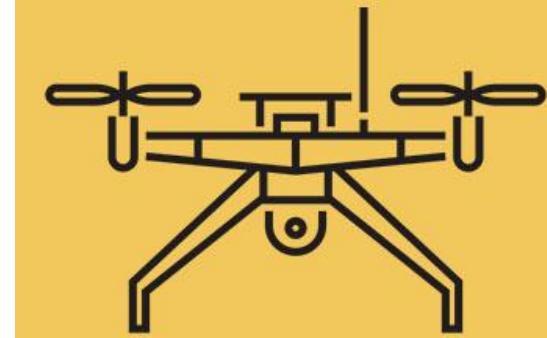
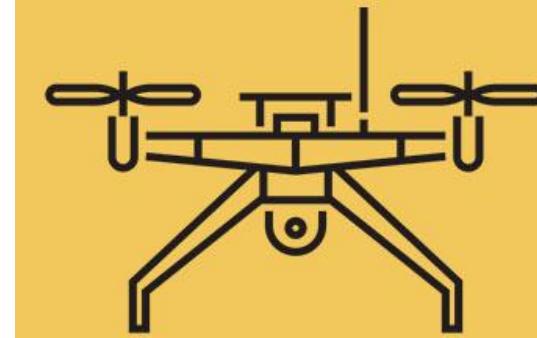


Today's lecture

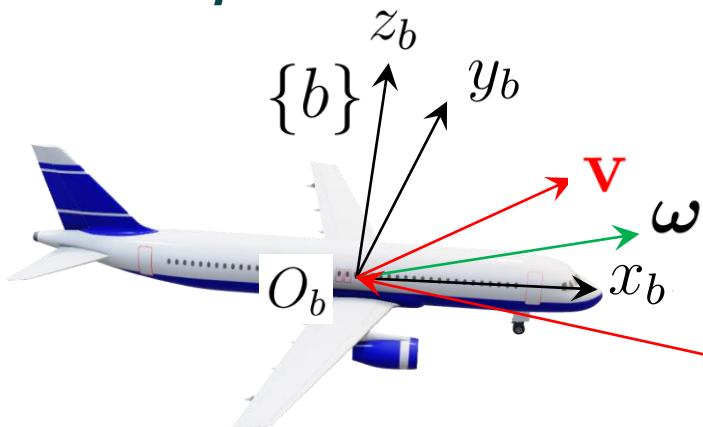
- **Basics of kinematics**
- **Notations for frames and vectors**
- **Rotations matrices and Euler angles**
- **Transformation of angular velocities**



Reference frames and notations



Body-fixed frame

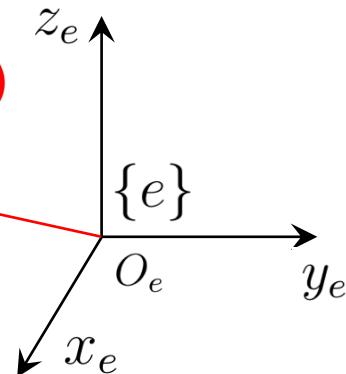


\mathbf{p} (coordinate-free vector)

$\mathbf{p}_{b/e}^e$: Position of point O_b with respect to frame $\{e\}$ expressed in frame $\{e\}$
(coordinate vector)

Note that $\mathbf{p}_{b/b}^e = \mathbf{p}_{b/b}^b = 0$

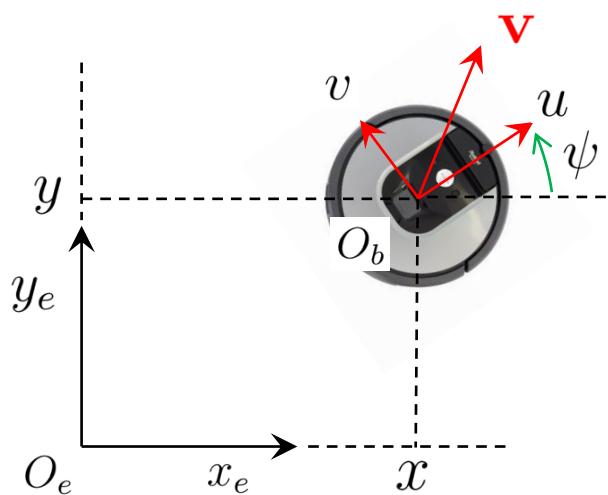
Earth-fixed frame



$\mathbf{v}_{b/e}^b$: Linear velocity of point O_b with respect to frame $\{e\}$ expressed in frame $\{b\}$

$\omega_{b/e}^b$: Angular velocity of frame $\{b\}$ with respect to frame $\{e\}$ expressed in frame $\{b\}$

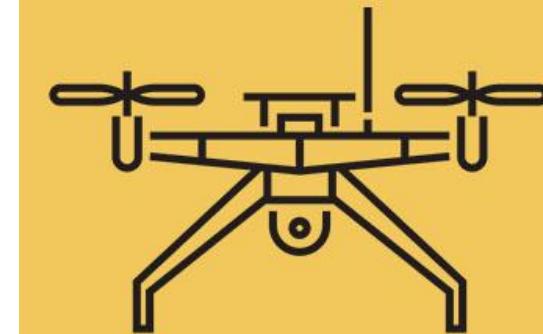
Rotation matrix in 2D



where

$$\mathbf{p}_{b/e}^e = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\rightarrow \frac{d}{dt} \mathbf{p}_{b/e}^e = \mathbf{v}_{b/e}^e = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$



In the body-fixed frame:

$$\mathbf{v}_{b/e}^b = \begin{bmatrix} u \\ v \end{bmatrix}$$

Simple geometry gives:

$$\begin{cases} \dot{x} = u \cos(\psi) + v \cos(\psi + \frac{\pi}{2}) \\ \dot{y} = u \sin(\psi) + v \sin(\psi + \frac{\pi}{2}) \end{cases}$$

How are $\mathbf{v}_{b/e}^e$ and $\mathbf{v}_{b/e}^b$ related?

$$\begin{cases} \dot{x} = u \cos \psi - v \sin \psi \\ \dot{y} = u \sin \psi + v \cos \psi \end{cases}$$

Hence since $\begin{cases} \cos(\psi + \frac{\pi}{2}) = -\sin \psi \\ \sin(\psi + \frac{\pi}{2}) = \cos \psi \end{cases}$

Hence

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

OR

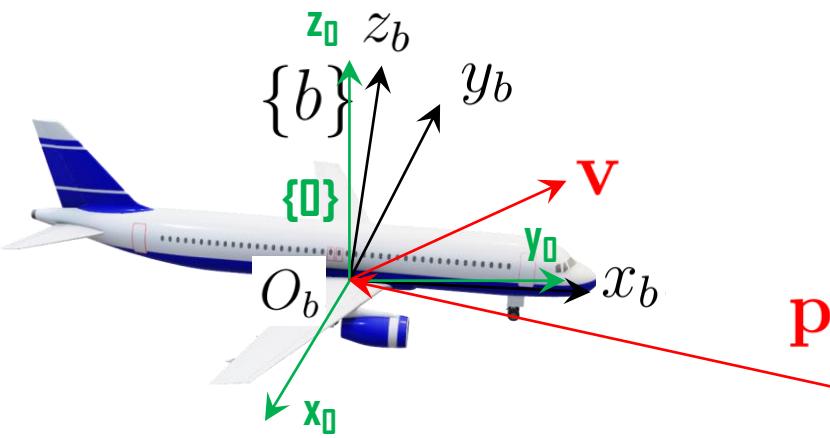
SDU

$$\boxed{\mathbf{v}_{b/e}^e = \mathbf{R}(\psi) \mathbf{v}_{b/e}^b}$$

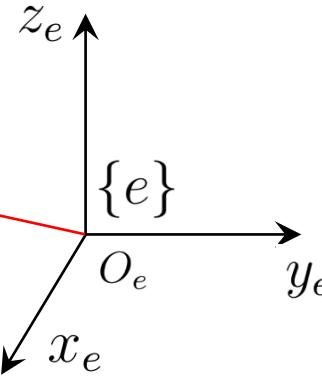
where $\mathbf{R}(\psi) := \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix}$

Important property: $\boxed{\mathbf{R}^{-1} = \mathbf{R}^T}$

Rotation in 3D and Euler angles



Consider the previous scenario but in 3D



Question: How are $\mathbf{v}_{b/e}^e$ and $\mathbf{v}_{b/e}^b$ related?

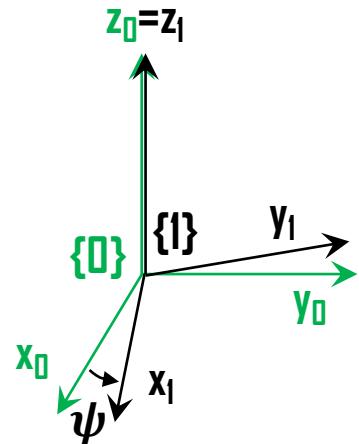
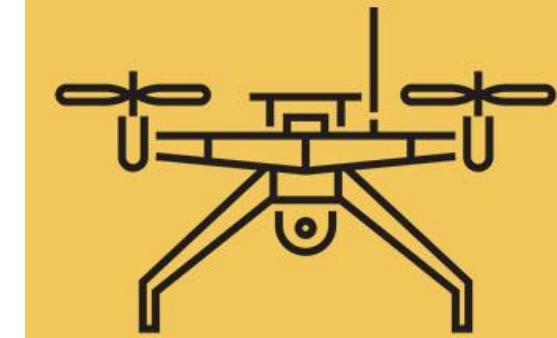
Related to how a frame $\{0\}$ parallel to $\{e\}$ is rotated to obtain frame $\{b\}$

How to describe/represent \mathbf{R}_b^e :

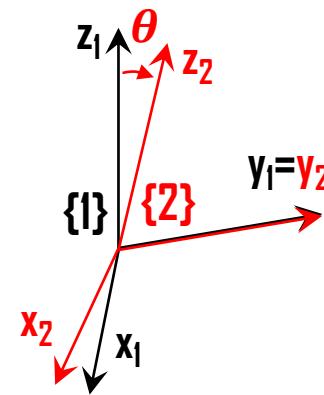
- Apply 3 successive rotations, each one in a different plane/around one of the 3 different axes x, y, z
- Each rotation of a specific angle connects one frame to another, starting from $\{0\}$ and ending with $\{b\}$
- The angles are roll, noted ϕ (angle around the x axis), pitch, noted θ (angle around the y axis), yaw, noted ψ (angle around the z axis)

From inertial frame to intermediate frames to body frame

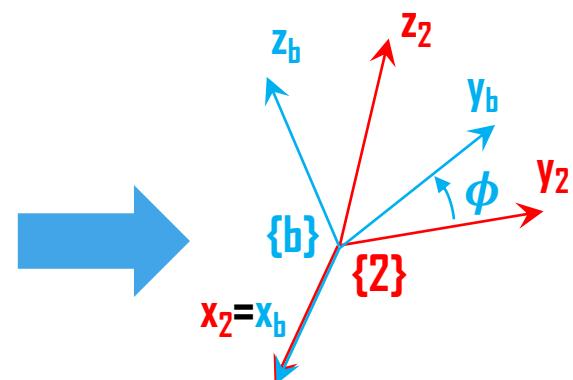
Apply 3 successive rotations around one of the main axes



From frame {0}
(parallel to earth-fixed
frame {e}) to frame {1}



From frame {1}
to frame {2}



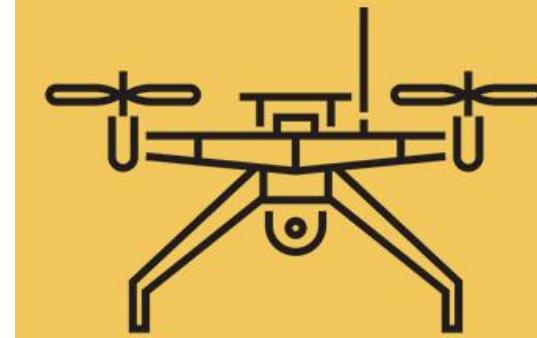
From frame {2} to
frame {b} (body-fixed
frame)

Important remark: we have chosen the sequence yaw-pitch-roll,
but other sequences are also possible...

Calculating the rotation matrix (1/2)

Start with

$$\mathbf{v}_{b/e}^e = \begin{bmatrix} v_x^e \\ v_y^e \\ v_z^e \end{bmatrix} = \mathbf{v}_{b/e}^0 = \begin{bmatrix} v_x^0 \\ v_y^0 \\ v_z^0 \end{bmatrix}$$



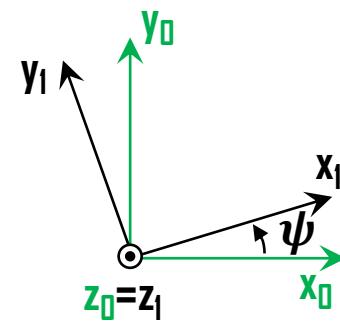
We want to obtain

$$\mathbf{v}_{b/e}^0 = \mathbf{R}_z(\psi) \mathbf{v}_{b/e}^1 \quad \text{rotation around z (right-hand rule)}$$

→ $\begin{bmatrix} v_x^0 \\ v_y^0 \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} v_x^1 \\ v_y^1 \end{bmatrix}$

Or, in 3D:

$$\begin{bmatrix} v_x^0 \\ v_y^0 \\ v_z^0 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}_z(\psi)} \begin{bmatrix} v_x^1 \\ v_y^1 \\ v_z^1 \end{bmatrix}$$



Calculating the rotation matrix (2/2)

rotation around y (right-hand rule)

$$\mathbf{v}_{b/e}^1 = \mathbf{R}_y(\theta) \mathbf{v}_{b/e}^2 \rightarrow \begin{bmatrix} v_z^1 \\ v_x^1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_z^2 \\ v_x^2 \end{bmatrix}$$

Or, in 3D:

$$\begin{bmatrix} v_x^1 \\ v_y^1 \\ v_z^1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} v_x^2 \\ v_y^2 \\ v_z^2 \end{bmatrix}$$

rotation around x

$$\mathbf{v}_{b/e}^2 = \mathbf{R}_x(\phi) \mathbf{v}_{b/e}^b \rightarrow \begin{bmatrix} v_y^2 \\ v_z^2 \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} v_y^b \\ v_z^b \end{bmatrix}$$

Or, in 3D:

$$\begin{bmatrix} v_x^2 \\ v_y^2 \\ v_z^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix}$$

Hence

$$\mathbf{v}_{b/e}^e = \mathbf{v}_{b/e}^0 = \mathbf{R}_z(\psi) \mathbf{v}_{b/e}^1$$

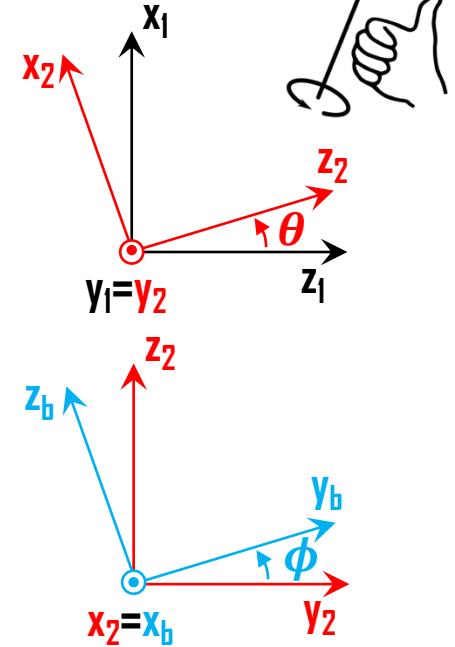
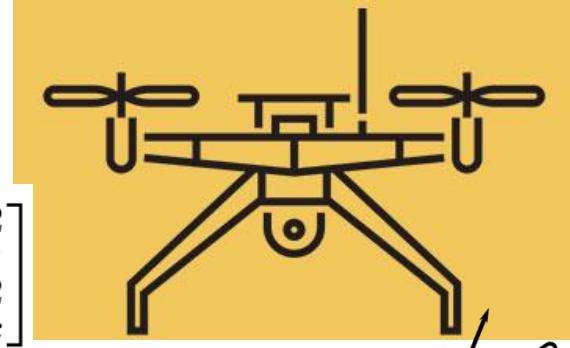
$$= \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{v}_{b/e}^2$$

and $\boxed{\mathbf{v}_{b/e}^e = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi) \mathbf{v}_{b/e}^b}$

SDU or

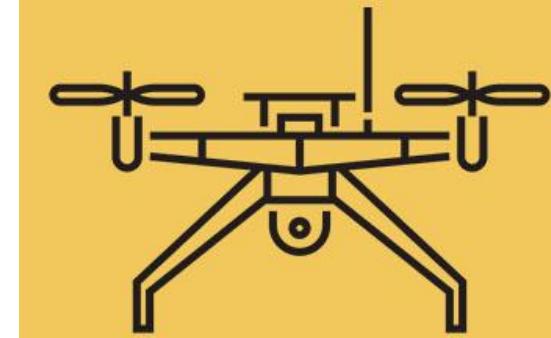
$\boxed{\mathbf{v}_{b/e}^e = \mathbf{R}_b^e(\Theta) \mathbf{v}_{b/e}^b}$

with $\mathbf{R}_b^e(\Theta) = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi)$

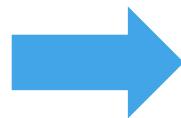
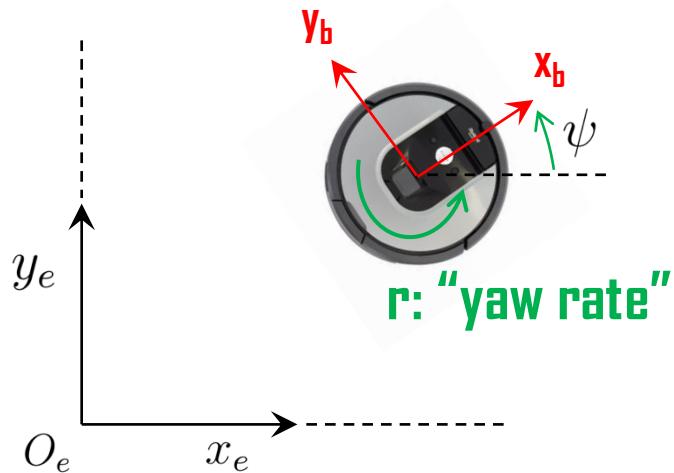


$\Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$ Euler angle vector

Angular velocity transformation: the 2D case



Let us come back to the 2D case:



$$\frac{d}{dt} \psi = \dot{\psi} = r$$

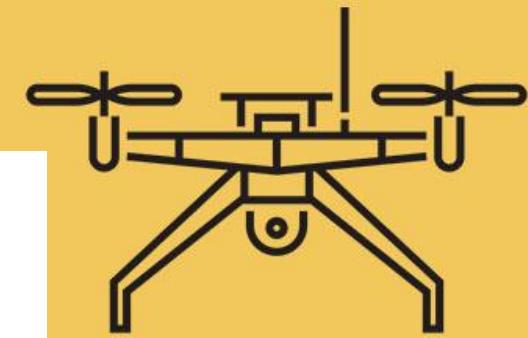
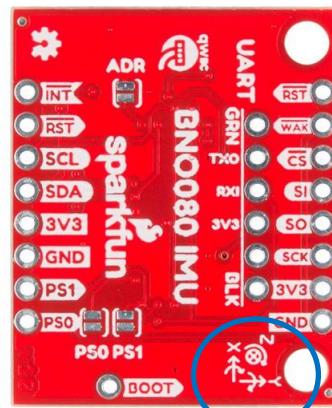
The link between $\dot{\psi}$ and r is
pretty easy 😊

But things are a bit
more involved in 3D...

Angular velocity transformation in 3D (1/3)

“strapdown” sensors, i.e.
gyroscopes give angular
velocities in the body-fixed frame,
so that we have

$$\omega_{b/e}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$



How can we link this back to the inertial frame?

i.e. we want to relate $\omega_{b/e}^b$ and $\omega_{b/e}^e = \frac{d}{dt}\Theta = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$

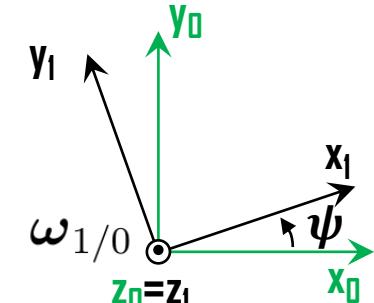
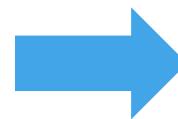
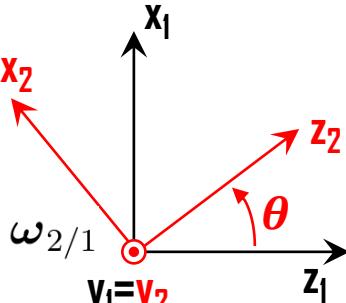
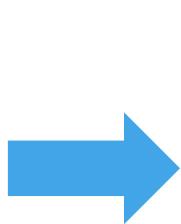
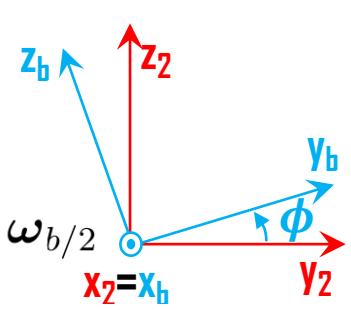
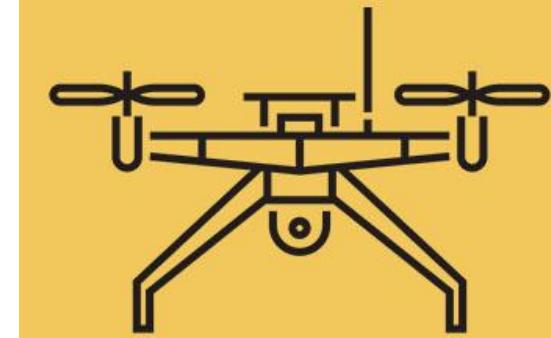
Each of the 3 rotations around
one axis contributes to angular
velocity, and we have

$$\omega_{b/e}^b = \omega_{b/2}^b + \omega_{2/1}^b + \omega_{1/e}^b$$

Angular velocity transformation in 3D (2/3)

Use again

$$\omega_{b/e}^b = \omega_{b/2}^b + \omega_{2/1}^b + \omega_{1/e}^b$$



$$\omega_{b/2}^2 = \omega_{b/2}^b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \dot{\phi} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

Expressed in $\{b\}$ frame, OK.

$$\omega_{2/1}^1 = \omega_{2/1}^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \dot{\theta} = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$$

OK, but not expressed in $\{b\}$ yet.

$$\omega_{1/0}^0 = \omega_{1/0}^1 = \omega_{1/e}^e = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\psi} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

OK, but not expressed in $\{b\}$ yet.

$$\omega_{b/2}^b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$



$$\omega_{2/1}^b = \mathbf{R}_x^{-1}(\phi) \omega_{2/1}^2 = \mathbf{R}_x^T(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$$

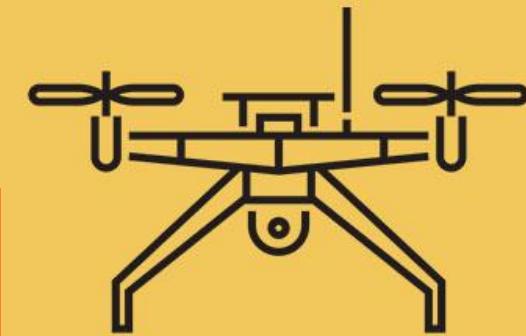


$$\begin{aligned} \omega_{1/e}^b &= \mathbf{R}_x^{-1}(\phi) \omega_{1/e}^0 \\ &= \mathbf{R}_x^{-1}(\phi) \mathbf{R}_y^{-1}(\theta) \omega_{1/e}^1 \\ &= \mathbf{R}_x^T(\phi) \mathbf{R}_y^T(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \end{aligned}$$

Finally, we have

$$\omega_{b/e}^b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x^T(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_x^T(\phi) \mathbf{R}_y^T(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (1)$$

Angular velocity transformation in 3D (3/3)



Previous equation (1)
can be rewritten as

$$\omega_{b/e}^b = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Let the angular velocity transformation between $\omega_{b/e}^b$ and $\dot{\Theta}$ be $T_\Theta(\Theta)$

Then, the above equation reads

$$\omega_{b/e}^b = T_\Theta^{-1}(\Theta) \dot{\Theta}$$

Inverting this relation gives

$$\dot{\Theta} = T_\Theta(\Theta) \omega_{b/e}^b, \quad \Theta(0) = \Theta_0$$

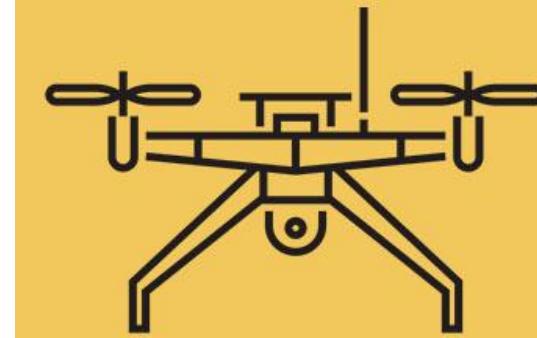
with

$$T_\Theta(\Theta) = \begin{bmatrix} 1 & \sin \phi \frac{\sin \theta}{\cos \theta} & \cos \phi \frac{\sin \theta}{\cos \theta} \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix}$$

Remarks: $T_\Theta(\Theta)$ is not a rotation matrix, i.e. $T_\Theta^T(\Theta) \neq T_\Theta^{-1}(\Theta)$

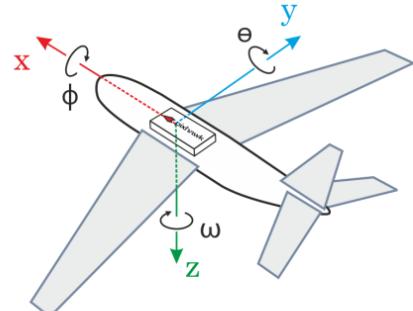
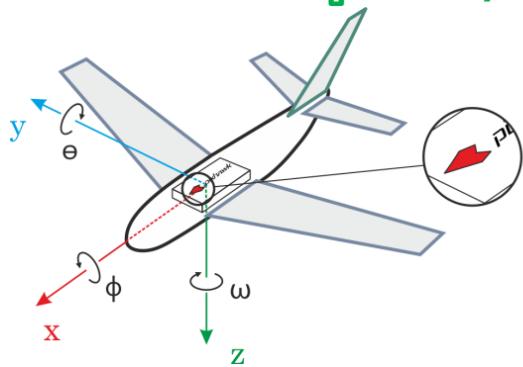
NED and ENU coordinates

There are different conventions for choosing the axes on a particular vehicle (always right-hand rule frames)

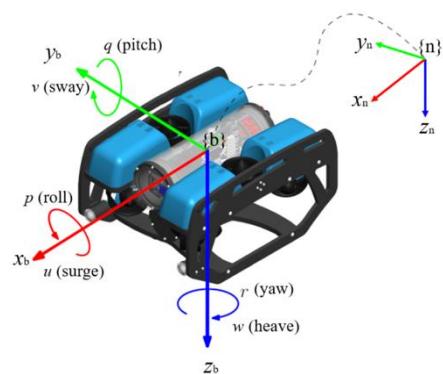


**NED coordinates
(North-East-Down)** (yaw seen CW from the top)

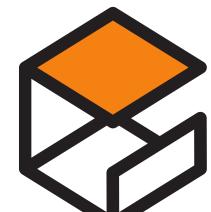
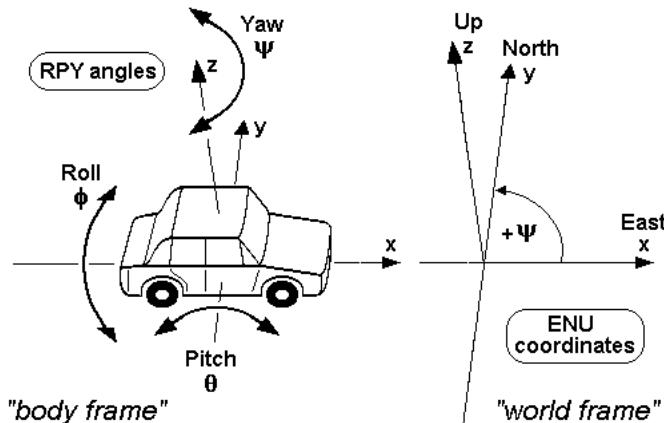
(on the vehicle: Forward-Right-Down)



ARDUPILOT



**ENU coordinates
(East-North-Up)**



GAZEBO

**PX4
autopilot**

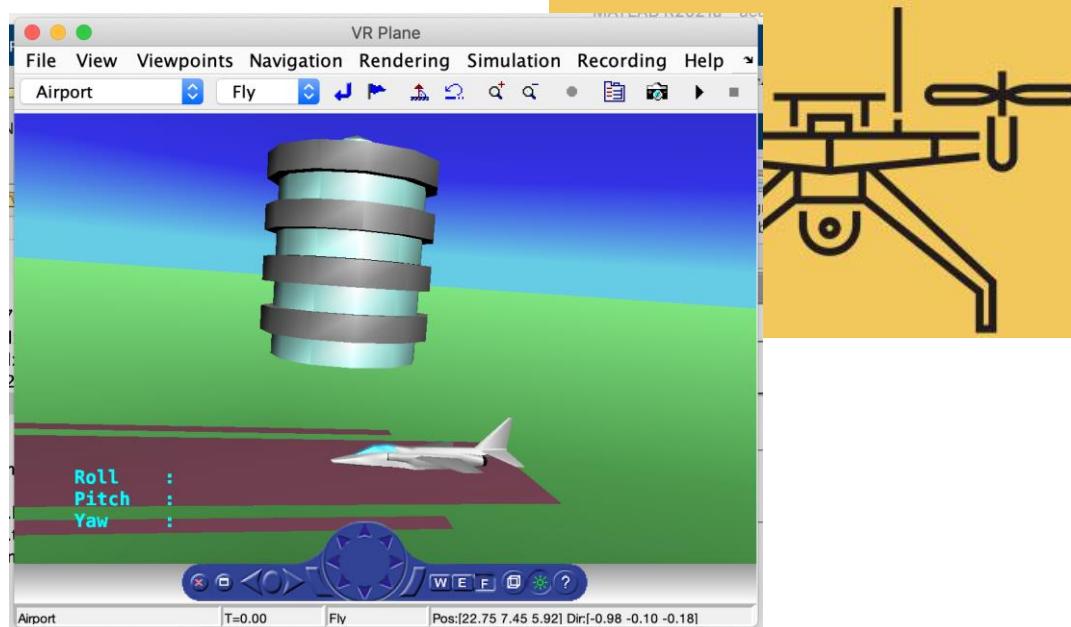


Let us take a break



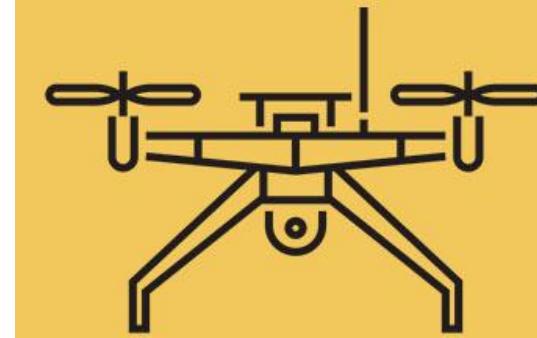
Today's exercise (1/2)

Program your own rotation matrix to start your own simulator in Matlab/Simulink



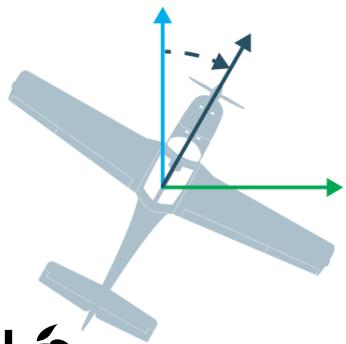
- Q1) From Itslearning, download and extract the zip file **Plane_files**. In Simulink, open the file **plane_temp.slx** and run it to make sure that it works.
- Q2) Make a simple program to implement a rotation matrix so that the plane has all Euler angles at 90 degrees (use the 'observe plane' viewpoint to in the VR view to see things better).
- Q3) Find 2 other ways to obtain the same attitude by re-organizing the order of the rotation matrices.

Today's exercise (2/2)

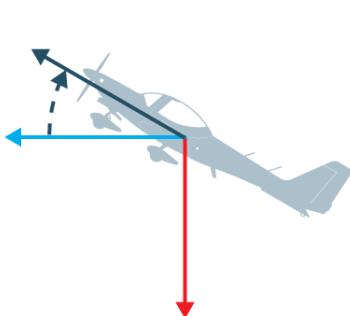


- Q4) Add some sliding gains so that you can control the Euler angles online
- In Matlab, use the symbolic toolbox to obtain an expression of rotation matrix R as a function of roll, pitch and yaw
- Q5) Include a "Rotation angles to direction cosines matrix" block and compare this with your previous results. Where is the difference coming from? (feel free to look under the mask of this block: right-click "mask/look under mask")

Yaw
(around Z axis)



Pitch
(around Y axis)



Roll
(around X axis)

