of a body frame it is necessary to adopt a convention (analogous to other used conventions like the Right Hand Rule).

- In total there are 12 unique conventions that may be adopted to go from one frame to another as per the Euler Rotation Theorem.

Rotation Matrix Conventions

24 Total

12 Fixed Frame Rotation (Fixed Angles Used)

↓

All rotations are performed about the Inertial Frame of reference

12 Rotating Frame Convention (Euler Angles Used)

↓

All rotations are performed using the body frame itself. As the body frame itself rotates with every applied rotation, we use the term "rotating about the moving reference frame".

- Mathematically & graphically investigating the two approaches above reveal that there is a special relationship b/w them:

  Rotating about three Inertial frame axis = Rotation about the same three body frame axis but in reverse

Example

$R_{xyz} = R_{zyx}$

First axis of rotation

Second axis of rotation

Third axis of rotation

$R_{zyx} = R_{xyz}$

Both approach use different set of angles

Euler Angles

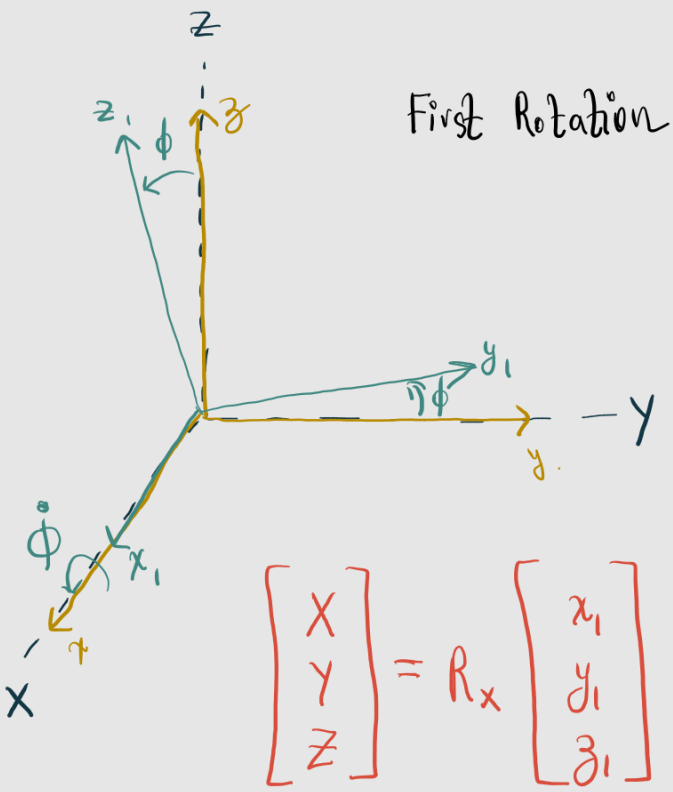
- According to Euler's rotation theorem, any rotation may be described using three rotations. Starting from a known reference orientation, any target orientation can be reached through the sequence of rotations.
- The three rotational angles are called the Euler angles.
- There are two approaches of defining the angles:

Extrinsic → Angles wrt the fixed inertial frame
Intrinsic → Angles wrt the rotating body frame

Extrinsic Rotation Derivation

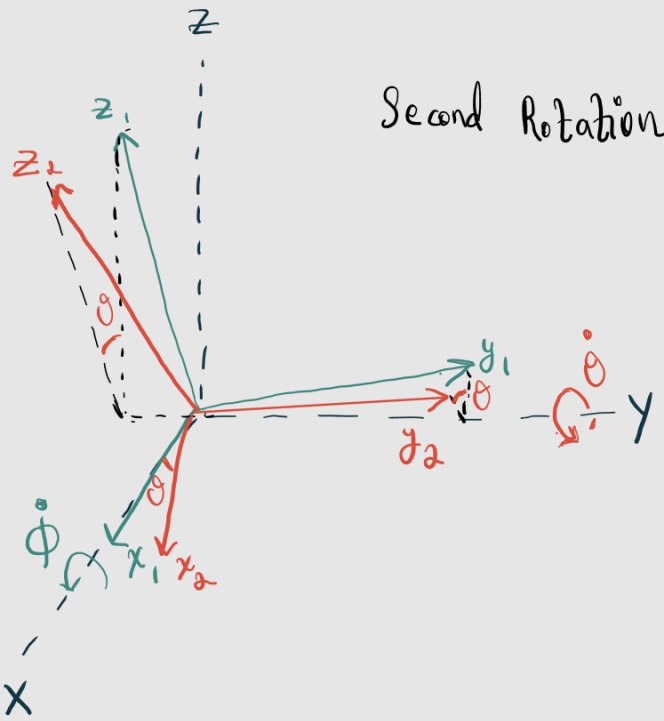
(Our convention R_{xyz})

↳ Inertial frame axis in rotational order

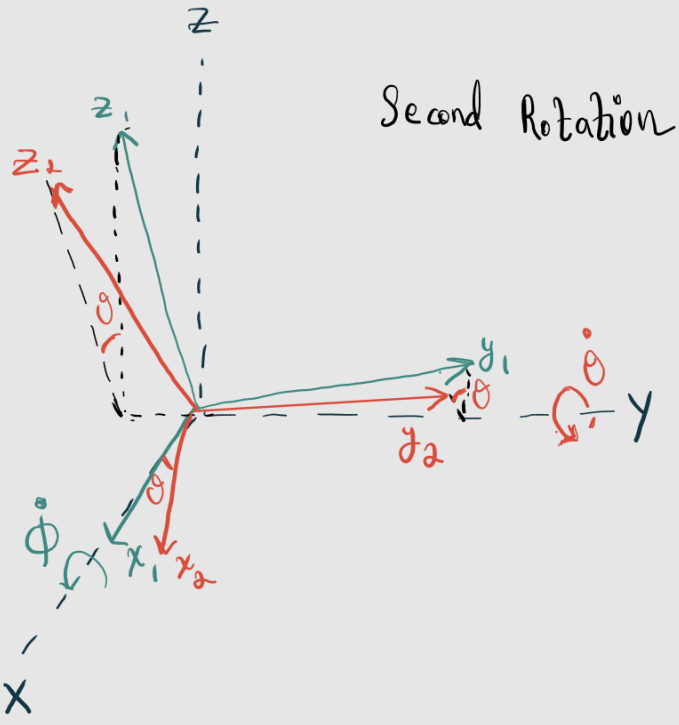


First Rotation

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_x \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$



Second Rotation

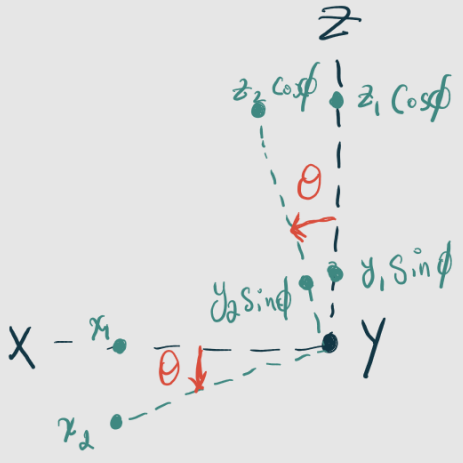


Second Rotation

- It may be difficult to imagine rotating the body frame about Y-axis as it is now at an offset. Imagine x_1, y_1 & z_1 to be individually linked with Y and then simulate them rotating in mind independantly from one another. y_1 & z_1 will create a rotational profile of a cone while x_1 will create a profile of a circle.

Note All vectors z_1, z_2, x_1, x_2, y_1 & y_2 have the same X-coordinate value. The rotation has strictly occured in the YZ plane.

Using diagram on right



Global Z

$$Z = y_1\sin\phi + z_1\cos\phi = y_2\sin\phi\cos\theta + z_2\cos\phi\cos\theta - x_2\sin\theta$$

Global X

$$X = x_1 = y_2\sin\phi\sin\theta + z_2\cos\phi\sin\theta + x_2\cos\theta$$

Global Y (Using first and second Rotation Diagram)

$$Y = y_1\cos\phi - z_1\sin\phi = y_2\cos\phi - z_2\sin\phi$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}}_{R_x \text{ as expected}} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}}_{\text{This product is actually } R_y R_x} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}}_{R_y} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}}_{R_x} = \begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

So we have,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_x \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = R_y R_x \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

What this means,

- The math reveals a very clever trick to get back to the original inertial axis system from $\langle x_2, y_2, z_2 \rangle$ system.
- We get to $\langle x_2, y_2, z_2 \rangle$ by first rotating ϕ rad about X axis and then θ rad about Y axis.

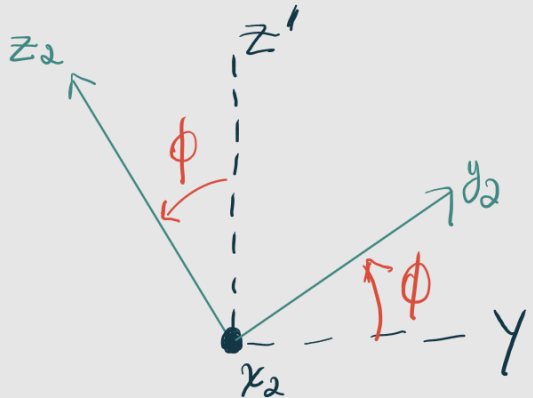
- To get back, the math says to spin back ϕ degrees about the x_2 axis (body frame axis). When this rotation completes it can be vizualized using the second rotation diagram that $\langle z_2 \rangle$ axis comes into the XZ plane and then $\langle y_2 \rangle$ exactly coincides with the Y axis.

- Then the math tells us to spin back θ degrees about the y_2 (moved body frame axis) or in this case coincidently the Y axis to coincide x_2 and z_2 with X and Z respectively as well.

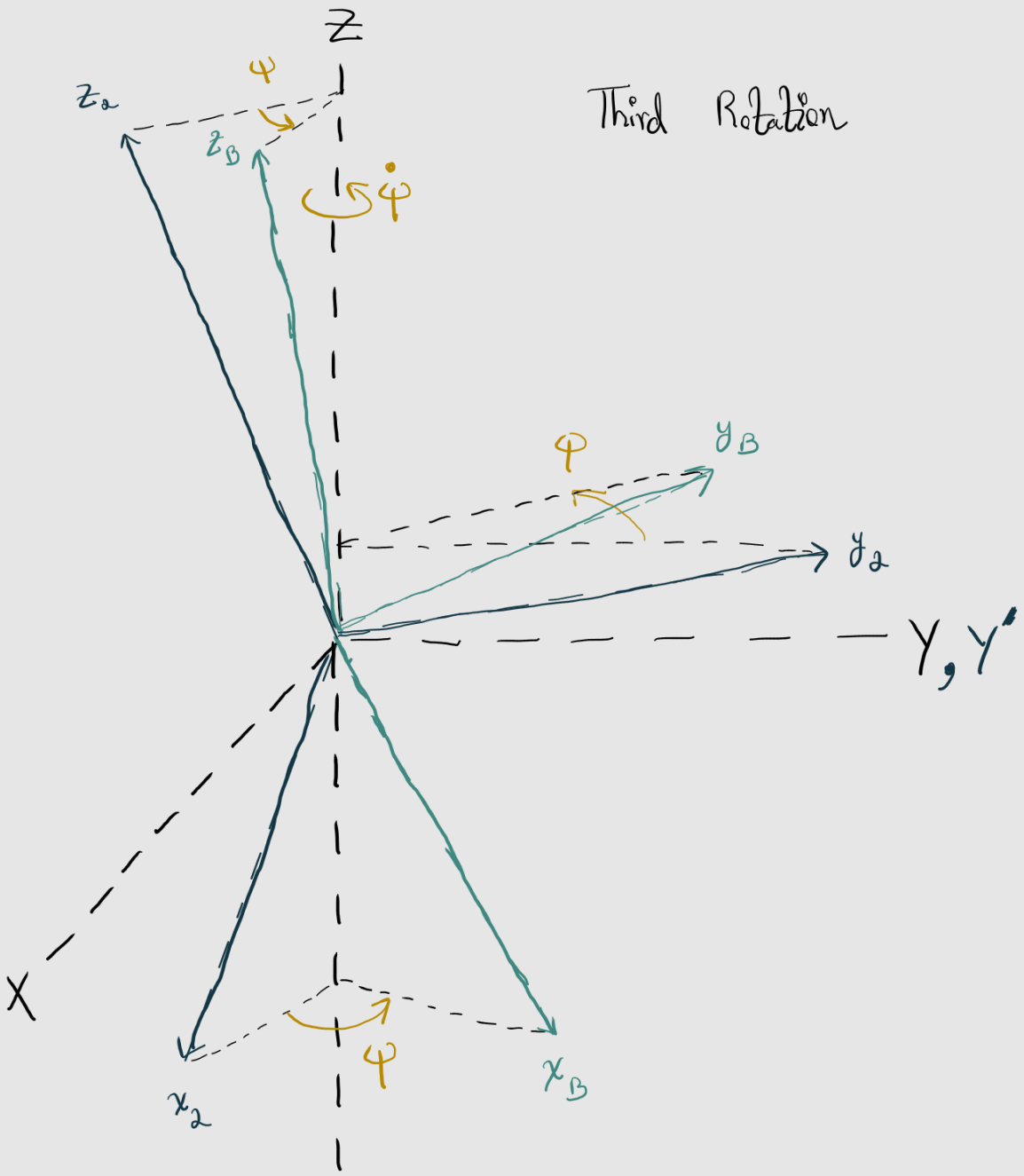
- Now although intuitively, we are rotating about the body axis, but the subscripts of the 2D rotational matrices are kept capitalized on last page because the angles within them are w.r.t the inertial frame. Honestly, the subscript don't mean much as long as we know what the angle of rotation is and what the axis of rotation is. This insignificance of subscripts is for 2D matrices ONLY.

Some More diagrams to help visualize the derivation

- Rotate Z axis θ degrees about Y axis and call the new axis Z'. We now have:



- With the same logic a final third rotation can be performed about the Z axis:



① $\cos(\underbrace{\angle z_2 z}_{\gamma}) = \dfrac{z_2 \cdot z}{|z_2||z|}$

$$= \frac{\langle 0,0,1\rangle \cdot \langle -\sin\theta, \cos\theta\sin\phi, \cos\theta\cos\phi\rangle}{\sqrt{\sin^2\theta + \cos^2\theta\sin^2\phi + \cos^2\theta\cos^2\phi}}$$

$$= \frac{\cos\theta\cos\phi}{\sqrt{\sin^2\theta + \cos^2\theta(\sin^2\phi + \cos^2\phi)}}$$

$$\cos\gamma = \cos\theta\cos\phi \Rightarrow \sin\gamma = \sqrt{1-\cos^2\theta\cos^2\phi}$$

② $\cos(\underbrace{\angle y_2 z}_{\beta}) = \dfrac{y_2 \cdot z}{|y_2||z|} = \dfrac{\langle 0,1,0\rangle \cdot \langle -\sin\theta, \cos\theta\sin\phi, \cos\theta\cos\phi\rangle}{\sqrt{\sin^2\theta + \cos^2\theta\sin^2\phi + \cos^2\theta\cos^2\phi}}$

$$\cos\beta = \cos\theta\sin\phi \Rightarrow \sin\beta = \sqrt{1-\cos^2\theta\sin^2\phi}$$

③ $\cos(\underbrace{\angle x_2 z}_{\alpha}) = \dfrac{x_2 \cdot z}{|x_2||z|} = \dfrac{\langle 1,0,0\rangle \cdot \langle -\sin\theta, \cos\theta\sin\phi, \cos\theta\cos\phi\rangle}{\sqrt{\sin^2\theta + \cos^2\theta\sin^2\phi + \cos^2\theta\cos^2\phi}}$

$$\cos\alpha = -\sin\theta \Rightarrow \sin\alpha = \cos\theta$$

Checking Results: $\cos^2\gamma + \cos^2\beta + \cos^2\alpha = 1$

$\cos^2\theta\cos^2\phi + \cos^2\theta\sin^2\phi + \sin^2\theta = 1$

$\cos^2\theta(\cos^2\phi + \sin^2\phi) + \sin^2\theta = 1$

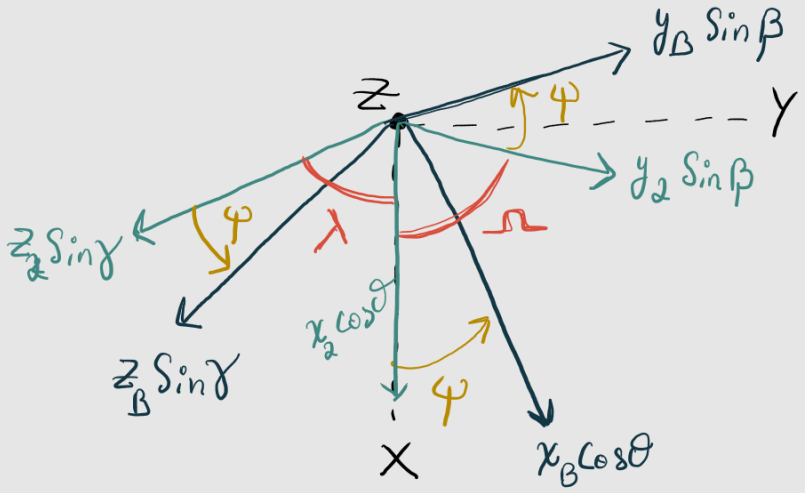
$1 = 1$

$\therefore$ Verified

$\gamma = \angle z_2 z = \angle z_B z$

$\beta = \angle y_2 z = \angle y_B z$

$\alpha = \angle x_2 z = \angle x_B z$



$$z_2 \sin\gamma \cos\lambda + y_2 \sin\beta \cos\Omega + x_2\cos\theta$$
$$= x_2\cos\theta + y_2 \sin\theta\sin\phi + z_2 \sin\theta\cos\phi$$

$$z_2\left(\sqrt{1-\cos^2\theta\cos^2\phi}\right)\cos\lambda + y_2\left(\sqrt{1-\cos^2\theta\sin^2\phi}\right)\cos\Omega$$
$$= y_2 \sin\theta\sin\phi + z_2 \sin\theta\cos\phi$$

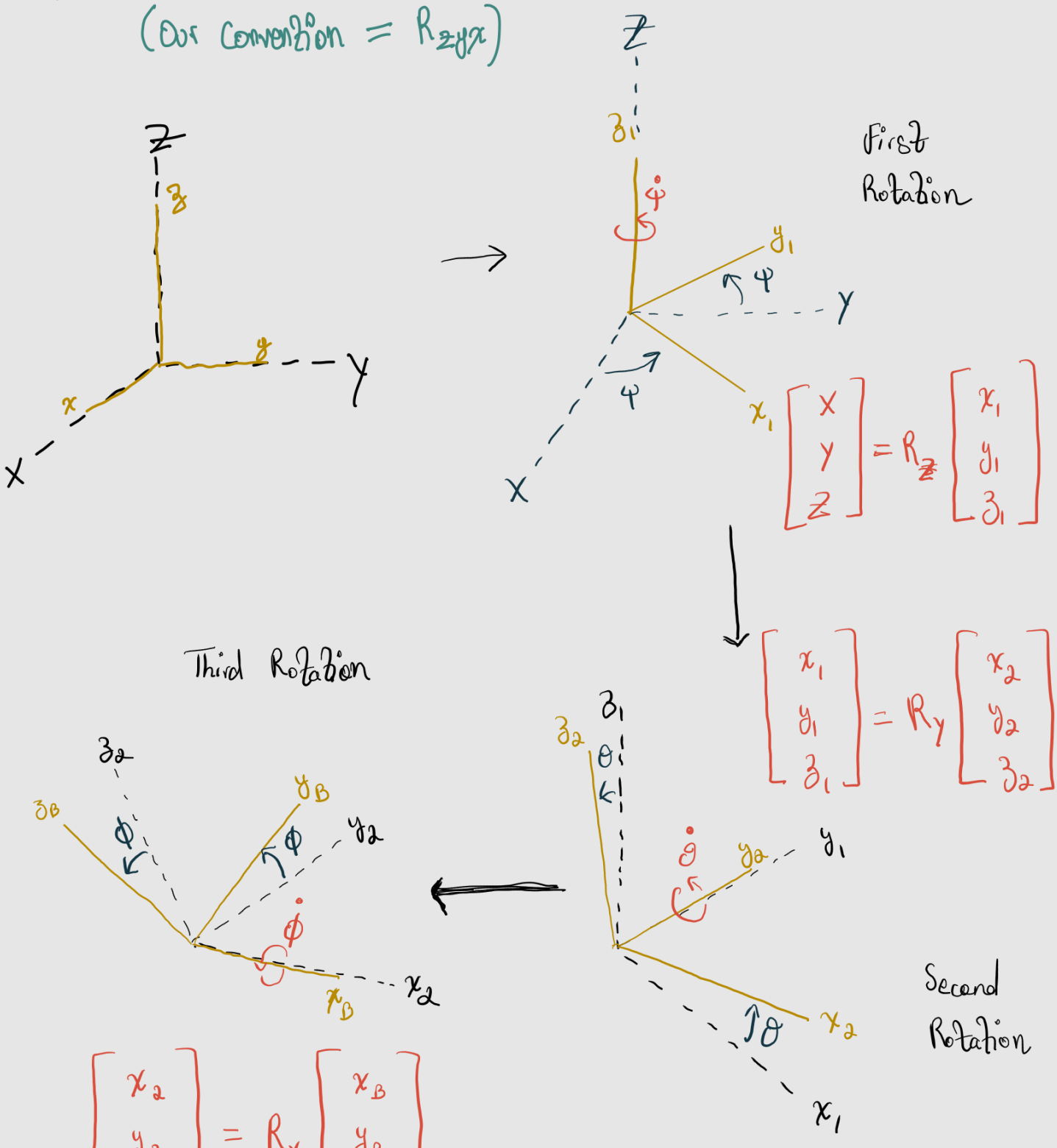
Equate z_2 and y_2 terms

$$\cos\Omega\sqrt{1-\cos^2\theta\sin^2\phi} = \sin\theta\sin\phi$$
$$\cos\Omega = \frac{\sin\theta\sin\phi}{\sqrt{1-\cos^2\theta\sin^2\phi}}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}}_{R_x \text{ as expected}} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}}_{\text{This product is actually } R_y R_x} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

Intrinsic Rotation Derivation

(Our convention = R_{zyx})



$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = R_z \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = R_y \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = R_x \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

Putting All Together:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{R_z R_y R_x} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}$$

Same result as that of Extrinsic approach proving the earlier claim: $R_{xyz} = R_{zyx}$

Important differences:

- The same intrinsic logic was also evident in the comments of the Extrinsic derivation results but note that the two approaches are very different from each other.

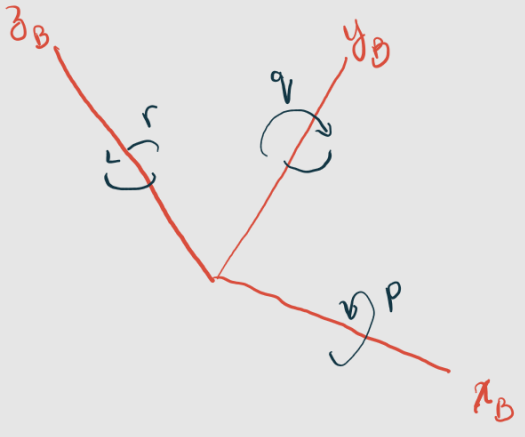
(1) Extrinsic Angles ≠ Intrinsic Angles

(2) $[\dot{\theta}, \dot{\phi}, \dot{\psi}]_{extrinsic}$ are directed along inertial frames whereas $[\dot{\theta}, \dot{\phi}, \dot{\psi}]$ are directed along moving body frames.

(3) Intrinsic approach has very simple rotational relations b/w the intermediate states, whereas, the intermediate rotations are quite difficult to visualize in the extrinsic case.

Transfer Matrix

For the intrinsic approach a special transfer matrix would be needed to connect body frame $<p, q, r>$ rotational velocities with $\dot{\theta}$, $\dot{\phi}$ and $\dot{\psi}$



$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

→ For Extrinsic case

$T = R_z R_y R_x$

- However, realize that $\dot{\phi}, \dot{\theta}$ & $\dot{\psi}$ aren't at 90° from another and so each rotation will have a combined relation with $<p,q,r>$.

- Now conceptually the rotational vector should have the same direction and angle in the inertial frame, no matter if p,q,r or $\dot{\theta}, \dot{\phi}, \dot{\psi}$ are treated as its individual components.

- Resolving p,q,r into non-orthogonal axis system that points in θ, ϕ and ψ directions is difficult so we adopt the opposite approach. We will resolve $\dot{\phi}, \dot{\theta}$ & $\dot{\psi}$ individually in the final orthogonal body frame and add all the components to get p,q & r.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = R_y R_x \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}, \quad \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = R_x \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}, \quad \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = R_y R_x \begin{bmatrix} p_1 \\ q_1 \\ r_1 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} = R_x \begin{bmatrix} 0 \\ q_2 \\ r_2 \end{bmatrix}, \quad \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p_3 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} p_1 + p_3 \\ q_1 + q_2 \\ r_1 + r_2 \end{bmatrix} = (R_y R_x)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R_x^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + R_x^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + (R_y R_x)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \underbrace{\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + R_x^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + (R_y R_x)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)}_{T^{-1}} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

- Using matrix inverse and orthonormal matrix property of rotational matrices we get T^{-1} upon computation as:

$$T^{-1} = \begin{bmatrix} 1 & 0 & -\sin\phi \\ 0 & \cos\phi & \sin\phi \cdot \cos\theta \\ 0 & -\sin\phi & \cos\phi \cdot \cos\theta \end{bmatrix}$$

Not orthonormal

↓ Taking Inverse

$$T = (T^{-1})^{-1} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \cdot \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi / \cos\theta & \cos\phi / \cos\theta \end{bmatrix}$$

Our Choice:

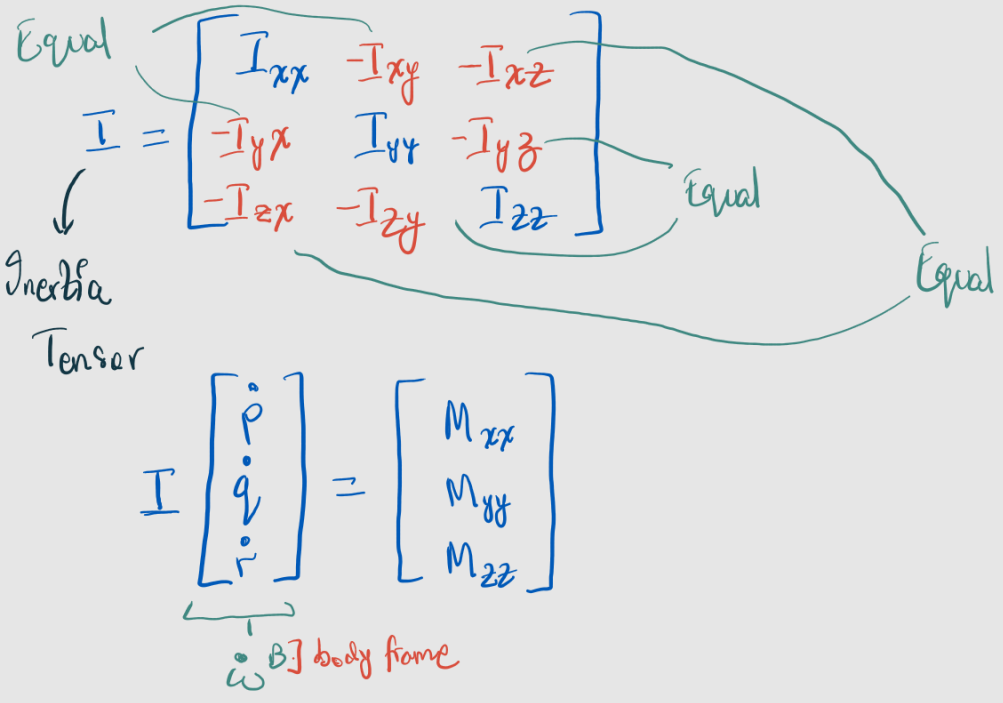
We will use the intrinsic convention for our controller development for the following reasons:

- Angles are easier to visualize
- Sensors provide angles much like in the intrinsic approach
- Transfer matrix doesn't need the yaw angle unlike in extrinsic
- Transfer matrix can be significantly simplified by taking assumptions. This will be seen later in derivations. The assumption we will take couldn't have been applied to the yaw angle Ψ.

Stating $R_z R_y R_x$ for reference

$$R_z R_y R_x = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix}$$

Mass Moment of Inertia & Inertia Tensor



- Products of inertia become zero if the chosen axis system is such that about each individual axis, the mass is equally distributed. (mass symmetric)
- Such an axis is called the principle axis.

$$I_{xx} = \int_m r_x^2 \, dm = \int_m (y^2 + z^2) \, dm$$
$$I_{yy} = \int_m r_y^2 \, dm = \int_m (x^2 + z^2) \, dm$$
$$I_{zz} = \int_m r_z^2 \, dm = \int_m (x^2 + y^2) \, dm$$

$$I_{xy} = I_{yx} = \int_m xy \, dm$$
$$I_{yz} = I_{zy} = \int_m yz \, dm$$
$$I_{xz} = I_{zx} = \int_m xz \, dm$$

= Make a CAD model for your drone and calculate its moment of inertia's. Try keeping drone during fabrication as mass symmetric other wise moment calculations will become very difficult.

Assumption

Assumption taken to decouple rotational motion along different axis

- Mass Symmetry exists about the taken body frame coordinate system:

$$\underline{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

Parallel Axis Theorem

$$\underline{I} = I_c + Mh^2$$

- $\underline{I}$: Moment of Inertia
- I_c: Moment of Inertia another axis
- M: Mass of body
- h: distance b/w the two axis

Dynamics

Translational Motion

- We will have velocity in body frame through IMU readings and we need to calculate the net force acting on the body as seen from the inertial frame.

- From dynamics course recall:

$$\vec{F}_{net} = m\vec{a}^{E}$$

$$\vec{F}_{net} = m\left(\dot{\vec{V}}^{B} + \vec{\omega}^{B} \times \vec{V}^{B}\right)$$

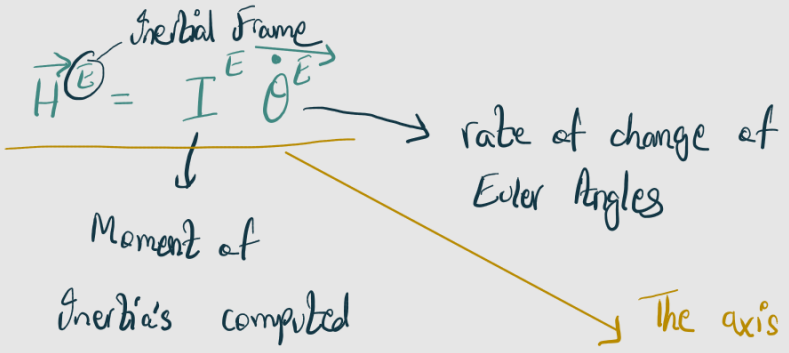
Choriolis Effect due to body frame's rotation.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \qquad \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Rotational Motion

- Similar to translational motion, a relation for moments will now be derived. Just like before we want the inertial moments to be in terms of body frame's dynamics as body frame information is what we will probably have to work with.

Angular Momentum = Moment of Inertia x Angular Velocity



$$\begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

May not necessarily be in the same direction due to inequivalence of I_{xx}, I_{yy}, I_{zz}

Angular velocity and angular acceleration can have different directions in two cases:

(1) Axis of rotation isn't symmetric

(2) Axis of rotation is moving

Takeaway: It is advised to have the angular momentum vector and angular velocity very close to one another to ensure a stable and predictable rotational motion of the system.

Read about this in greater detail and make notes

Moment Determination

$$\vec{M}_{net} = \dot{\vec{H}}^E = \dot{\vec{H}}^B$$

- We will use the body axis for the derivation because, representation of angular momentum in inertial frame will be very hard to derivate due to the following reasons.

Difficult & computationally intensive

- Whenever the body frame will rotate, mass distribution wrt X, Y, Z axis will change and I_{xx}, I_{yy} & I_{zz} will need to be recalculated.
- Whenever the body frame translates, parallel axis theorem will need to be applied to find the correct new Moment Tensor.

due Intrinsic adoption

- $\dot{\phi}$, $\dot{\theta}$ & $\dot{\psi}$ aren't along inertial X, Y, Z and so a relationship would be needed for necessary resolution along X, Y & Z axis.
- From practical point of view, the control forces and moments are often provided in terms of the body frame.
- Choosing body frame for derivations will thus ease things a lot.

$$\vec{H}^B = H_x \hat{i} + H_y \hat{j} + H_z \hat{k}$$
$$= I_{xx}^B p \hat{i} + I_{yy}^B q \hat{j} + I_{zz}^B r \hat{k}$$

body frame signs omitted for ease

$$\dot{\vec{H}}_B = \cancel{\dot{I}_{xx}^{\,0}\, p \hat{i}} + \cancel{\dot{I}_{yy}^{\,0}\, q \hat{j}} + \cancel{\dot{I}_{zz}^{\,0}\, r \hat{k}}$$
$$+ I_{xx} \dot{p} \hat{i} + I_{yy} \dot{q} \hat{j} + I_{zz} \dot{r} \hat{k}$$
$$+ I_{xx} p \underbrace{\dot{\hat{i}}}_{\vec{\omega}^B \times \hat{i}} + I_{yy} q \underbrace{\dot{\hat{j}}}_{\vec{\omega}^B \times \hat{j}} + I_{zz} r \underbrace{\dot{\hat{k}}}_{\vec{\omega}^B \times \hat{k}}$$
$$= I^B \dot{\vec{\omega}}^B$$

$$\vec{H}^B = \underline{I}^B \vec{\omega}^B + I_{xx}^B p(\vec{\omega}^B \times \hat{i}) + I_{yy}^B q(\vec{\omega}^B \times \hat{j}) + I_{zz}^B r(\vec{\omega}^B \times \hat{k})$$

$$\dot{\vec{H}}^B = \underline{I}^B \dot{\vec{\omega}}^B + \vec{\omega}^B \times I_{xx}^B p\hat{i} + \vec{\omega}^B \times I_{yy}^B q\hat{j} + \vec{\omega}^B \times I_{zz}^B r\hat{k}$$

Final complete vector change is represented using the body axis system

$$\left[\dot{\vec{H}}^B = \underbrace{\underline{I}^B \dot{\vec{\omega}}^B}_{\text{Change in Angular Momentum within the body frame alone}} + \underbrace{\vec{\omega}^B \times (\underline{I}^B \vec{\omega}^B)}_{\text{Chorriolis Effect to account for complete change in angular momentum wrt inertial frame}} \right]$$

Dynamics Takeaway (Newton Euler Formulation for 6DOF Rigid Body)

$$\vec{F}^B = m\left(\dot{\vec{V}}^B + \vec{\omega}^B \times \vec{V}^B\right)$$

$$\vec{M}^B = \underline{I}^B \dot{\vec{\omega}}^B + \vec{\omega}^B \times (\underline{I}^B \vec{\omega}^B)$$

$$\underbrace{\begin{bmatrix} m\underline{I}_{3\times3} & O_{3\times3} \\ O_{3\times3} & \underline{I}^B \end{bmatrix}}_{M^B} \underbrace{\begin{bmatrix} \dot{\vec{V}}^B \\ \dot{\vec{\omega}}^B \end{bmatrix}}_{\dot{\vec{V}}^B} + \underbrace{\begin{bmatrix} \vec{\omega}^B \times (m\vec{V}^B) \\ \vec{\omega}^B \times (\underline{I}\,\vec{\omega}^B) \end{bmatrix}}_{C^B(\vec{V}_B)} = \underbrace{\begin{bmatrix} \vec{F}^B \\ \vec{M}^B \end{bmatrix}}_{\vec{\Lambda}^B}$$

$$\begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix}$$

M^B · $\dot{\vec{V}}^B$ · $C^B(\vec{V}_B)$ · $\vec{V}_B$ · $\vec{\Lambda}^B$

C^B as a function of $\vec{V}_B$

Newton Euler Formulation for a 6 DOF Rigid Body :

$$M^B \dot{\vec{v}}^B + C^B(\vec{v}^B)\vec{v}^B = \vec{\Lambda}^B$$

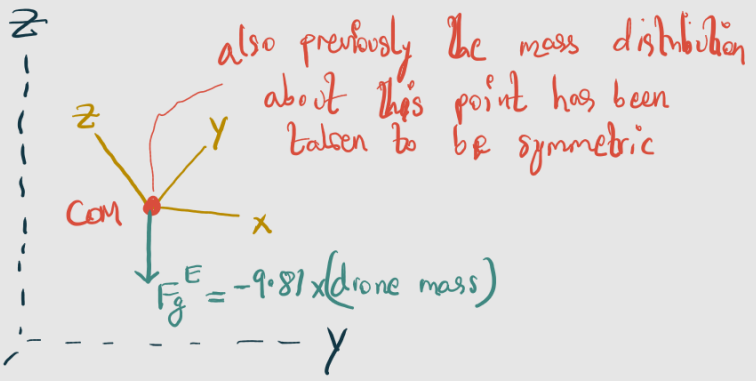
- Note that the derived kinematics and dynamics equations are general in nature for a 6-DOF rigid system and not specific to a drone in nature.
- The relation basically tells us how forces and moments applied to a body connects it to its states and their derivatives.
- Often the forces applied to a system are represented in the body frame. Thus, the body frame convention we used will aid us greatly to adapt this formulation to our quadcopter.
- One assumption taken till this point is of mass symmetry leading to zero products of inertia.

UAV Specific Plant Model

1) Gravitational Force:

- Our body frame passes through the center of gravity so that the gravitational force has no moment contribution.

- Rigid body assumption allows us to treat the gravitational force as a point force. Non-rigid body would want us to take force distribution in account to evaluate structural deformations.



$$\vec{F}_g^{\,E} = R_{zyx}\,\vec{F}_g^{\,B}$$

$$\vec{F}_g^{\,B} = (R_{zyx})^{-1}\begin{bmatrix}0\\0\\-mg\end{bmatrix}$$

$$\vec{\Lambda}_G^B = \begin{bmatrix} \overbrace{(R_{zyx})^{-1}}^{R_{zyx}^T} \begin{bmatrix}0\\0\\-mg\end{bmatrix} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

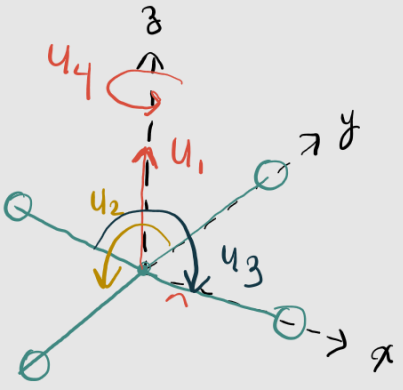
$$\vec{\Lambda}^{B}_{G} = \begin{bmatrix} mg\sin\theta \\ -mg\sin\phi\cos\theta \\ -mg\cos\phi\cos\theta \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

≡ Upon calculation this is the final form that we get

2) Control Inputs

$$\vec{\Lambda}^{B}_{in} = \begin{bmatrix} 0 \\ 0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

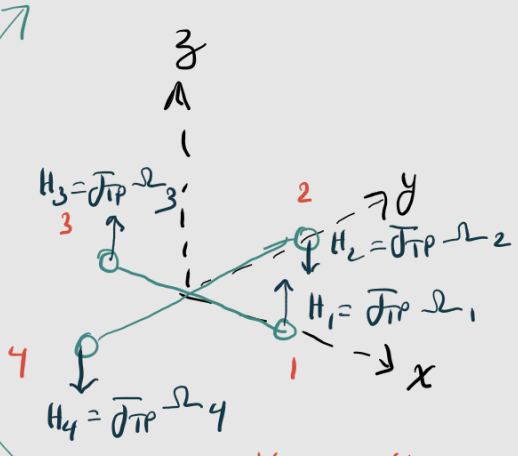
u_1 → Thrust Command
u_2 → Roll Command
u_3 → Pitch Command
u_4 → Yaw Command



3) Gyroscopic Effect

- Our rotors are the way through which we can exert moment on our system. This is when we assume that there exist no external disturbances.

Moment of inertia of propeller about it's axis of rotation

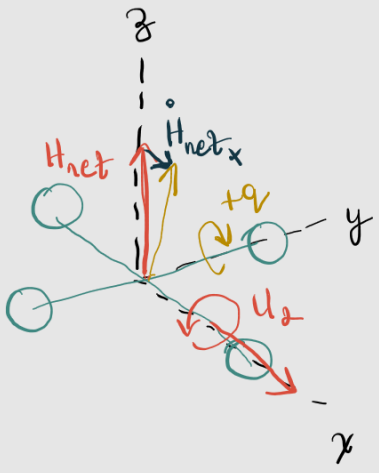


$$H_{net} = H_1 + H_2 + H_3 + H_4$$
$$= J_{TP}(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)$$

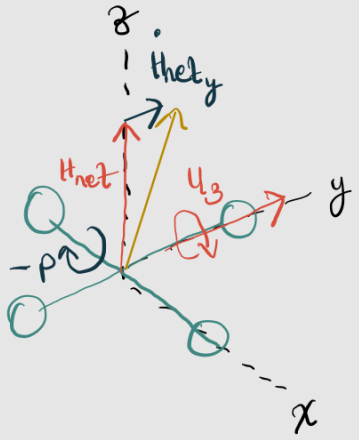
- When $\Omega_1 + \Omega_3 = \Omega_2 + \Omega_4$, the system will have zero angular momentum and any moment accompanying this scenario will lead to rotations as we may have predicted.

- However, whenever a moment is applied to the system and H_{net} is not zero then the angular momentum vector will chase the applied moment vector.

Roll Command



Pitch Command



$$\dot{H}_{net_x} = \vec{q} \times \vec{H}_{net} \qquad \dot{H}_{net_y} = -\vec{p} \times \vec{H}_{net}$$

≡ The $\dot{H}_{net}$ vector forces the whole drone to reorient and keep it body 'z' axis parallel with the new yellow colored H_{net}.

$$\dot{\vec{H}}_{net} = \vec{\omega}^{B} \times \vec{H}^{B}_{net}$$

$$= \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ p & q & r \\ 0 & 0 & H_{net} \end{vmatrix}$$

$$= \hat{i}\,(q H_{net}) - \hat{j}\,(p H_{net})$$

We are analyzing the angular momentum imparted due to rotor rotation. Obviously there will be momentum due to full drone rotation but that is left out due to reasons that will be explained later.

Let: $\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$

$$\dot{\vec{H}}_{net} = \begin{bmatrix} q J_{TP} \Omega \\ -p J_{TP} \Omega \\ 0 \end{bmatrix}$$

+p gives -ve $\dot{H}_{net_y}$ which is in agreement with our pitch command graphical investigation.

- Take a look at the formula of $\vec{H}_{net}$:

$$\vec{H}_{net} = \vec{\omega}^B \times \vec{H}^B_{net}$$

- Ideally $\vec{H}^B_{net}$ should have been replaced by the complete angular momentum of the whole drone $\vec{H}^B$ and not of the propellers alone.
- However, we can't do that because we don't know how at a given time the general $\vec{H}_{net}$ will affect the angular velocities. The rotor $\vec{H}^B_{net}$ we know always acts in the '3' direction so it possible to reliably work with it as mathematically done on the last page.

- Also, this is the reason why it recommended to have $\vec{\omega}^B$ and $\vec{H}^B$ always in the same direction so that cross product resulting from them becomes zero.

- Anyways, this $\vec{H}_{net}$ moment vector is a consequence of application of control inputs so we alter the Newton-Euler formulation as:

$$M^B \dot{\vec{v}}^B + C^B(\vec{v}^B)\vec{v}^B + \vec{\omega}^B \times \vec{H}^B_{net} = \vec{\Lambda}^B_G + \vec{\Lambda}^B_{in}$$

$$M^B \dot{\vec{v}}^B + C^B(\vec{v}^B)\vec{v}^B = \vec{\Lambda}^B_G + \vec{\Lambda}^B_{in} + \underbrace{\left(-\vec{\omega}^B \times \vec{H}^B_{net}\right)}_{\vec{\Lambda}^B_{GR}}$$

$$\vec{\Lambda}^B_{GR} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -q J_{TP} \Omega \\ p J_{TP} \Omega \\ 0 \end{bmatrix}$$

Creating The State Space Form

$$M^B \dot{\vec{v}}^B + C^B \vec{v}^B = \vec{\Lambda}^B$$

$$\dot{\vec{v}}^B = \underbrace{(M^B)^{-1}}\left(-C^B \vec{v}^B + \vec{\Lambda}^B\right)$$

↓

For a diagonal matrix with no zero as a diagonal element, the matrix inverse will be simply the reciprocal of all the diagonal entries.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 1/m & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_{zz} \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & 0 & 0 & -mw & mv \\ 0 & 0 & 0 & mw & 0 & -mu \\ 0 & 0 & 0 & -mv & mu & 0 \\ 0 & 0 & 0 & 0 & -I_{zz}r & I_{yy}q \\ 0 & 0 & 0 & I_{zz}r & 0 & -I_{xx}p \\ 0 & 0 & 0 & -I_{yy}q & I_{xx}p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} + \vec{\Lambda}^B \right)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 1/m & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_{zz} \end{bmatrix} \left(\begin{bmatrix} -mwq + mvr \\ mwp - mur \\ -mvp + muq \\ -I_{zz}rq + I_{yy}qr \\ I_{zz}rp - I_{xx}pr \\ -I_{yy}qp + I_{xx}pq \end{bmatrix} + \begin{bmatrix} mg\sin\theta \\ -mg\sin\phi\cos\theta \\ -mg\cos\phi\cos\theta + U_1 \\ U_2 - qJ_{TP}\Omega \\ U_3 + pJ_{TP}\Omega \\ U_4 \end{bmatrix} \right)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 1/m & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_{zz} \end{bmatrix} \begin{bmatrix} -mwq + mvr + mg\sin\theta \\ mwp - mur - mg\sin\phi\cos\theta \\ -mvp + muq - mg\cos\phi\cos\theta + U_1 \\ -I_{zz}rq + I_{yy}qr + U_2 - qJ_{TP}\Omega \\ I_{zz}rp - I_{xx}pr + U_3 + pJ_{TP}\Omega \\ -I_{yy}qp + I_{xx}pq + U_4 \end{bmatrix}$$

State Space in Body Frame

$$\begin{aligned} \dot{u} &= vr - wq + g\sin\theta \\ \dot{v} &= wp - ur - g\sin\phi\cos\theta \\ \dot{w} &= uq - vp - g\cos\phi\cos\theta + U_1/m \end{aligned}$$

Much more difficult to Linearize

$$\begin{aligned} \dot{p} &= \frac{(I_{yy} - I_{zz})}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\ \dot{q} &= \frac{(I_{zz} - I_{xx})}{I_{yy}} pr + \frac{J_{TP}}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \\ \dot{r} &= \frac{(I_{xx} - I_{yy})}{I_{zz}} pq + \frac{U_4}{I_{zz}} \end{aligned}$$

Relatively easier to treat as linear through techniques

- State Space is a very convenient form as finding state derivatives allows us to easily approximate future states:

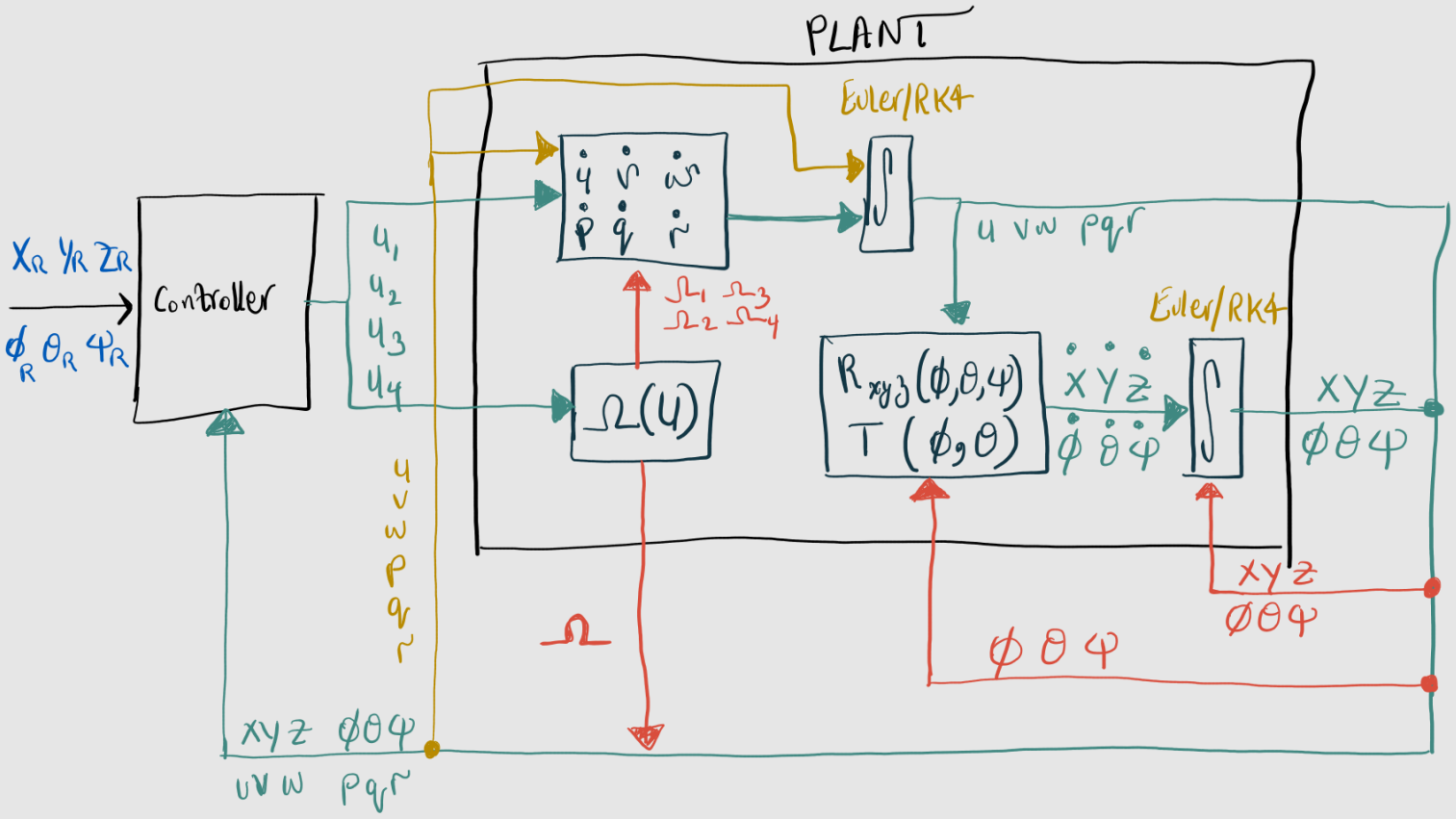
Euler Integration Method

$$State_{k+1} = State_k + \dot{State}_k \cdot \frac{T_s}{N}$$

T_s: Time Interval of one iteration

N: Dividing Time Interval into smaller stamps to reduce prediction error.

↳ Simple but imprecise.
RK4 is actually used in the course but for ease of presentation show the Euler Integration Method

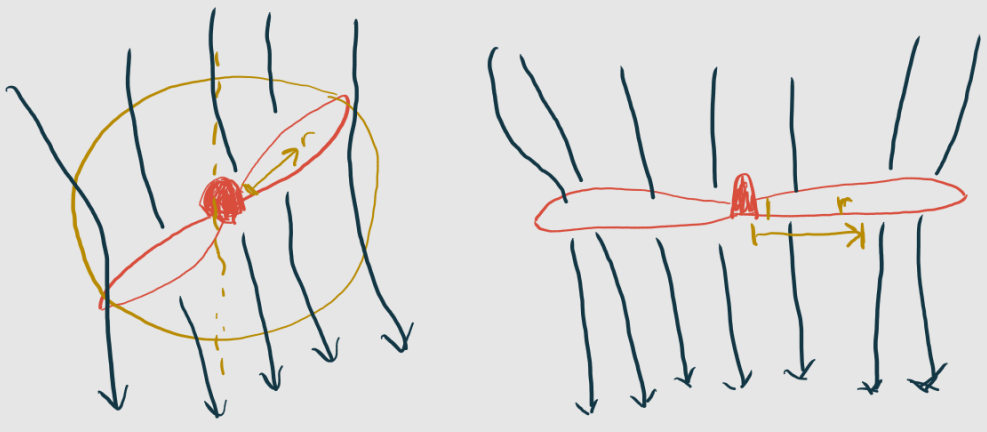


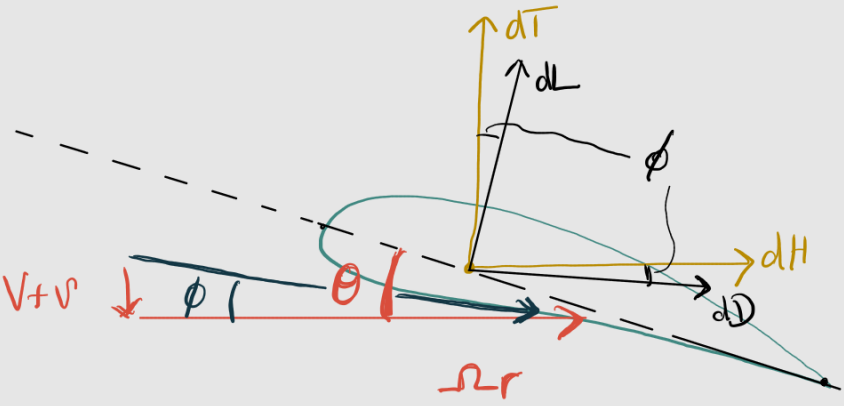
- After presentation, consult "numerical methods" course and make notes on the RK4 method.

Blade Element Momentum Theory

- We use this to go from control inputs towards the angular velocities of the rotor blades.

- Rotation of the propellers will create a stream tube through the rotor disc that will contribute a downwash component of airflow across rotor blades cross-sections.





For U_1, U_2, U_3

$$dT = dL\cos\phi - dD\sin\phi$$

$$dT = \frac{1}{2}\rho V_{total}^2\, dA\,(C_l\cos\phi - C_d\sin\phi)$$

$$dT = \frac{1}{2}\rho\underbrace{(\Omega^2 r^2 + (V+v)^2)}\, c\,(C_l\cos\phi - C_d\sin\phi)\,dr$$

* $\Omega r \gg V+v$
* $(\Omega r)^2 + (V+v)^2 \approx \Omega^2 r^2$
* $\Omega r \gg V+v \rightarrow \phi$ is small
* $\cos\phi \approx 1$, $\sin\phi \approx \phi$

$$dT = \frac{1}{2}\rho\,\Omega^2 r^2 c\,(C_l - C_d\phi)\,dr$$

$$\text{Single Motor Thrust} = T = N_b\int_0^R \frac{1}{2}\rho\,\Omega^2 r^2 c\,(C_l - C_d\phi)\,dr$$

$$T = \Omega^2\left(\frac{N_b\rho}{2}\right)\underbrace{\int_0^R c\,(C_l - C_d\phi)\,r^2\,dr}_{C_T}$$

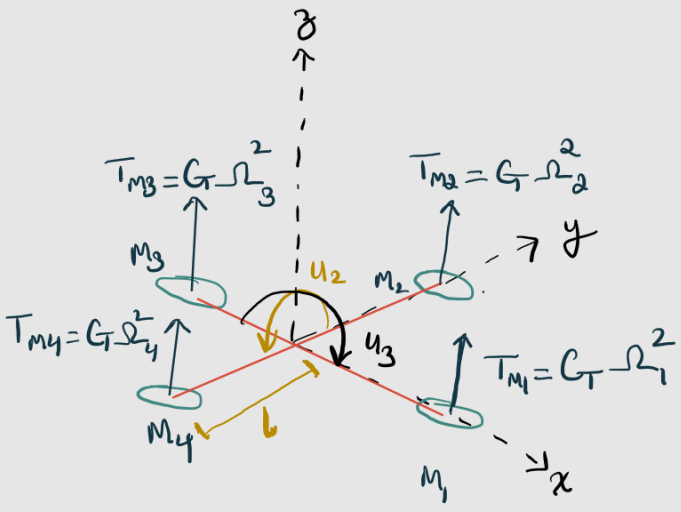
$$\boxed{T = C_T\,\Omega^2}$$ ∴ Thrust is directly proportional to Ω^2

- For FYP calculate the value of C_T either analytically or find it through existing experiment data/blade specifications sheet.

$$U_1 = T_{M1} + T_{M2} + T_{M3} + T_{M4}$$

$$\rightarrow U_1 = C_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right)$$

↳ Thrust Command



$$U_2 = T_{M2}\,l - T_{M4}\,l$$

$$\rightarrow U_2 = C_T l\left(\Omega_2^2 - \Omega_4^2\right)$$

↳ Roll Command

* Parameter 'l' needs to be determined for the quadcopter

$$U_3 = T_{M3}\,l - T_{M1}\,l$$

$$\rightarrow U_3 = C_T l\left(\Omega_3^2 - \Omega_1^2\right)$$

For U_4

* Assuming that $\dot{\Omega}$ for every motor is zero, then the moment that the air exerts on the rotor blades will be equal to the moment that the motor exerts on the rotor blades.

$$dQ = r\,dH$$

$$dQ = r\left(dL\sin\phi + dD\cos\phi\right)$$

$$dQ = r\left(\frac{1}{2}\rho V_{Total}^2\,dA\left(C_l\sin\phi + C_d\cos\phi\right)\right)$$

Taking same prior assumptions:

$$dQ = r\left(\frac{1}{2}\rho\Omega^2 r^2 c\left(C_l\phi + C_d\right)\right)dr$$

$$dQ = \frac{r^3 \rho}{2} \Omega^2 c(C_l \phi + C_d) dr$$

$$Q = \Omega^2 \frac{N_b \rho}{2} \int_0^R c(C_l \phi + C_d) r^3 dr$$

Negative assumed here on purpose $\ominus C_Q$

* Just C_T, compute C_Q for the quadcopter as well.

$$Q = -C_Q \Omega^2$$

↳ Moment exerted by air onto the propeller

Moment exerted by drone/motor onto the propeller $M_D = +C_Q \Omega^2$ (For a CCW rotating propeller)

$$C_Q = -\frac{N_b \rho}{2} \int_0^R c(C_l \phi + C_d) r^3 dr$$

∴ Expression for Moment Coefficient

$$M_{D_{total}} = M_{D_1} \ominus M_{D_2} + M_{D_3} - M_{D_4}$$

↳ Clockwise propellers will have opposite moment direction as computed through CCW blade derivation.

$$M_{D_{total}} = C_Q (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$$

- For yaw command, one rotor pair will rotate faster than the other pair. In doing so the drone will apply a net torque to the propellers and as a reaction an opposite but equal torque will be experienced by the drone causing it to Yaw. This reaction is our Yaw Command U_4.

$$\rightarrow U_4 = C_Q\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right)$$

Final Representation

$$U_1 = C_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right)$$
$$U_2 = C_T l\left(0 + \Omega_2^2 + 0 - \Omega_4^2\right)$$
$$U_3 = C_T l\left(-\Omega_1^2 + 0 + \Omega_3^2 + 0\right)$$
$$U_4 = C_Q\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right)$$

Assumption

$$\dot{\Omega}_1 = \dot{\Omega}_2 = \dot{\Omega}_3 = \dot{\Omega}_4 = 0$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} U_1/C_T \\ U_2/C_T l \\ U_3/C_T l \\ U_4/C_Q \end{bmatrix}$$

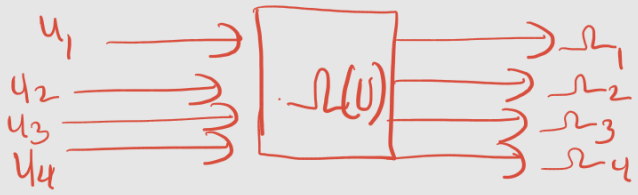
- Taking inverse of first matrix and multiplying on both sides:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} 1/4 & 0 & -1/2 & -1/4 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 & -1/4 \\ 1/4 & -1/2 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} U_1/C_T \\ U_2/C_T l \\ U_3/C_T l \\ U_4/C_Q \end{bmatrix}$$

- Use above formulation to find $\Omega_1, \Omega_2, \Omega_3, \Omega_4$
- Omega squares when computed may come out as negative. This can happen if the sample time interval is either too large or too small and if this happens then the simulation will fail. Extreme trajectories with very sharp turns can lead to this issue.

Recall: $\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$

12 We have now successfully completed the box below:



Assumptions Taken in the Plant and Room For Improvements For Agile Implementation

1) Quadcopter is assumed to be mass symmetric
2) Body Frame axis passes through COM
3) No winds
   - An OK assumption for indoor flight in controlled atmosphere. Controller performance can significantly change, as seen in a UTIt Zurich lab test, when wind is applied to physical quad with winds not accounted for in its controller design.
4) No aerodynamic drag forces on frame
   - For Agile and high speed applications this assumption is no valid. Drag forces aren't negligible.
5) Gyroscopic precession applied to propeller angular momentum only
   - See if this can be scaled to all influences including phenomenon that are yet to be modelled.
6) BMT assumption that down wash velocity is very small
   - Apply advanced rotor concepts, especially those with forward flight at high speeds for a greater fidelity.
7) $\dot{\Omega}_1 = \dot{\Omega}_2 = \dot{\Omega}_3 = \dot{\Omega}_4 = 0$ for C_Q calculation.
8) Rigid body Assumption
   - At high g's, the frame will experience deformations and EOMs would change. Try to capture this effect.

9) No complex fluid characteristics modelled.

- This is a necessary simplification due to the unpredictable nature of turbulance around quadcopter's rotors.