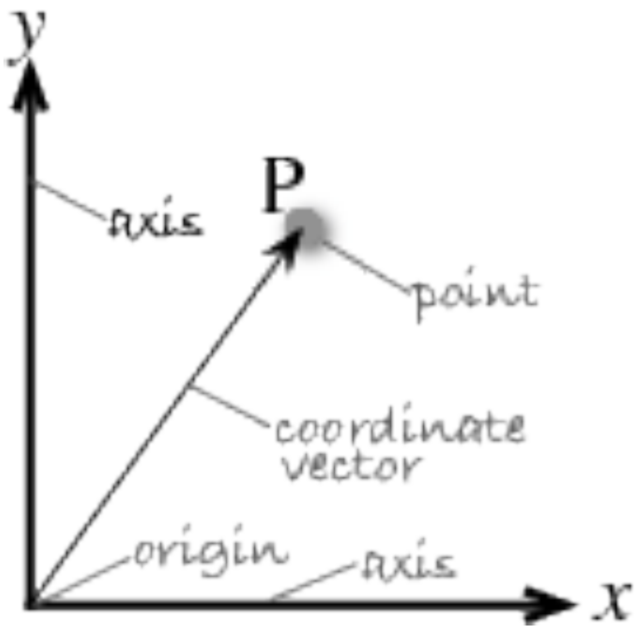
POSE AND ORIENTATION

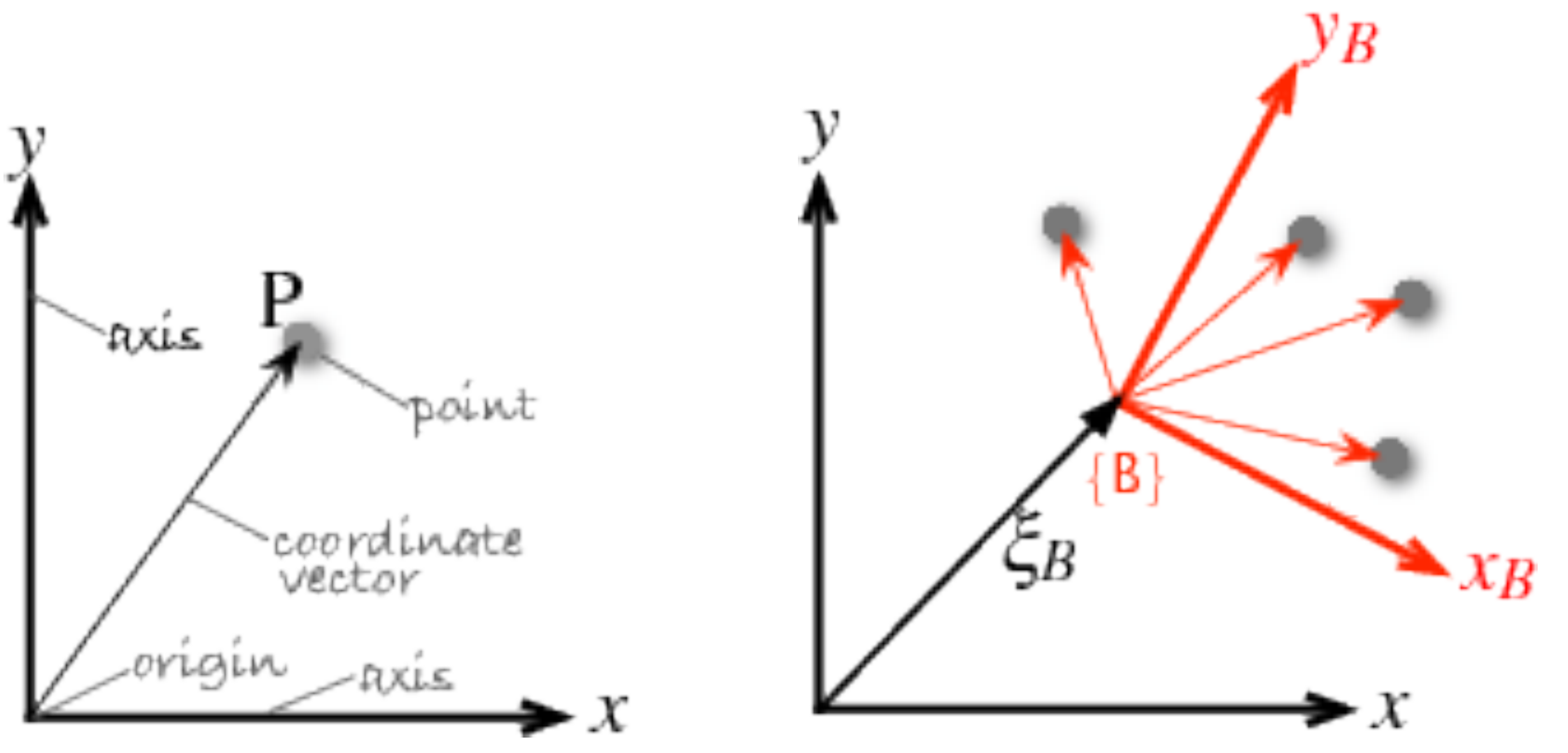
CS 3630 Introduction to Robotics and Perception

Frank Dellaert

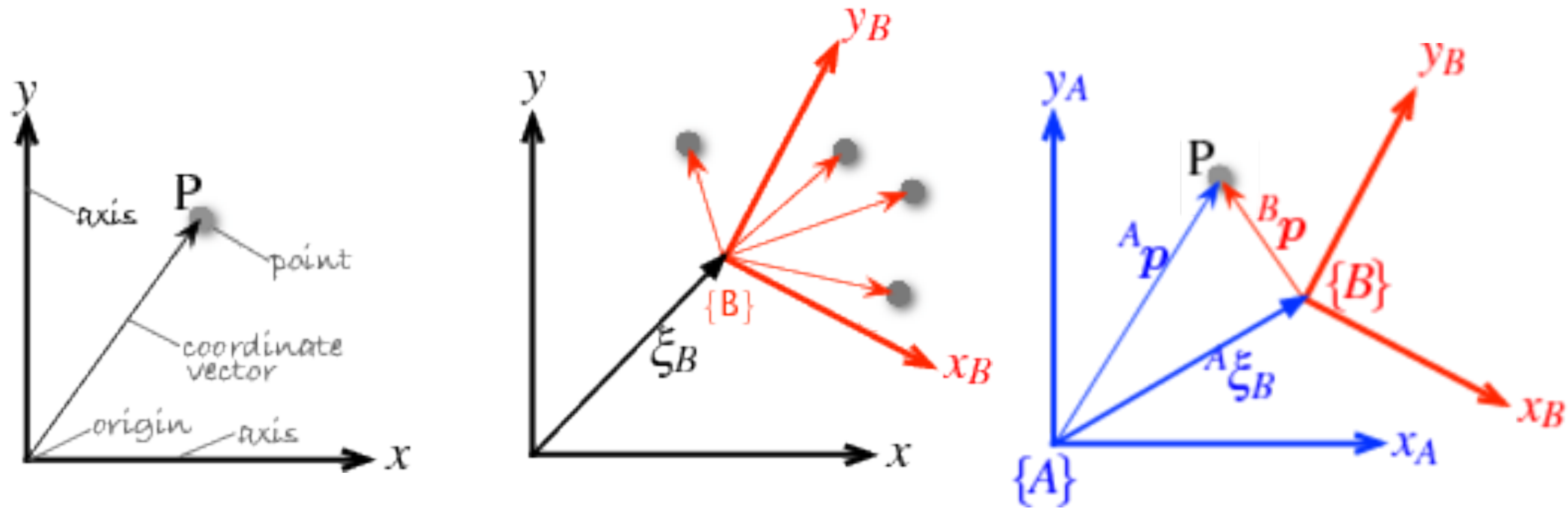
POINTS AND POSES



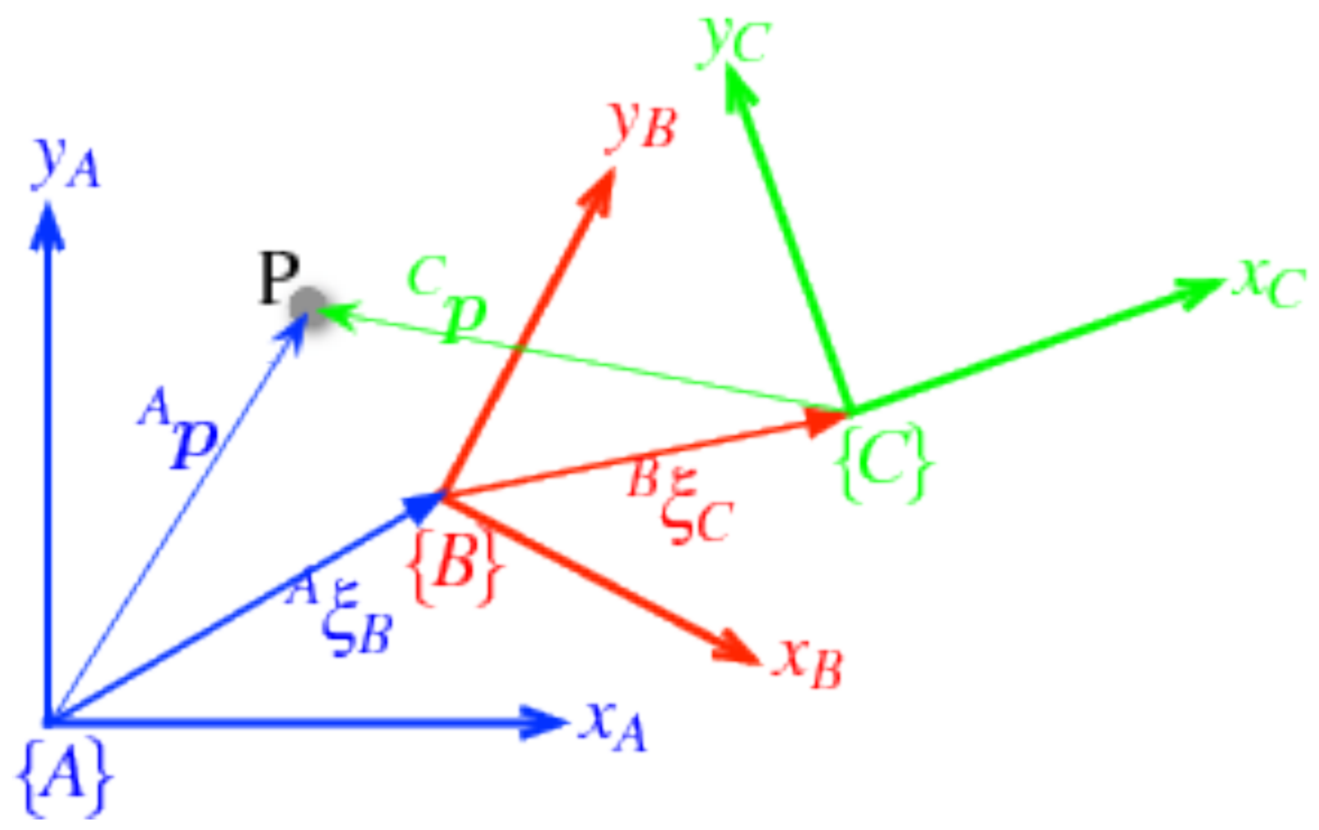
POINTS AND POSES



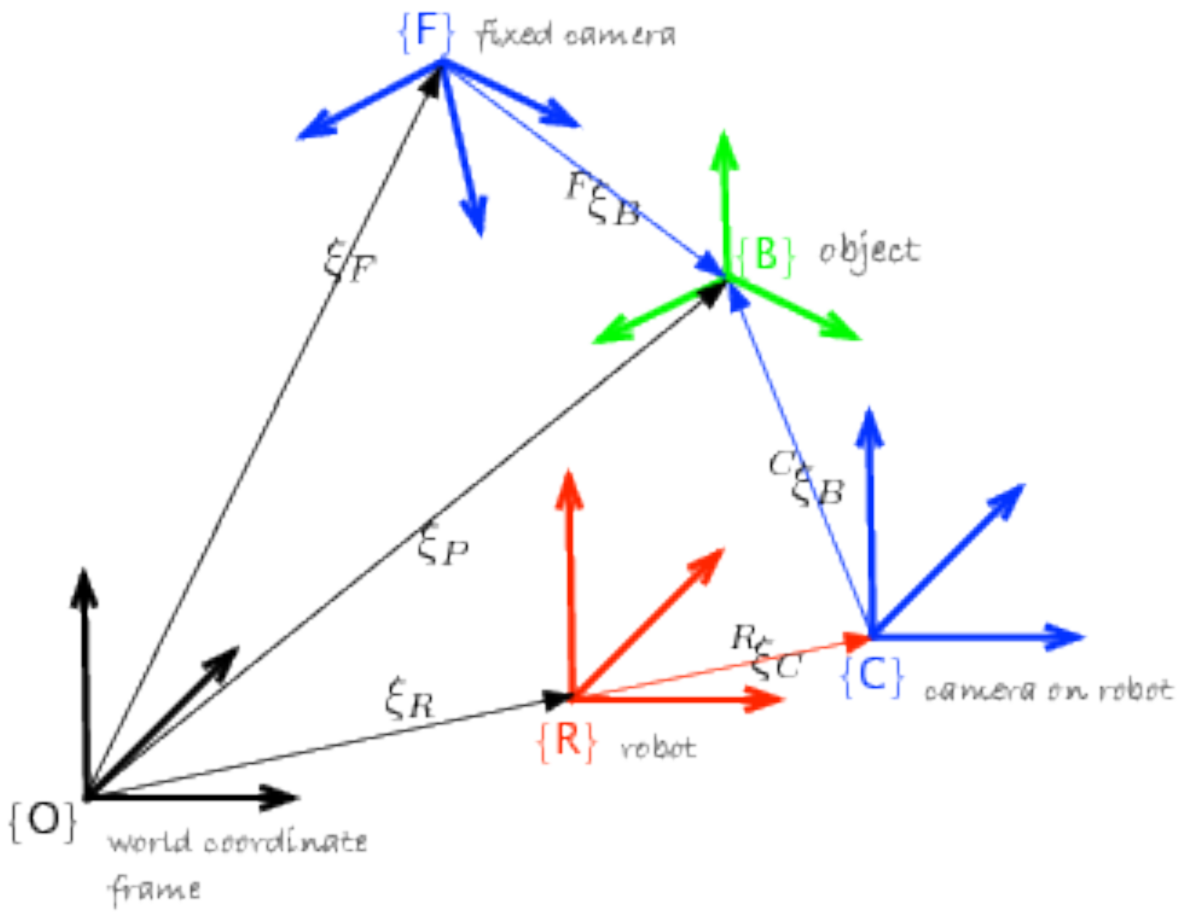
POINTS AND POSES



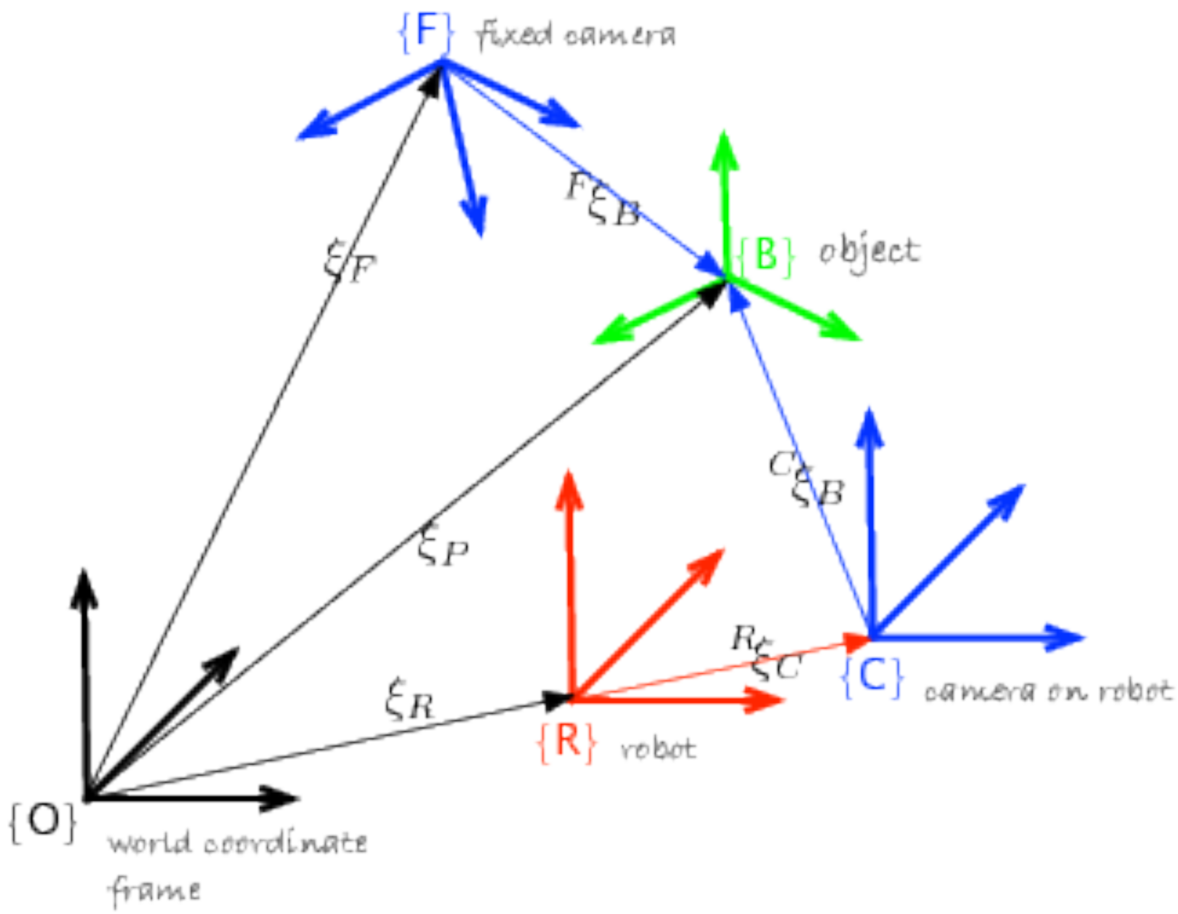
COMPOSING POSES

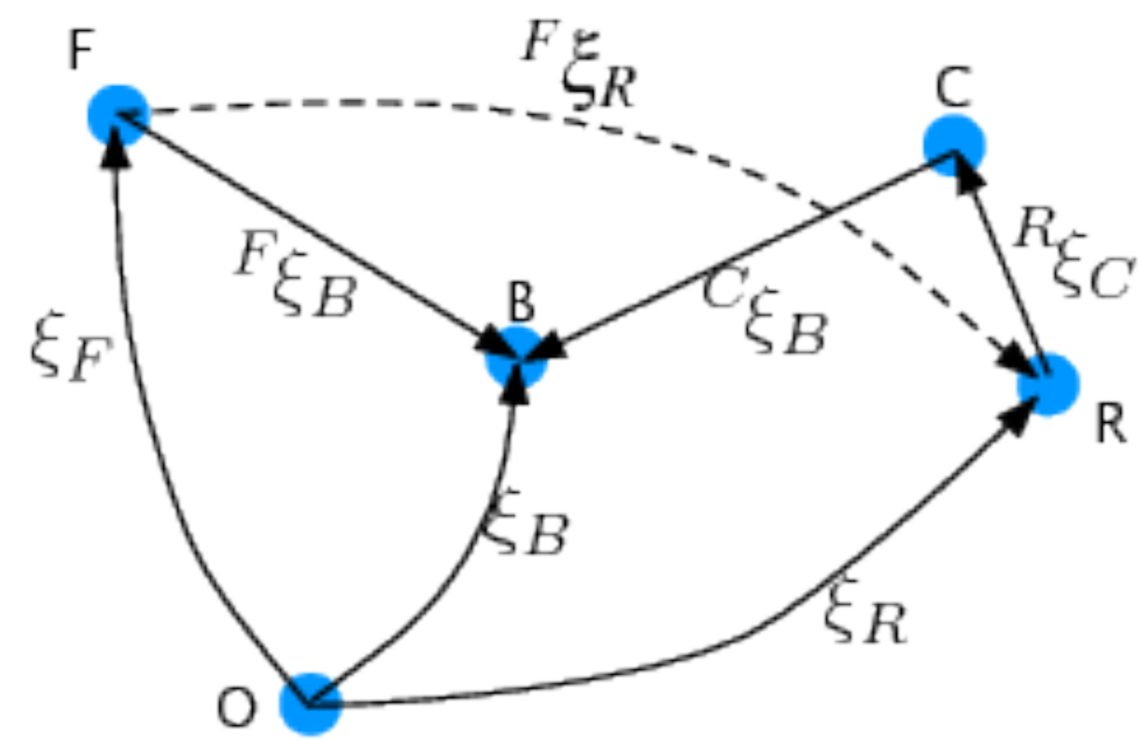


POSE ALGEBRA

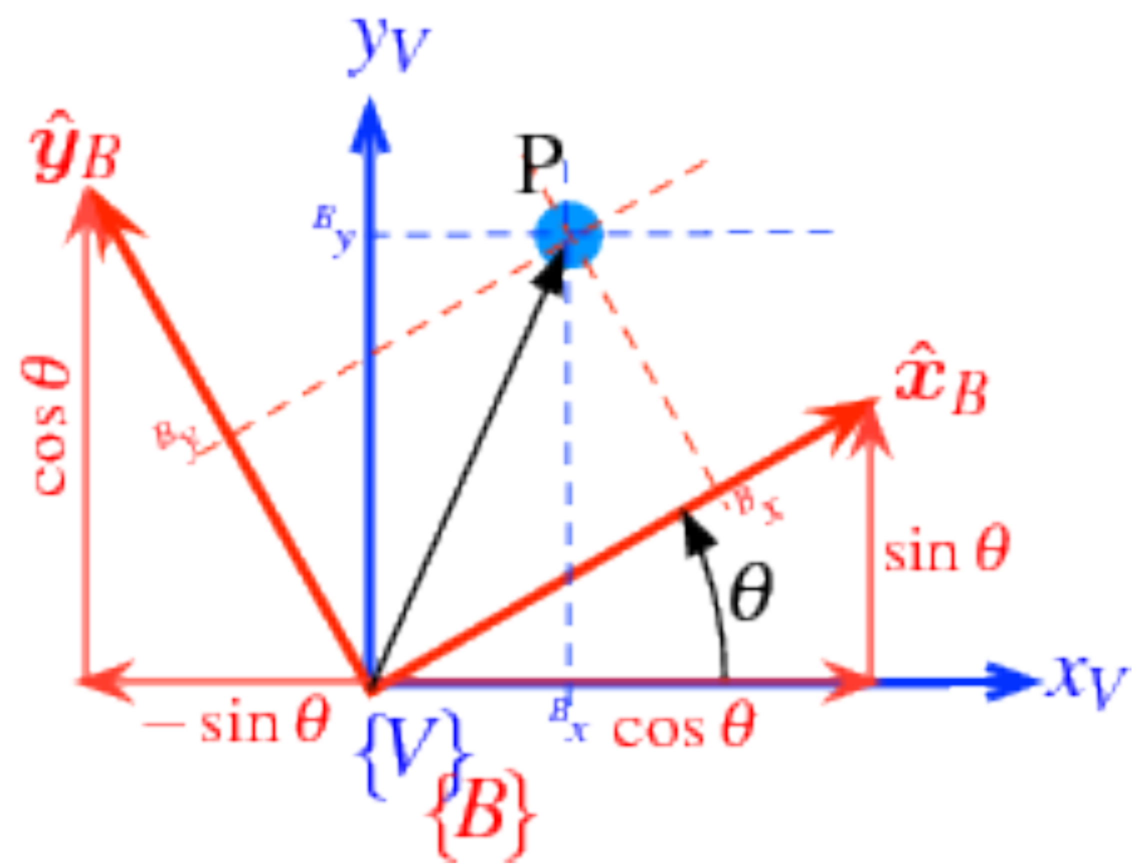


POSE ALGEBRA



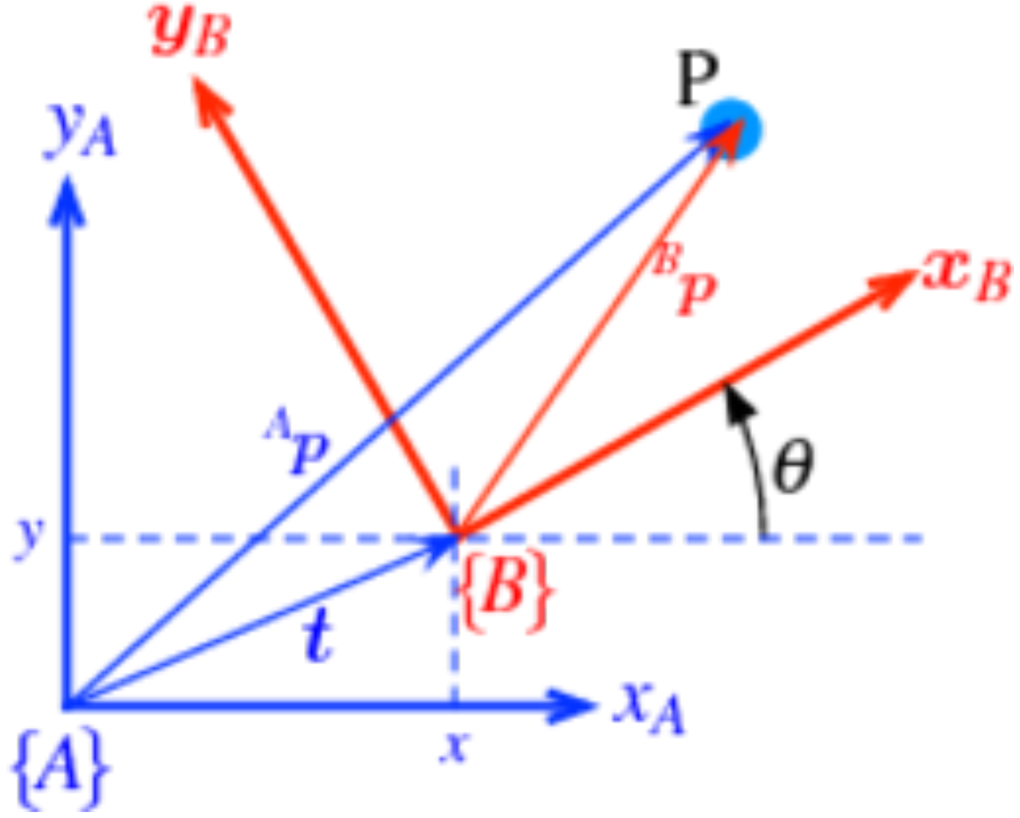


ROTATION IN 2D



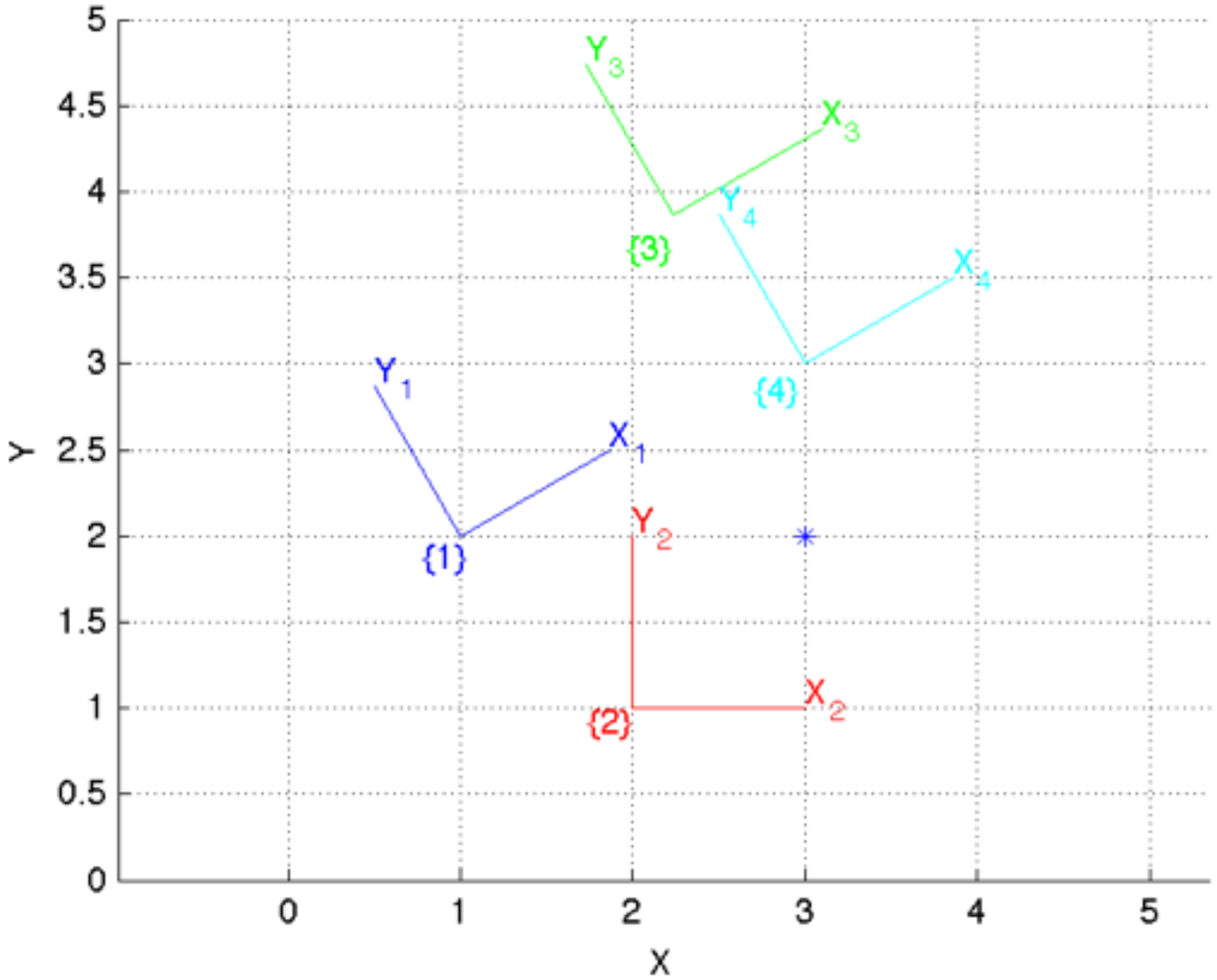
- Rotation Matrices (DCM), 2x2 \in **SO(2)**
- Columns of **vRb**: axes of B in V. In MATLAB notation: [c -s; s c]
- 4 numbers, but only 1D **Manifold**

POSES IN 2D



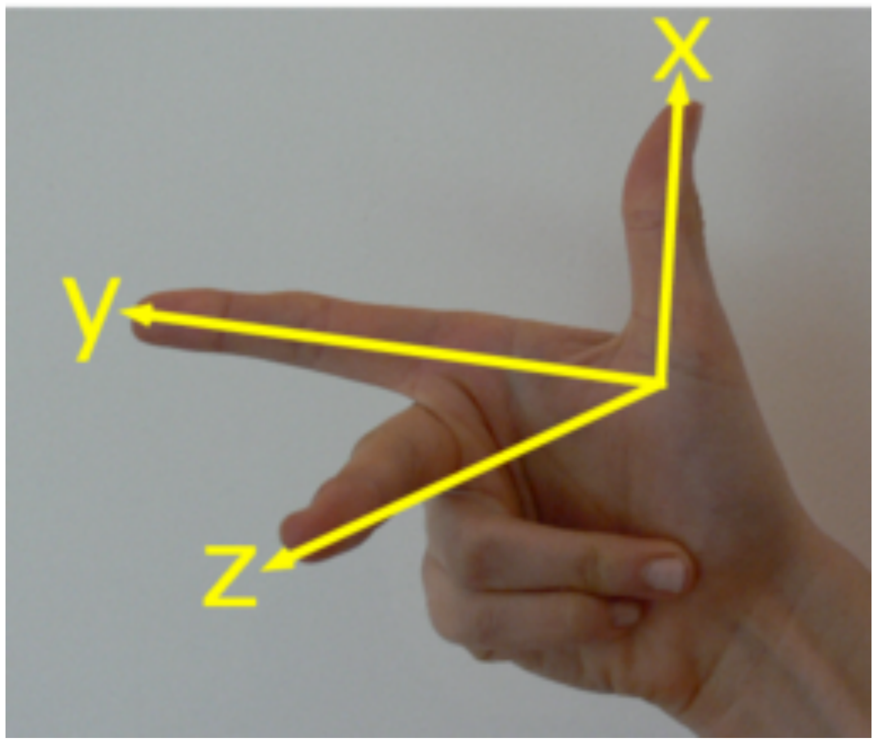
- (x,y,\theta) or (R,t)
- Better: SE(2), next slide

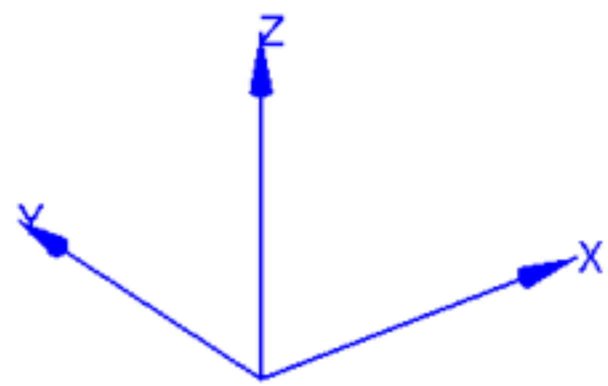
SE(2)

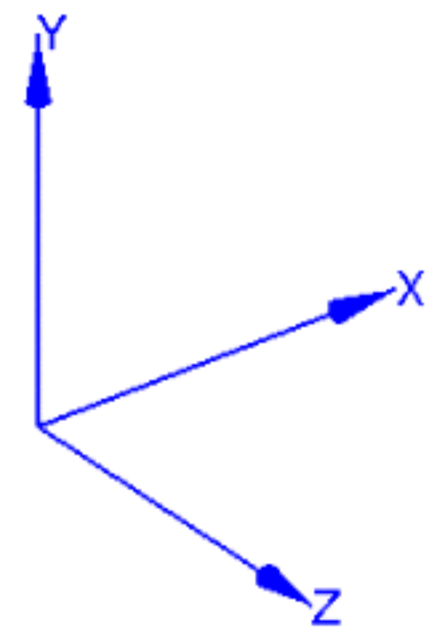


- [R t; 0 0 1] = 3x3 matrix \in **SE(2)**

ROTATIONS IN 3D

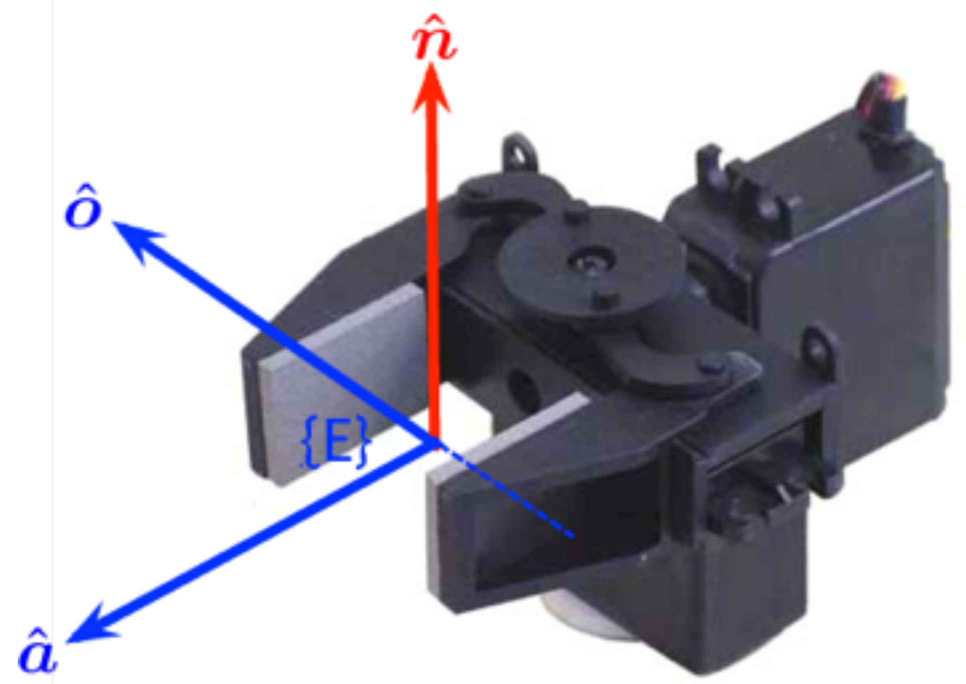






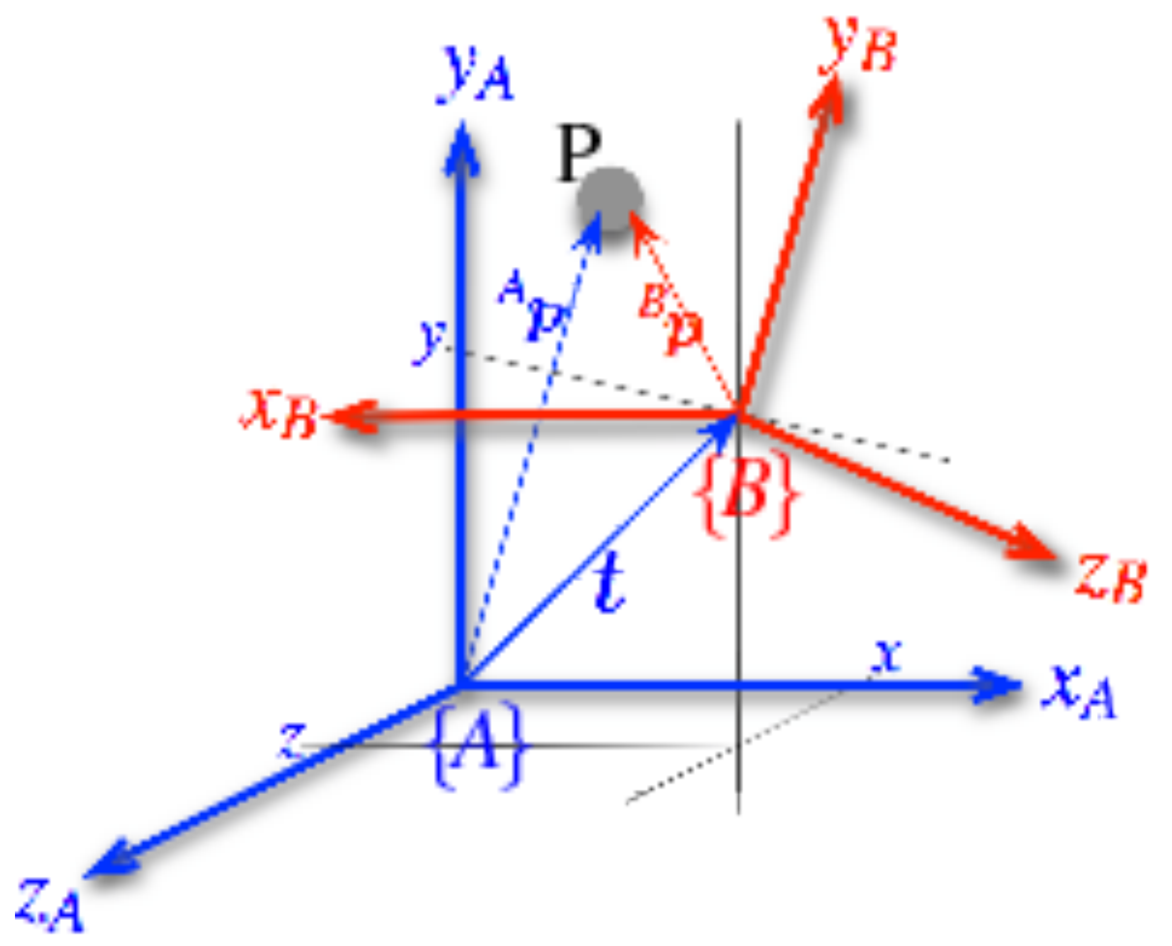
- Rotation Matrices (DCM), 3x3 \in **SO(3)**
- Columns of **aRb**: axes of B in A. In MATLAB notation: [Xb Yb Zb]
- 9 numbers, but 3D **Manifold**

REPRESENTING END-EFFECTOR POSE



- Z_E = approach vector a
- Y_E = orientation vector o
- X_E = o x a

POSES IN 3D



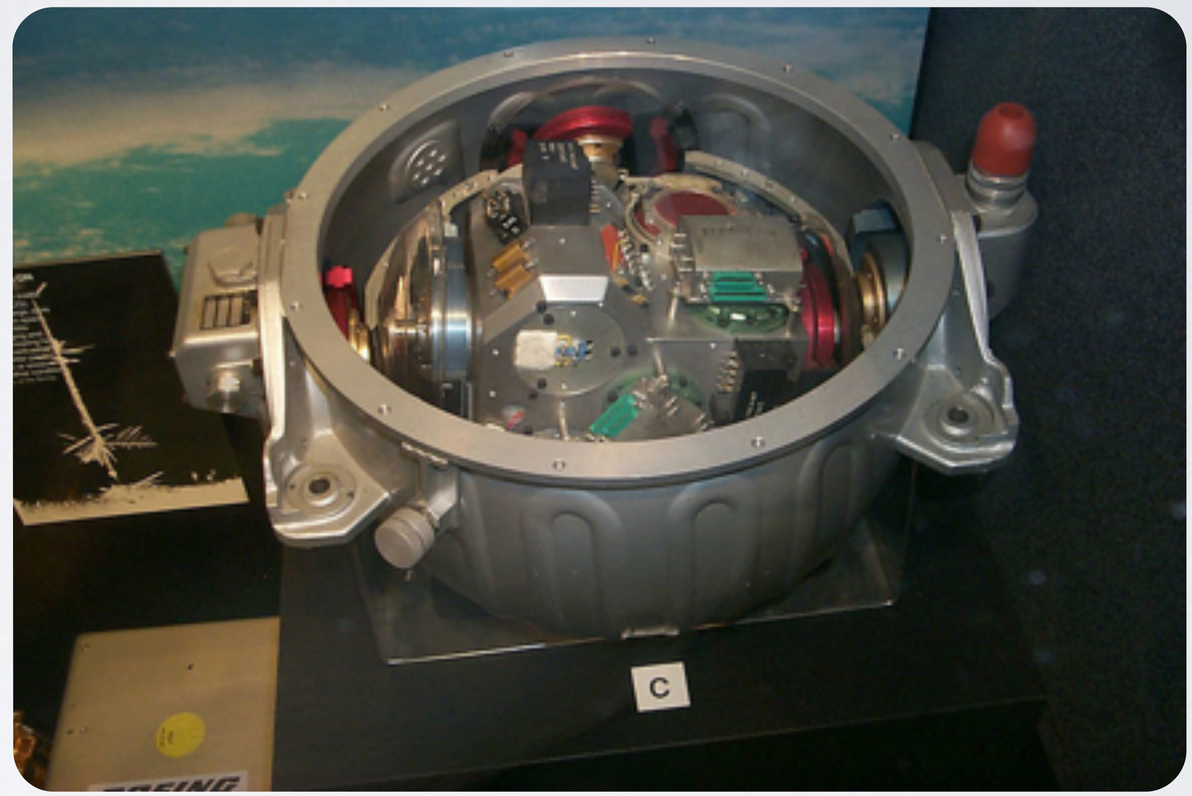
- [R t; 0 0 0 1], 4x4 matrix in **SE(3)**

REPRESENTING 3D ROTATIONS

- Rotation Matrices (DCM)
- Euler Angles
  - Eulerian
  - Cardanian
  - Gimbal Lock
- Axis-Angle
- Unit Quaternions

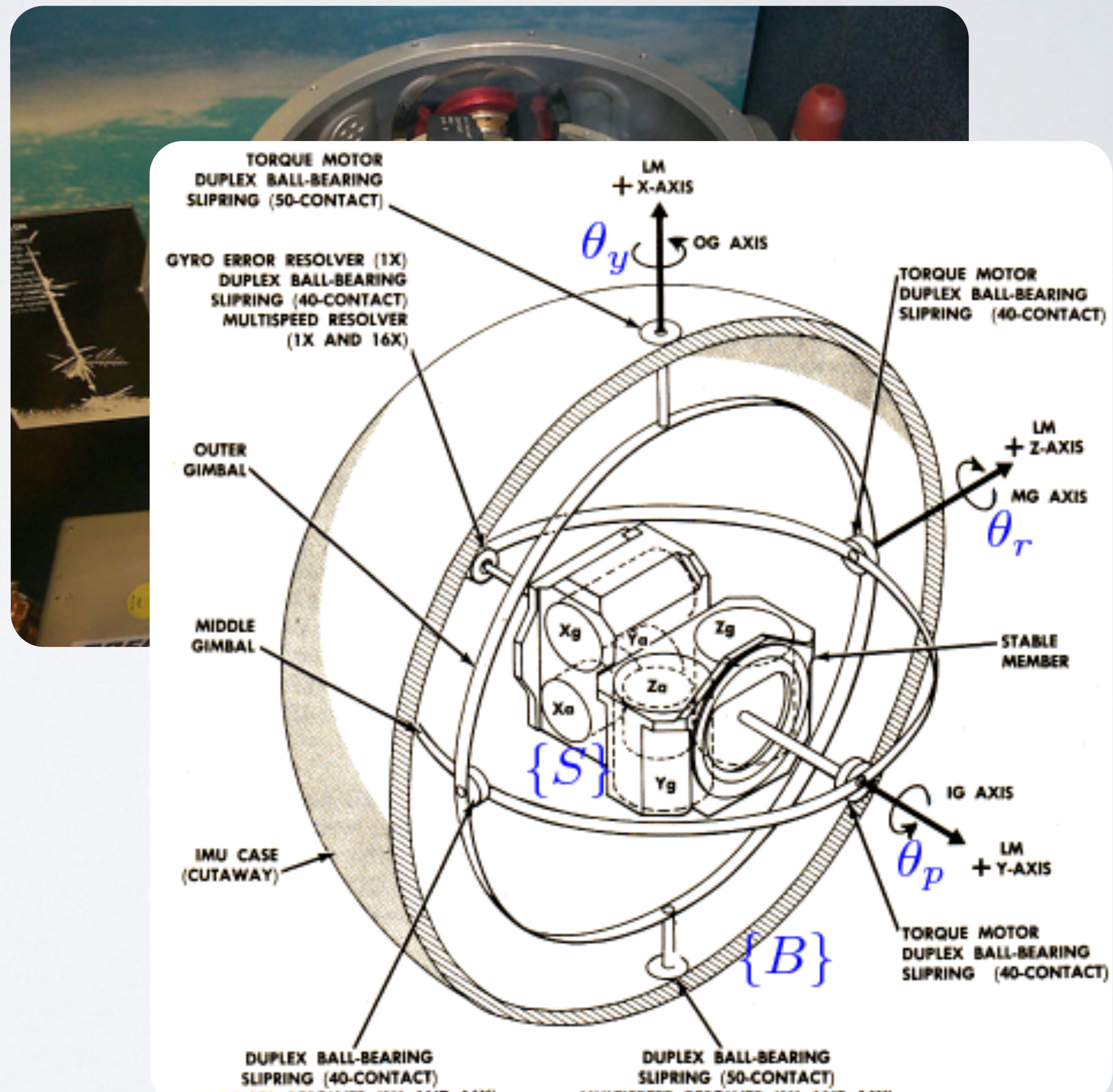
REPRESENTING 3D ROTATIONS

- Rotation Matrices (DCM)
- Euler Angles
  - Eulerian
  - Cardanian
  - Gimbal Lock
- Axis-Angle
- Unit Quaternions

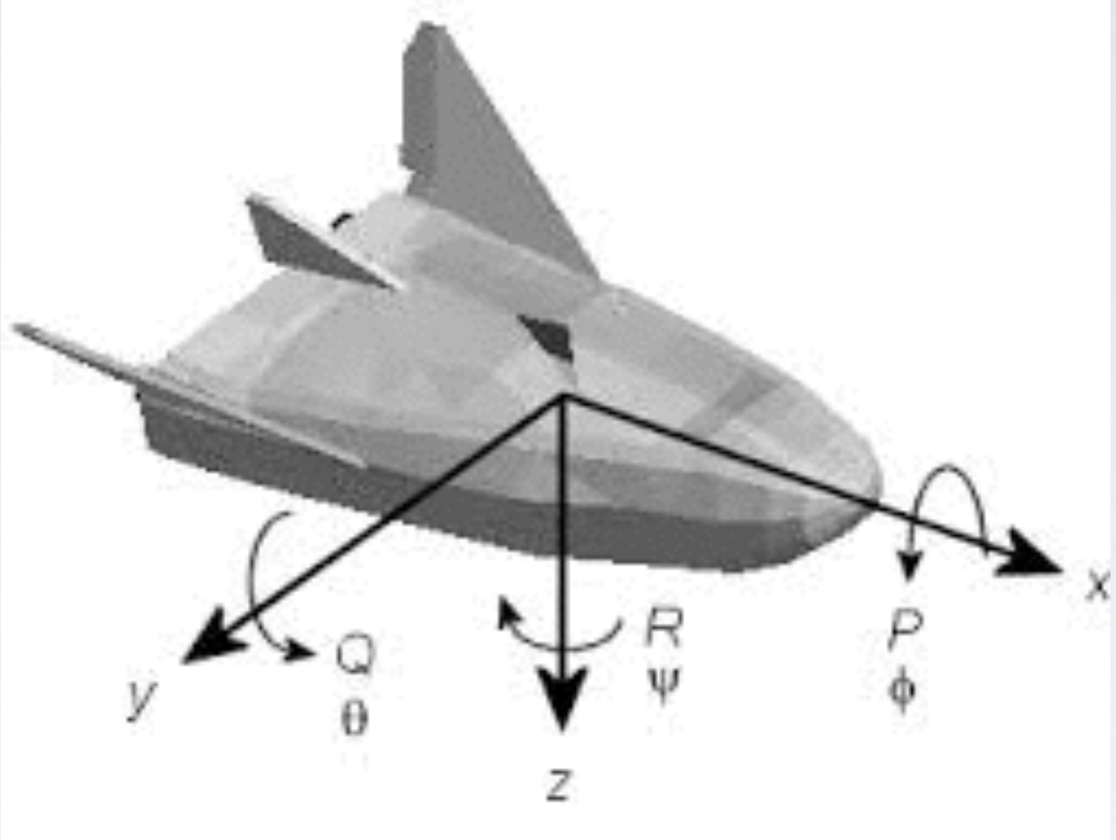


REPRESENTING 3D ROTATIONS

- Rotation Matrices (DCM)
- Euler Angles
  - Eulerian
  - Cardanian
  - Gimbal Lock
- Axis-Angle
- Unit Quaternions



ROLL-PITCH-YAW



$$\mathcal{R}_v^b(\phi,\theta,\psi) = \mathcal{R}_{v2}^b(\phi)\mathcal{R}_{v1}^{v2}(\theta)\mathcal{R}_v^{v1}(\psi) \tag{2.4}$$

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

$$= \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{pmatrix}, \tag{2.5}$$

Beard, 2011, Small Unmanned Aircraft

UNIT QUATERNIONS

- 2D:
  - θ
  - [cθ -sθ; sθ cθ]
  - z = (cθ, sθ)
- 3D:
  - θ_r, θ_p, θ_y
  - [r_{11} r_{12} r_{13};r_{21} r_{22} r_{23};r_{31} r_{32} r_{33}]
  - q = c(θ/2) <s(θ/2) ***v***>

For the case of quaternions our generalized pose is $\xi \sim \mathring{q} \in \mathbb{Q}$ and

$$\mathring{q}_1 \oplus \mathring{q}_2 \mapsto s_1 s_2 - \boldsymbol{v}_1 \cdot \boldsymbol{v}_2, < s_1 \boldsymbol{v}_2 + s_2 \boldsymbol{v}_1 + \boldsymbol{v}_1 \times \boldsymbol{v}_2 >$$

which is known as the quaternion or Hamilton product, and

$$\ominus \mathring{q} \mapsto \mathring{q}^{-1} = s, < -\boldsymbol{v} >$$

which is the quaternion conjugate. The zero pose $0 \mapsto 1 <0, 0, 0>$ which is the identity quaternion. A vector $\boldsymbol{v} \in \mathbb{R}^3$ is rotated $\mathring{q} \cdot \boldsymbol{v} \mapsto \mathring{q}\mathring{q}(\boldsymbol{v})\mathring{q}^{-1}$ where $\mathring{q}(\boldsymbol{v}) = 0, <\boldsymbol{v}>$ is known as a pure quaternion.