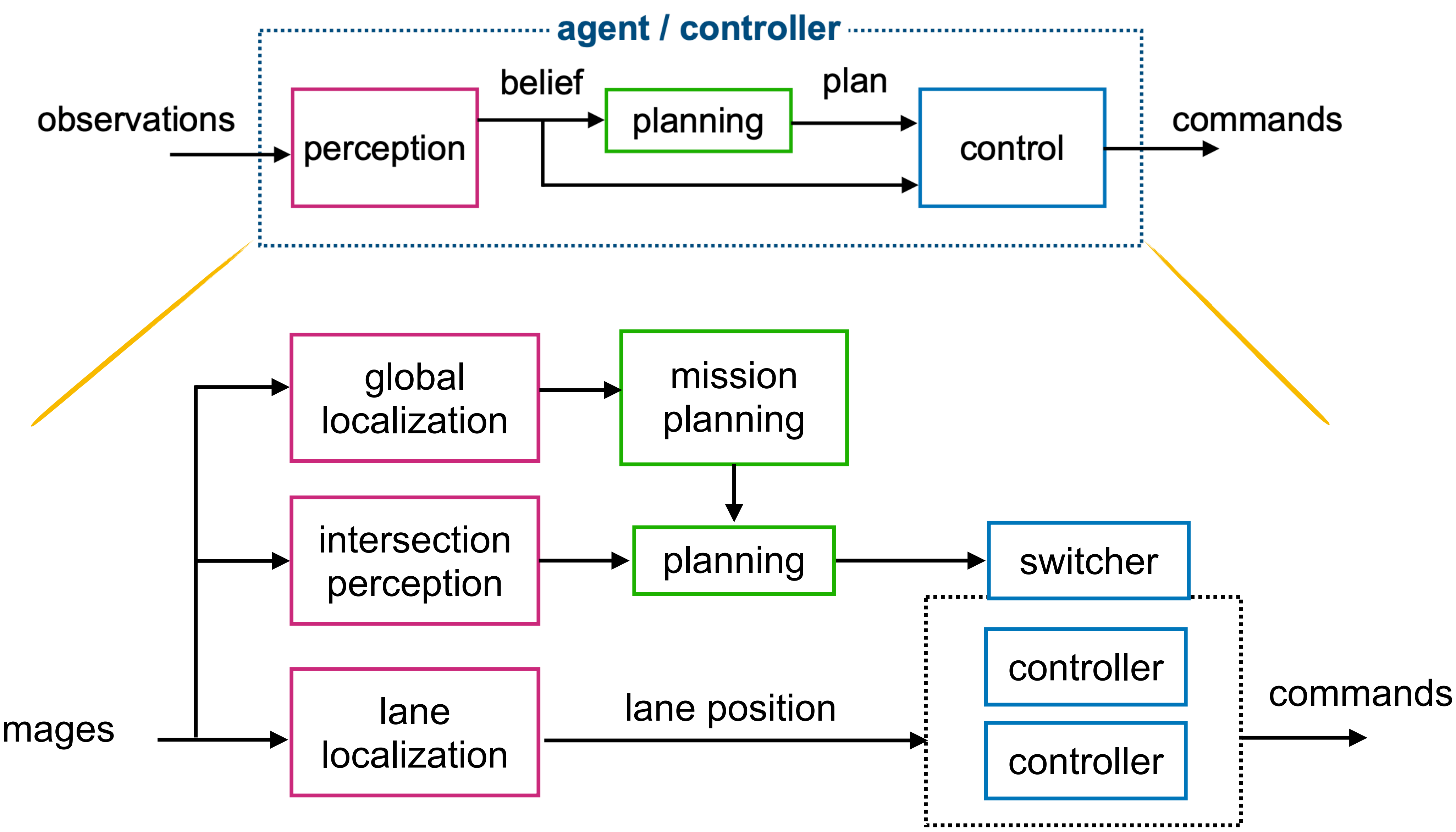
Representations

Representations: Fundamentals

- Representations of the **robot** and its **environment** are fundamental to the capabilities that make a robot autonomous (i.e., sense, plan, and act)



Representations: State

- The (**world**) ***state*** exists independently of you and your algorithms
  - What we usually call "state" (e.g., in control systems) is only a small part of the world state
  - What we call "noise" is usually used to mask our ignorance
- **Markov property*:*** the future is independent of the past given the present

$$x_t \in \mathscr{X} \qquad p(x_{t+1} \,|\, x_t, a_t, x_{t-1}, a_{t-1}, \ldots, x_0, a_0) = p(x_{t+1} \,|\, x_t, a_t)$$

- Markov representations occur throughout AI and machine learning (e.g., speech understanding, natural language processing, computer vision, …)

Representations: Measurement History

- The state is typically observed via the robot's sensors
- Measurement history seems like a logical choice for state
- **Pros**:
  - Sufficient: implicitly captures all knowledge that can be gleaned from sensor data
  - Lowest level representation
- **Cons**:
  - Measurements are redundant and convey unnecessary information
  - Computationally and memory inefficient: number of measurements increases linearly with time

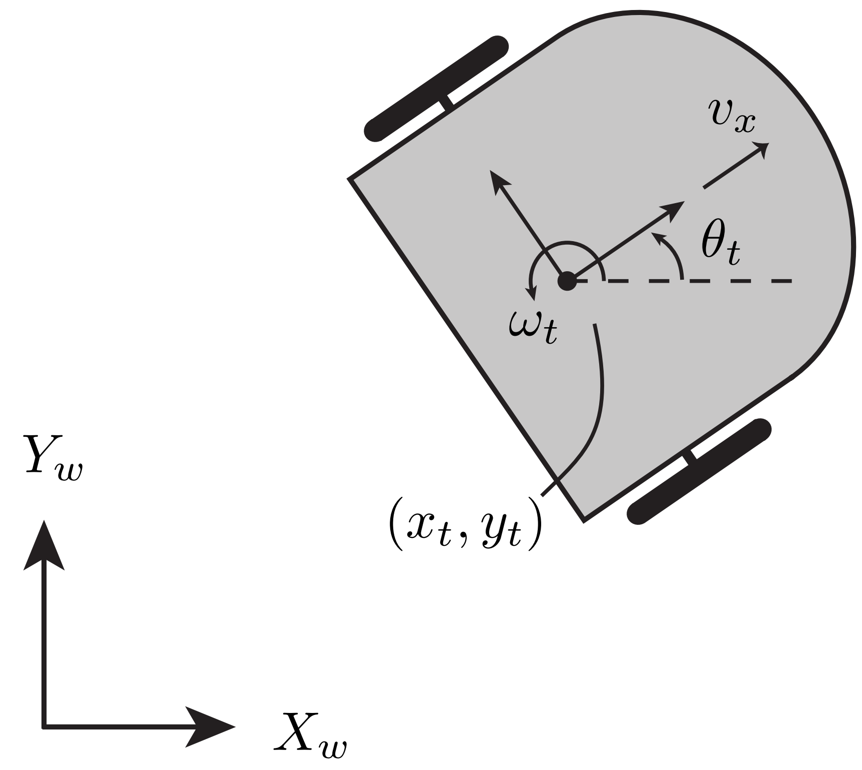
Robot Representations

- The robot's state typically includes its ***pose*** $\mathbf{x}_t$, which specifies its position and orientation relative to a fixed reference frame.
- The pose defines a rigid-body transformation from a robot-fixed frame to the "world frame"
- May also include body-frame linear angular velocities

$$\mathrm{x}_t = \{x_t, y_t, \theta_t\} \in SE(2)$$

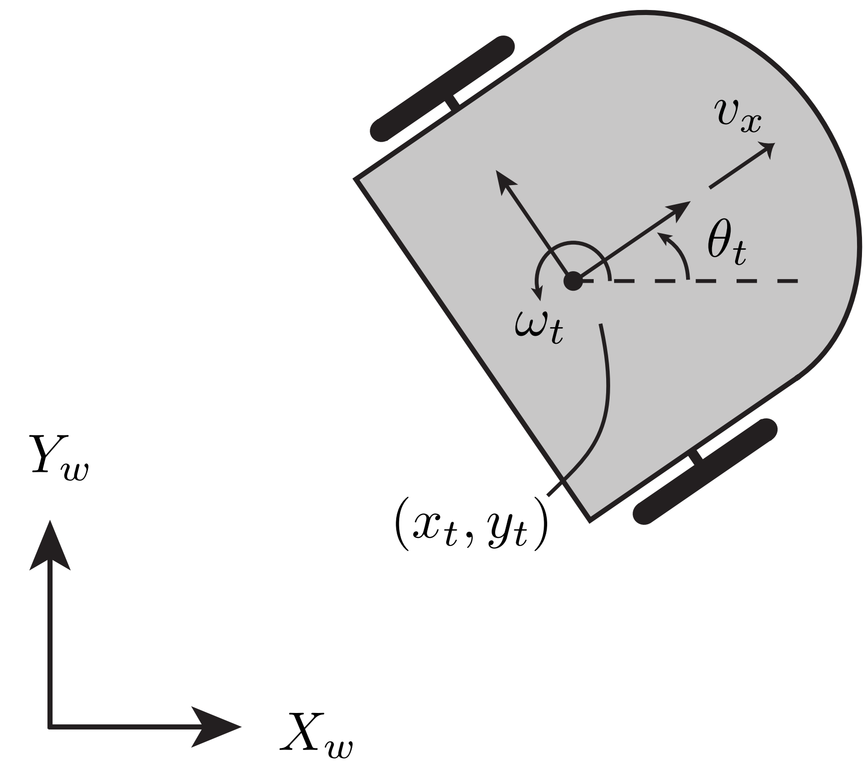
position $(x_t, y_t) \in \mathbb{R}^2$

orientation θ_t



Duckiebot Frames

- Space: $\mathbb{R}^2$
- **World Frame**: $\{x_w, y_w\}$ origin fixed at W
- **Body (robot) frame**: $\{x_r, y_r\}$ Center at A, axle midpoint x_r forms orientation angle θ with x_A



Translations

- Cartesian representation of point $P \in \mathbb{R}^2$ with respect to reference with origin in A:

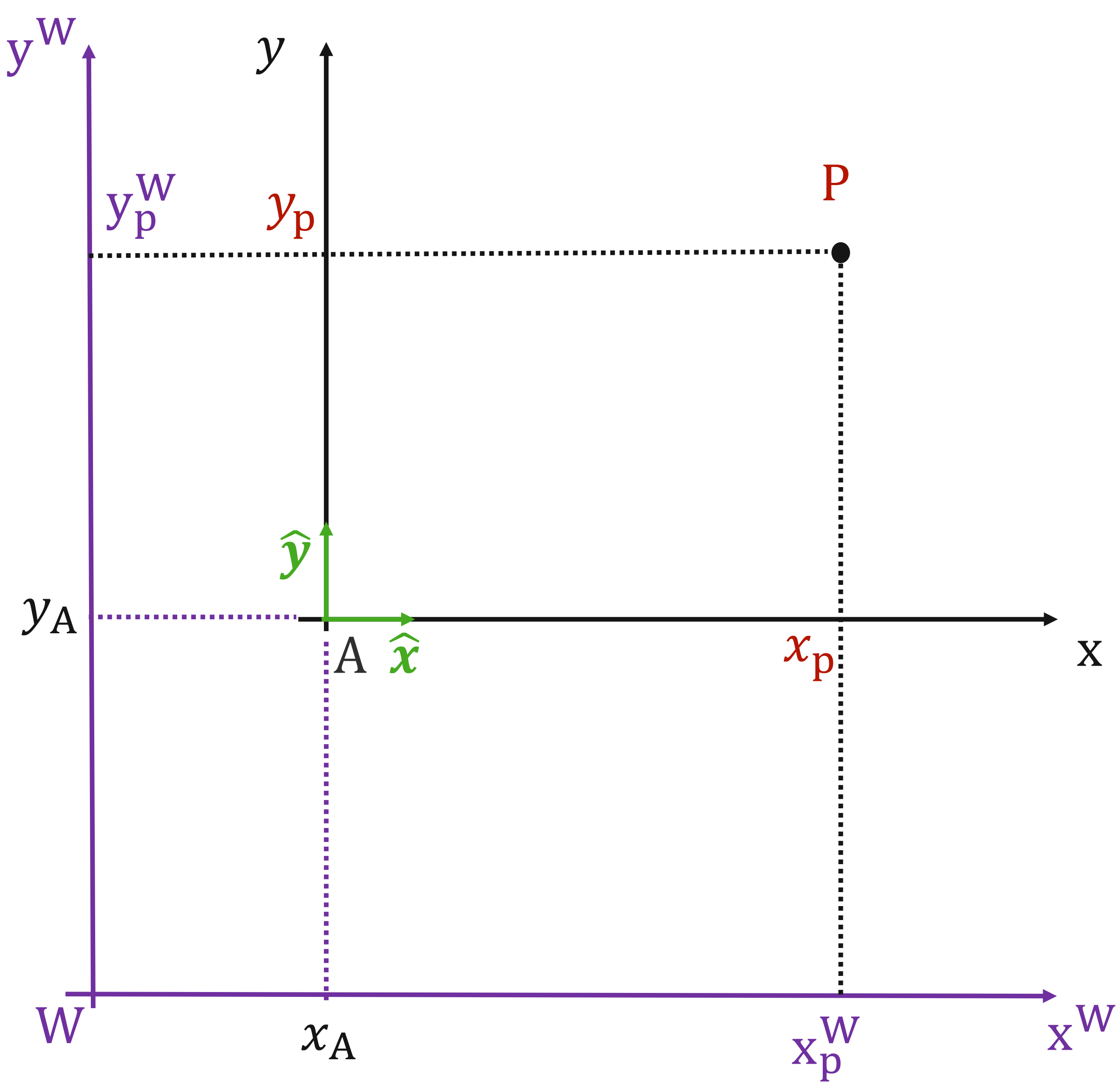
$$\mathrm{P}^{\mathrm{A}} = (\mathrm{x}^{\mathrm{A}}, \mathrm{y}^{\mathrm{A}}) = x_p\hat{\boldsymbol{x}} + y_p\hat{\boldsymbol{y}} = \begin{bmatrix} \mathrm{x_p} \\ \mathrm{y_p} \end{bmatrix}$$

- $P \in \mathbb{R}^2$ with respect to world frame:

$$\mathrm{P}^{\mathrm{W}} = (\mathrm{x}^{\mathrm{W}}, \mathrm{y}^{\mathrm{W}}) = \begin{bmatrix} \mathrm{x_p^W} \\ \mathrm{y_p^W} \end{bmatrix} = \begin{bmatrix} \mathrm{x_p} \\ \mathrm{y_p} \end{bmatrix} + \begin{bmatrix} \mathrm{x_A} \\ \mathrm{y_A} \end{bmatrix}$$

- A trick to express *translation* as matrix multiplication:

$$\Rightarrow \mathrm{X}^{\mathrm{W}} := \begin{bmatrix} \mathrm{x_p^W} \\ \mathrm{y_p^W} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mathrm{x_A} \\ 0 & 1 & \mathrm{y_A} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{x_p} \\ \mathrm{y_p} \\ 1 \end{bmatrix} = \mathbf{T}^{\mathrm{A}}\mathrm{X}^{\mathrm{A}}$$



Polar Coordinates in 2D

- **Cartesian** *representation* of point $P \in \mathbb{R}^2$ with respect to reference with origin in A:

$$P^A = \begin{bmatrix} x_p \\ y_p \end{bmatrix}$$

- **Polar** *representation* as a complex number:

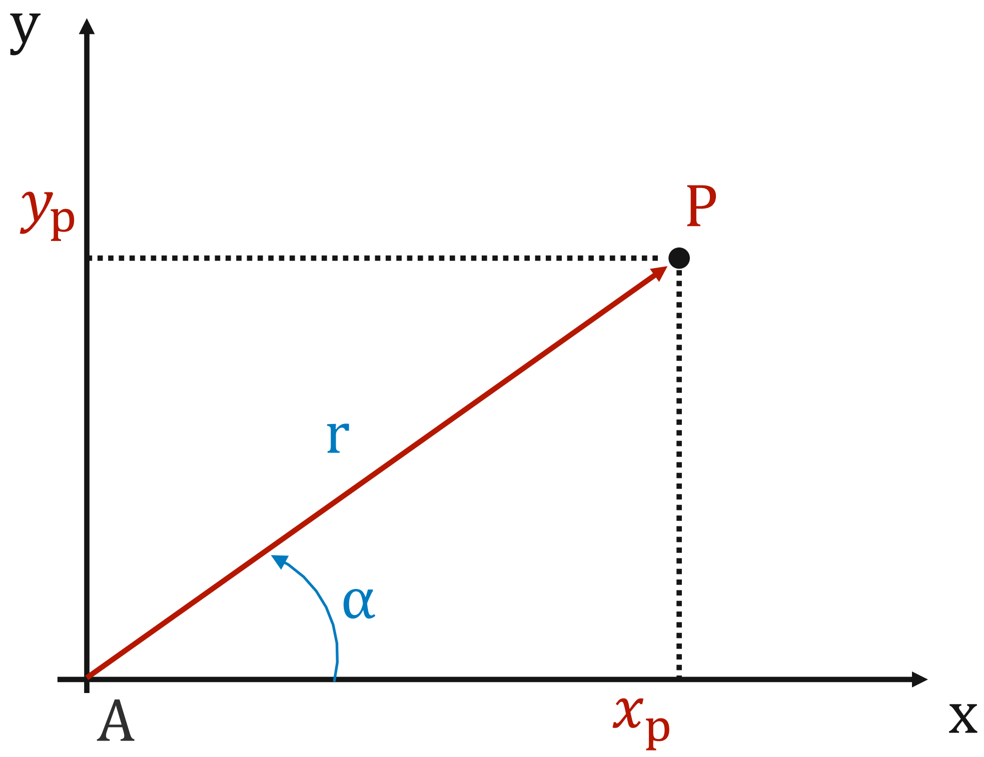
$$P^A = \mathbf{r} = re^{j\alpha} = (r, \alpha)$$

$$x_p = r\cos\alpha$$

$$y_p = r\sin\alpha$$

$$r = \sqrt{x_p^2 + y_p^2}$$

$$\alpha = \operatorname{atan}\left(\frac{y_p}{x_p}\right)$$



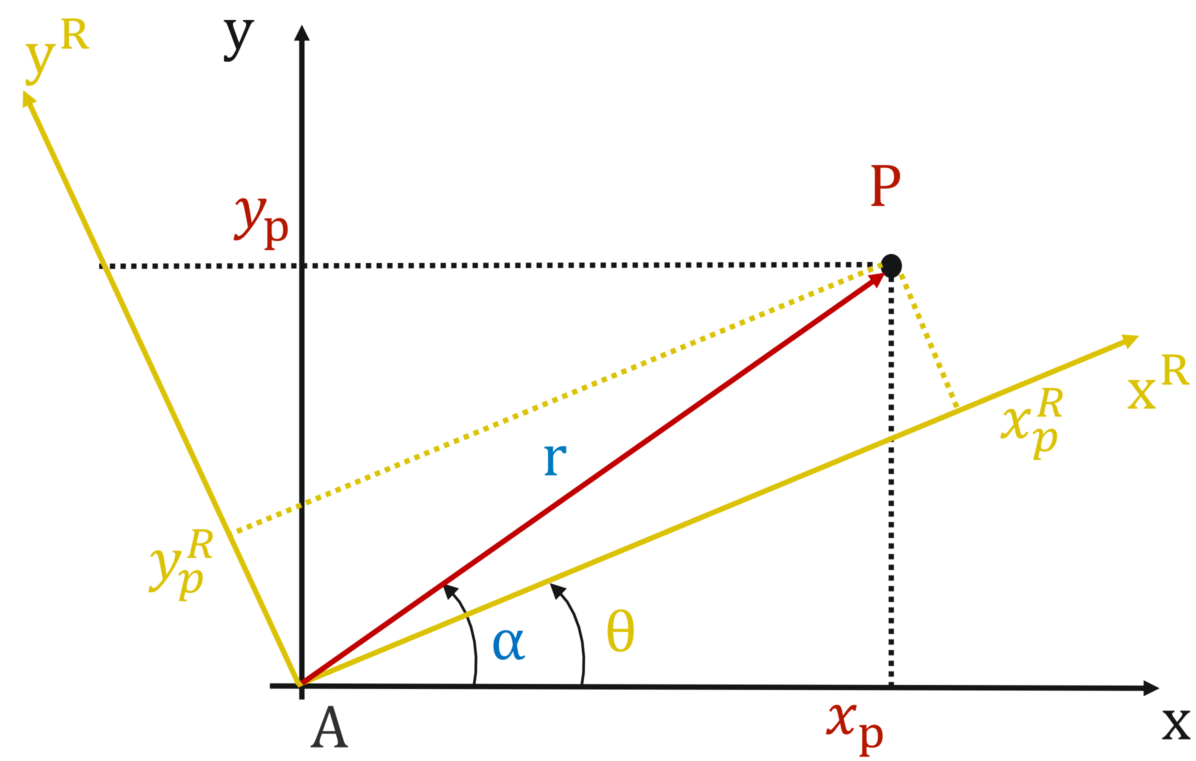
Rotations 2D

- Polar representation with respect to reference with origin in A:

$$P^A = \mathbf{r} = re^{j\alpha} = (r, \alpha)$$

$$x_p = r\cos\alpha$$

$$y_p = r\sin\alpha$$



- Polar representation with respect to reference with origin in A, but rotated of θ:

$$P^R = re^{j(\alpha-\theta)} = (r, \alpha - \theta)$$

$$x_p^R = r\cos(\alpha - \theta) = x_p\cos\theta + y_p\sin\theta$$

$$y_p^R = r\sin(\alpha - \theta) = -x_p\sin\theta + y_p\cos\theta$$

- Rotation:

$$\begin{bmatrix} x_p^R \\ y_p^R \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix}$$

Rotation matrix is orthogonal

Special Orthogonal group in 2d

$$\mathbf{M}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \in SO(2)$$

$$\Rightarrow \mathbf{M}^T(\theta) = \mathbf{M}^{-1}(\theta)$$

$$\det M(\theta) = 1, \forall\, \theta$$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_p^R \\ y_p^R \end{bmatrix}$$

Moving between frames in 2D

- Rotations (Cartesian representation):

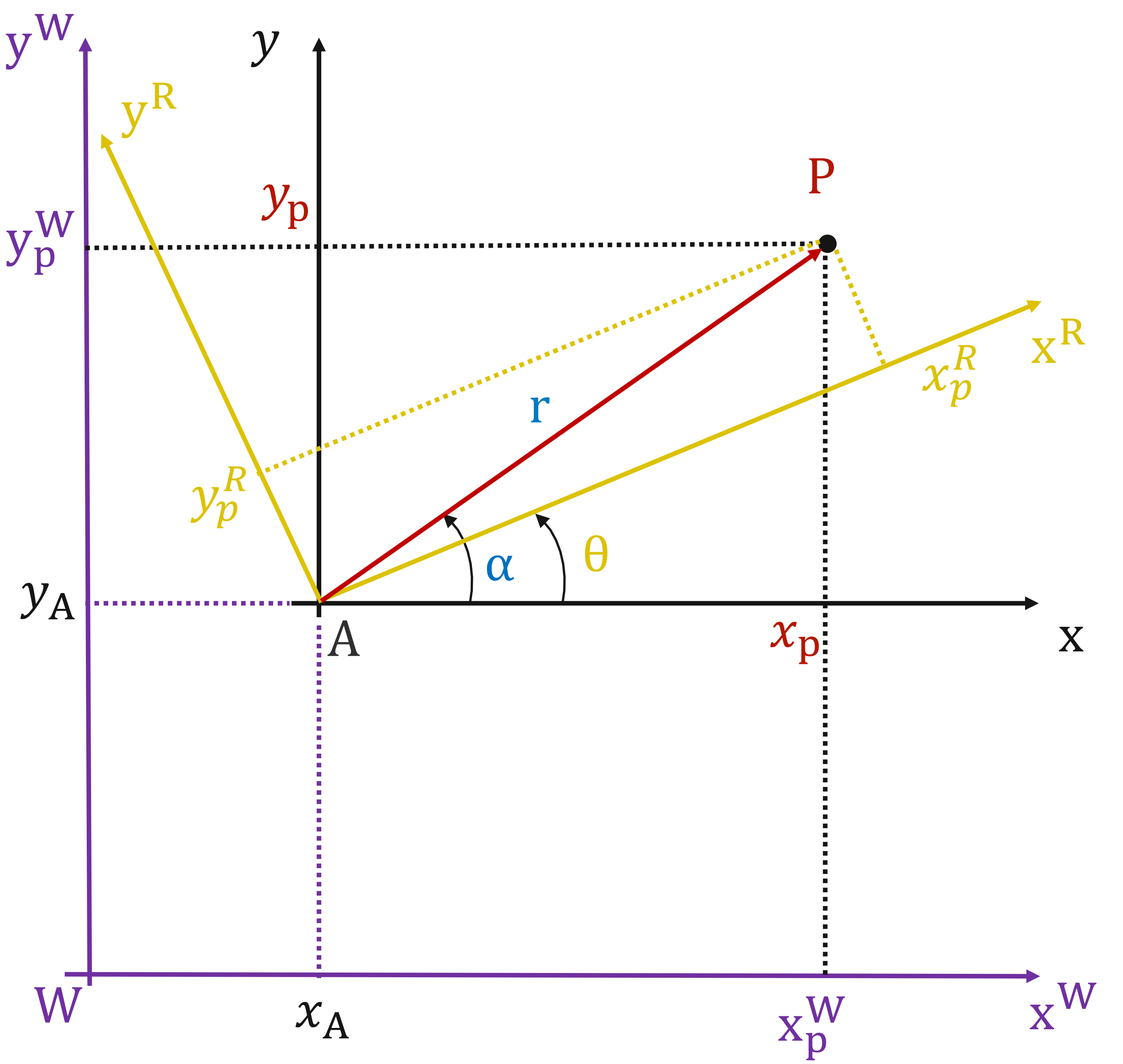
$$P^A = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p^R \\ y_p^R \\ 1 \end{bmatrix} = \mathbf{R}(\theta) P^R$$

- Translations:

$$P^W := \begin{bmatrix} x_p^W \\ y_p^W \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_A \\ 0 & 1 & y_A \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \mathbf{T} P^A$$

- Rotations and Translations together:

$$P^W = \mathbf{T} P^A = \mathbf{T}\mathbf{R}(\theta) P^R = \begin{bmatrix} \cos\theta & -\sin\theta & x_a \\ \sin\theta & \cos\theta & y_a \\ 0 & 0 & 1 \end{bmatrix} P^R$$

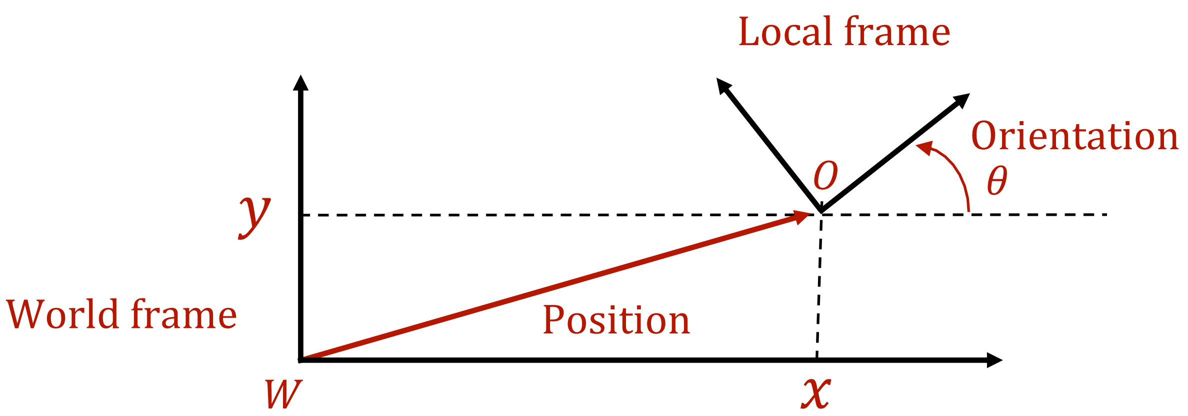


We can now express coordinates of points in roto-translated reference systems w.r.t. to a "world" reference

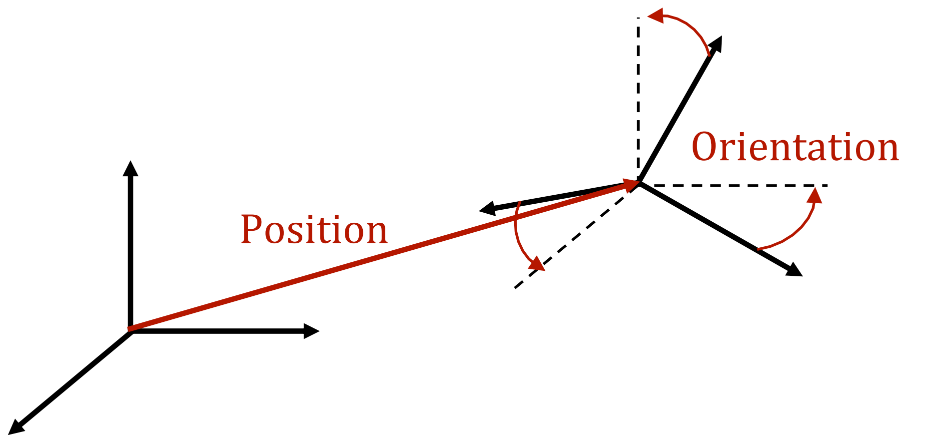
3D vs 2D

- What is "**pose**"? It is position and orientation of something (robot, sensor, duckie, ...)

- In **2D**: 3 dof (degrees of freedom)
  - position (x, y): 2 dof
  - orientation (θ): 1 dof

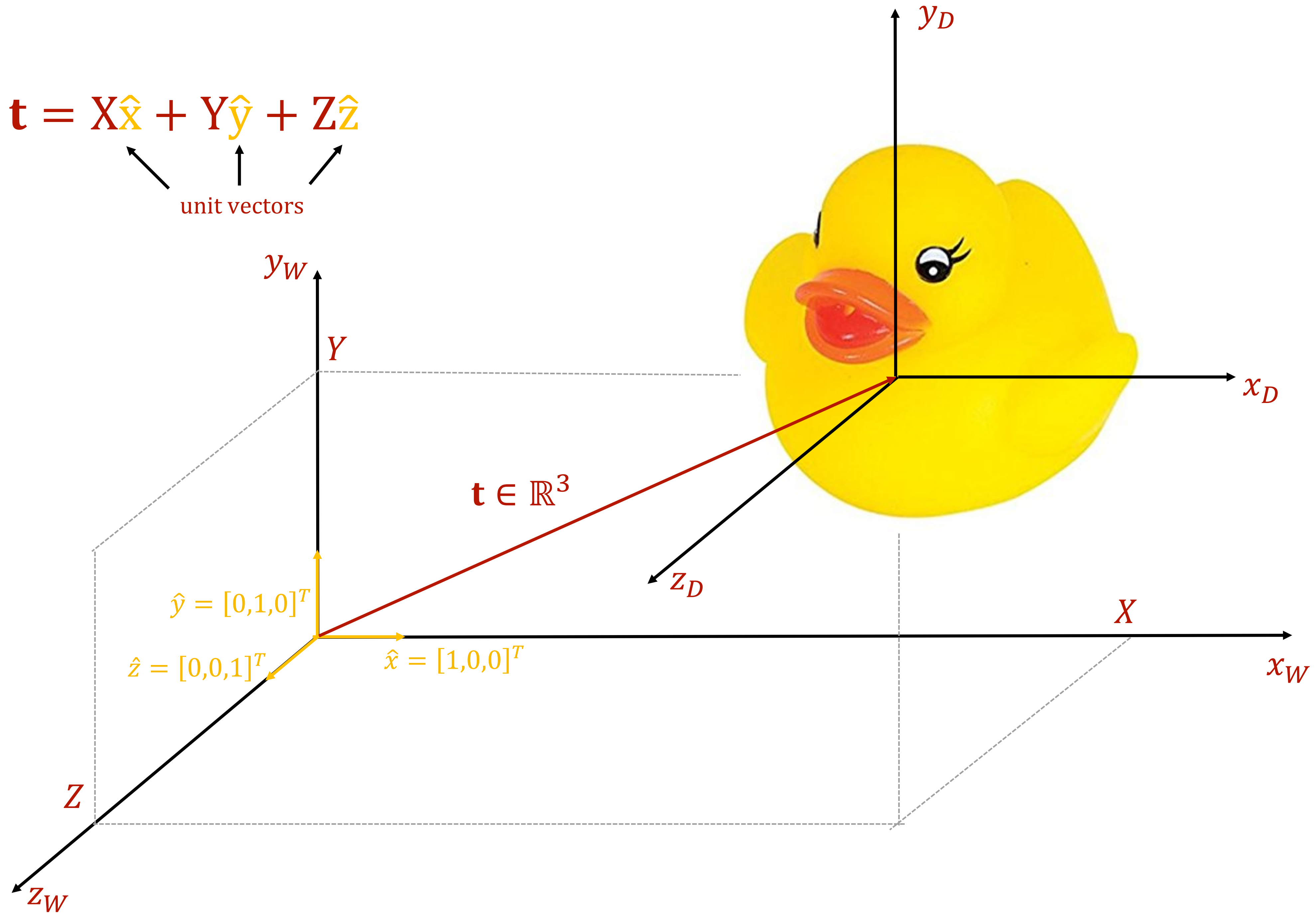


- In **3D**: 6 dof
  - position (x, y, z): 3 dof
  - orientation $(\theta_x, \theta_y, \theta_z)$: 3 dof



Translations 3D

$$\mathbf{t} = X\hat{x} + Y\hat{y} + Z\hat{z}$$



3D Rotations

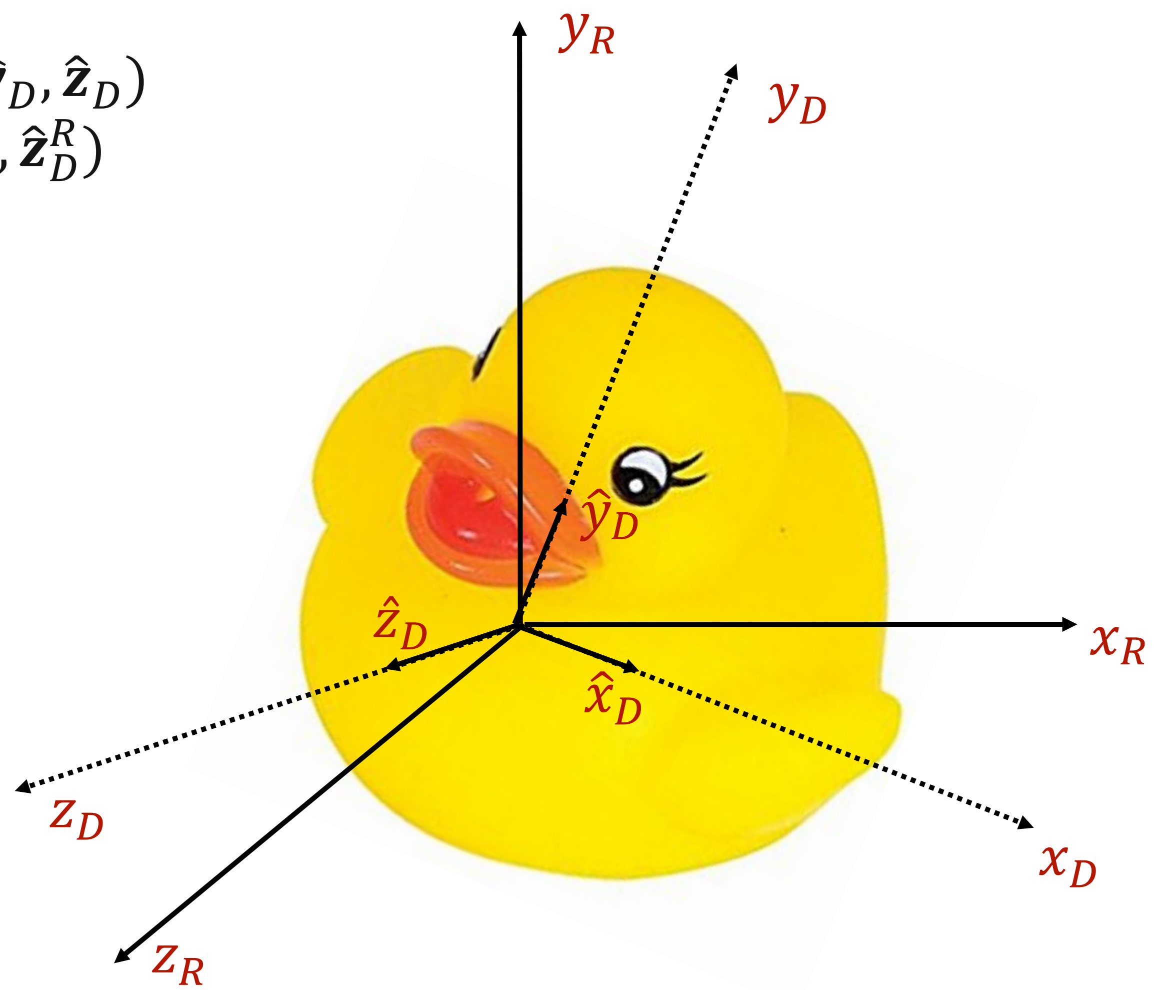
- Express unit vectors of rotated frame $(\hat{\boldsymbol{x}}_D, \hat{\boldsymbol{y}}_D, \hat{\boldsymbol{z}}_D)$ in terms of reference frame. Obtain $(\hat{\boldsymbol{x}}_D^R, \hat{\boldsymbol{y}}_D^R, \hat{\boldsymbol{z}}_D^R)$

Rotation matrix

$$\mathbf{R} = [\hat{\mathbf{x}}_\mathrm{D}^\mathrm{R}, \hat{\mathbf{y}}_\mathrm{D}^\mathrm{R}, \hat{\mathbf{z}}_\mathrm{D}^\mathrm{R}] \in \mathbf{SO(3)} \subset \mathbb{R}^3$$

$$\mathrm{SO}(3) = \{\mathbf{R} \in \mathbb{R}^3 : \mathbf{R}^\mathrm{T}\mathbf{R} = \mathbf{I}, |\mathbf{R}| = 1\}$$

Special Orthogonal group in 3d

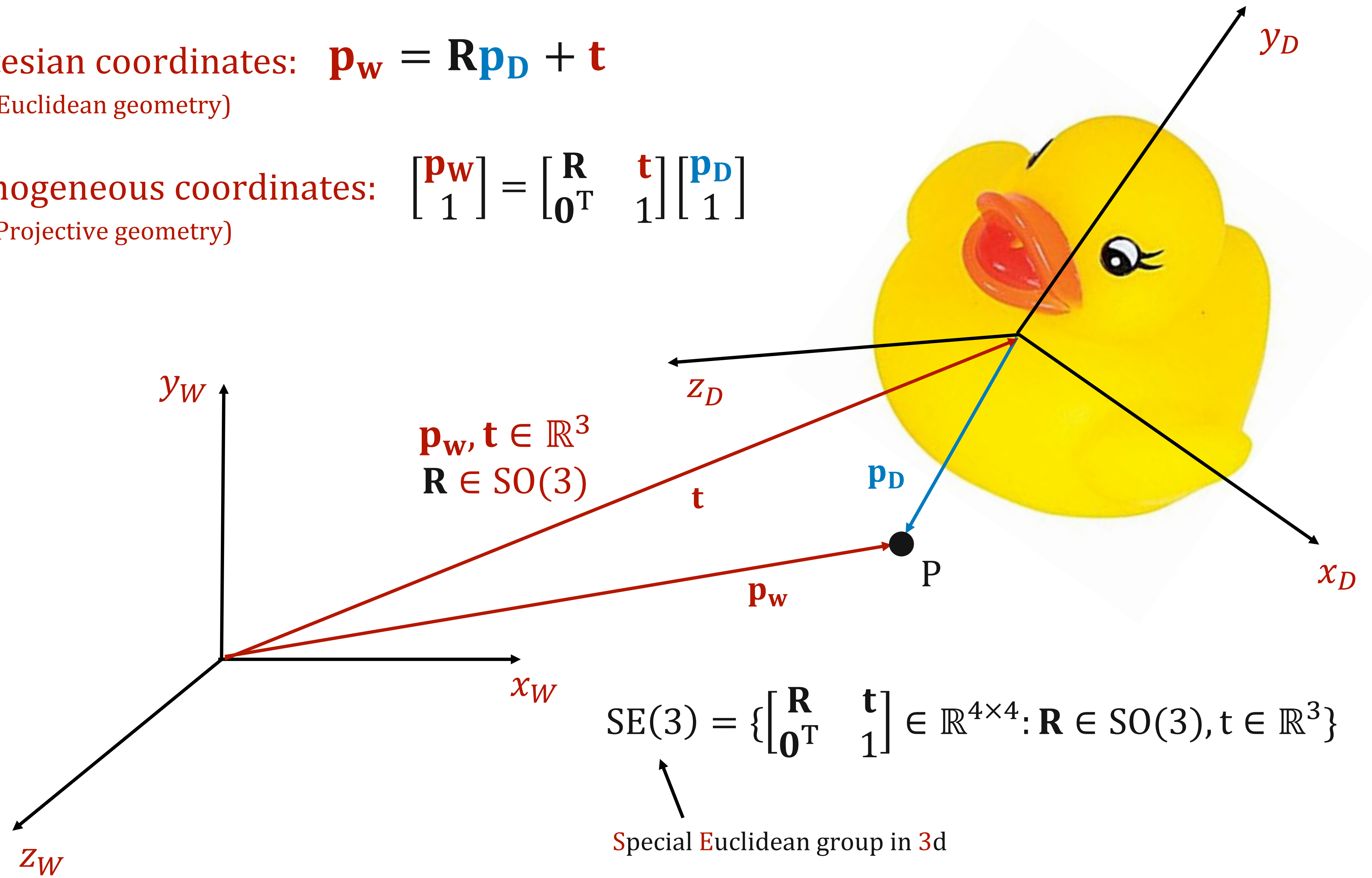


There are many equivalent representations of **R**, e.g.: cosine direction, Euler angles, quaternions

Moving between frames in 3D

How to express both translation and rotation in a unified framework?

- Cartesian coordinates: $\mathbf{p_W} = \mathbf{R}\mathbf{p_D} + \mathbf{t}$
  (Euclidean geometry)
- Homogeneous coordinates: $\begin{bmatrix} \mathbf{p_W} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p_D} \\ 1 \end{bmatrix}$
  (Projective geometry)



$$\mathrm{SE}(3) = \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} : \mathbf{R} \in \mathrm{SO}(3), \mathrm{t} \in \mathbb{R}^3 \right\}$$

Special Euclidean group in 3d

Summary

- Representations are important for building autonomy architectures
- States variables such as the pose (position and orientation) can be used for describing the robot's relation to the world
- Pose can be expressed in different reference frames (world, "body")
- We looked at how to move between reference frames that are rotated and translated w.r.t. each other
- We looked at how to express these transitions in an efficient way (homogeneous coordinates)

👣 Try it yourself!

```
dts exercises init
cd mooc-exercises/representations
dts exercises notebooks
```

Tutorial credits to Prof. Liam Paull, University of Montreal