



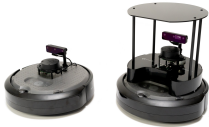
Robotics: Rigid body motion

Vladimír Petřík

vladimir.petrík@cvut.cz

22.09.2025

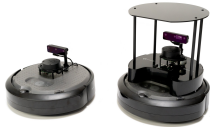
What is robot?



Mobilní robot, UGV -
unmanned ground vehicle



What is robot?

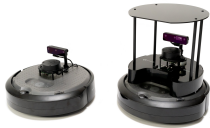


Mobilní robot, UGV -
unmanned ground vehicle



Flying robots (e.g. drones)

What is robot?



Mobilní robot, UGV -
unmanned ground vehicle

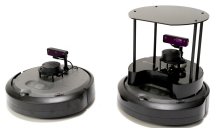


Flying robots (e.g. drones)



Walking robots
(e.g. humanoids)

What is robot?



Mobilní robot, UGV -
unmanned ground vehicle



Flying robots (e.g. drones)



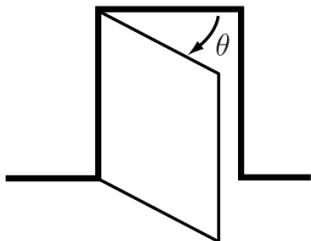
Walking robots
(e.g. humanoids)



Manipulators (např. Franka
Emika Panda)

Robot configuration

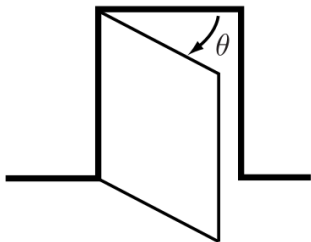
- Complete specification of the position of every point of the robot.



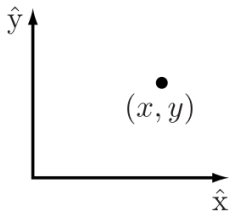
The configuration is described by the angle θ .

Robot configuration

- Complete specification of the position of every point of the robot.



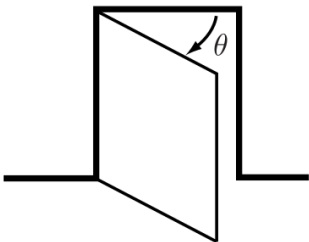
The configuration is described by the angle θ .



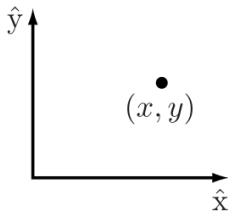
Point in plane is described by two coordinates.

Robot configuration

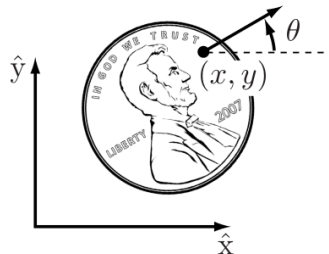
- Complete specification of the position of every point of the robot.



The configuration is described by the angle θ .



Point in plane is described by two coordinates.



Planar rigid object configuration consists of the position and orientation.

Degrees of freedom (DoF)

- ▶ The minimum number of real-valued coordinates needed to represent the configuration.
 - ▶ door: 1
 - ▶ planar point: 2
 - ▶ planar rigid object: 3
 - ▶ manipulators: from 1 (e.g. rotating table) to tens (e.g. humanoids)



Degrees of freedom (DoF)

- ▶ The minimum number of real-valued coordinates needed to represent the configuration.
 - ▶ door: 1
 - ▶ planar point: 2
 - ▶ planar rigid object: 3
 - ▶ manipulators: from 1 (e.g. rotating table) to tens (e.g. humanoids)
- ▶ Determining DoF
 - ▶ (sum of freedom of the points) - (number of independent constraints)
 - ▶ Rigid objects - how it is defined?



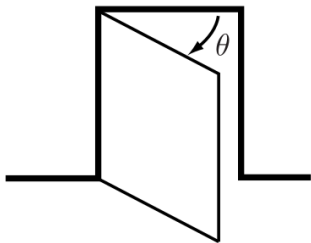
Degrees of freedom (DoF)

- ▶ The minimum number of real-valued coordinates needed to represent the configuration.
 - ▶ door: 1
 - ▶ planar point: 2
 - ▶ planar rigid object: 3
 - ▶ manipulators: from 1 (e.g. rotating table) to tens (e.g. humanoids)
- ▶ Determining DoF
 - ▶ (sum of freedom of the points) - (number of independent constraints)
 - ▶ Rigid objects - how it is defined?
 - ▶ The distance between any two given points on a rigid body remains constant
 - ▶ Exercise: write constraints for N points of planar rigid object
 - ▶ For some robots, determining number of DoF is non-trivial



Configuration space - $\mathcal{C}$

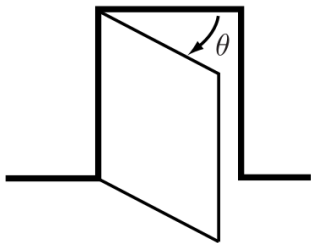
- ▶ The N -dimensional space (N correspond to number of DoF)
- ▶ Every point of configuration space correspond to one configuration
- ▶ Contains all possible configurations of the robot



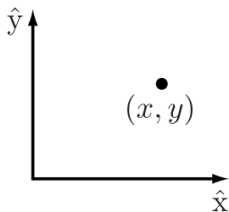
$$\mathcal{C} : \langle 0^\circ, 180^\circ \rangle \text{ or}$$
$$\mathcal{C} : \langle -180^\circ, 180^\circ \rangle$$

Configuration space - $\mathcal{C}$

- ▶ The N -dimensional space (N correspond to number of DoF)
- ▶ Every point of configuration space correspond to one configuration
- ▶ Contains all possible configurations of the robot



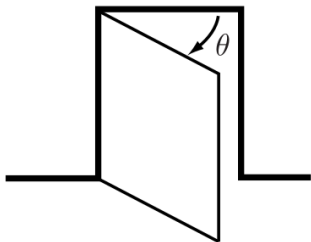
$$\mathcal{C} : \langle 0^\circ, 180^\circ \rangle \text{ or } \mathcal{C} : \langle -180^\circ, 180^\circ \rangle$$



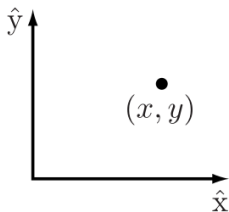
$$\mathcal{C} : \mathbb{R}^2$$

Configuration space - $\mathcal{C}$

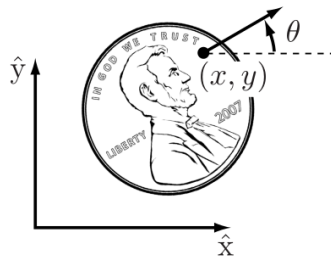
- ▶ The N -dimensional space (N correspond to number of DoF)
- ▶ Every point of configuration space correspond to one configuration
- ▶ Contains all possible configurations of the robot



$$\mathcal{C} : \langle 0^\circ, 180^\circ \rangle \text{ or } \mathcal{C} : \langle -180^\circ, 180^\circ \rangle$$



$$\mathcal{C} : \mathbb{R}^2$$

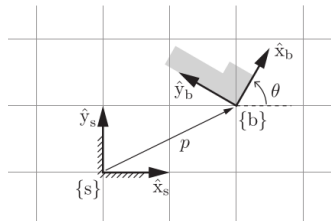


$$\mathcal{C} : \mathbb{R}^2 \times \langle 0^\circ, 360^\circ \rangle \text{ why not } \mathbb{R}^3?$$



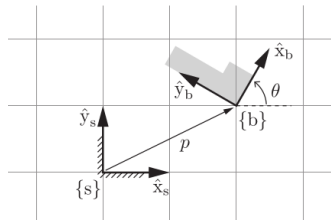
Rigid body motion in plane

- ▶ We attach a **body** frame to rigid body
 - ▶ Usually placed in the center of mass (but not required)
 - ▶ Can be placed outside of the body
 - ▶ Body frame is not moving w.r.t. to the body



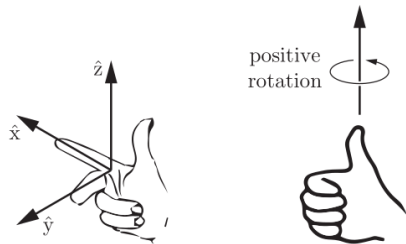
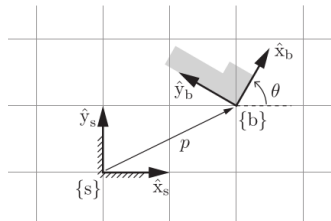
Rigid body motion in plane

- ▶ We attach a **body** frame to rigid body
 - ▶ Usually placed in the center of mass (but not required)
 - ▶ Can be placed outside of the body
 - ▶ Body frame is not moving w.r.t. to the body
- ▶ We select a fixed **reference** frame
 - ▶ center of the room
 - ▶ corner of the table
 - ▶ base of the manipulator



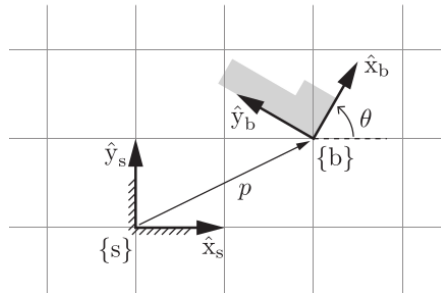
Rigid body motion in plane

- ▶ We attach a **body** frame to rigid body
 - ▶ Usually placed in the center of mass (but not required)
 - ▶ Can be placed outside of the body
 - ▶ Body frame is not moving w.r.t. to the body
- ▶ We select a fixed **reference** frame
 - ▶ center of the room
 - ▶ corner of the table
 - ▶ base of the manipulator
- ▶ All frames are right-handed



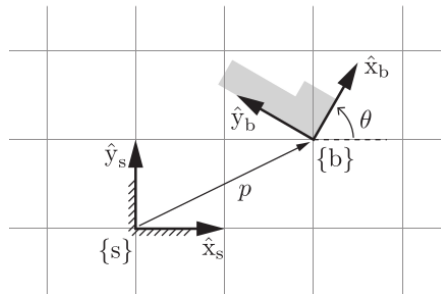
Rigid body motion in plane

- ▶ The configuration of body is given by
 - ▶ position of body frame w.r.t. reference frame
 - ▶ orientation of body frame w.r.t. reference frame



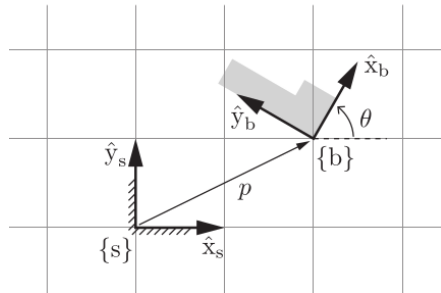
Rigid body motion in plane

- ▶ The configuration of body is given by
 - ▶ position of body frame w.r.t. reference frame
 - ▶ orientation of body frame w.r.t. reference frame
- ▶ Body frame origin
 - ▶ $\mathbf{p} = p_x \hat{\mathbf{x}}_s + p_y \hat{\mathbf{y}}_s \in \mathbb{R}^2$
 - ▶ If reference frame is clear from the context: $\mathbf{p} = (p_x, p_y)^\top$



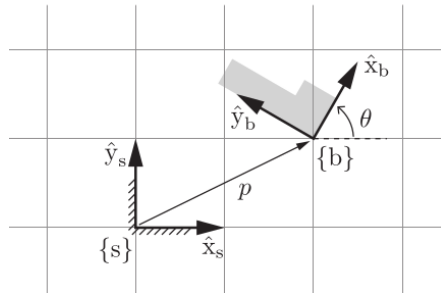
Rigid body motion in plane

- ▶ The configuration of body is given by
 - ▶ position of body frame w.r.t. reference frame
 - ▶ orientation of body frame w.r.t. reference frame
- ▶ Body frame origin
 - ▶ $\mathbf{p} = p_x \hat{\mathbf{x}}_s + p_y \hat{\mathbf{y}}_s \in \mathbb{R}^2$
 - ▶ If reference frame is clear from the context: $\mathbf{p} = (p_x, p_y)^\top$
- ▶ Orientation
 - ▶ Angle $\theta \in \langle 0^\circ, 360^\circ \rangle$



Rigid body motion in plane

- ▶ The configuration of body is given by
 - ▶ position of body frame w.r.t. reference frame
 - ▶ orientation of body frame w.r.t. reference frame
- ▶ Body frame origin
 - ▶ $\mathbf{p} = p_x \hat{\mathbf{x}}_s + p_y \hat{\mathbf{y}}_s \in \mathbb{R}^2$
 - ▶ If reference frame is clear from the context: $\mathbf{p} = (p_x, p_y)^\top$
- ▶ Orientation
 - ▶ Angle $\theta \in \langle 0^\circ, 360^\circ \rangle$
 - ▶ Convenient for next computations:
 $\hat{\mathbf{x}}_b = +\cos \theta \hat{\mathbf{x}}_s + \sin \theta \hat{\mathbf{y}}_s$
 $\hat{\mathbf{y}}_b = -\sin \theta \hat{\mathbf{x}}_s + \cos \theta \hat{\mathbf{y}}_s$
Rotation matrix $R = (\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$



$SO(2)$

- ▶ R has 4 numbers but only 1 DoF - 3 independent constraints
 - ▶ both columns are unit vectors
 - ▶ columns are orthogonal to each other



$SO(2)$

- ▶ R has 4 numbers but only 1 DoF - 3 independent constraints
 - ▶ both columns are unit vectors
 - ▶ columns are orthogonal to each other
- ▶ Set of all rotation matrix is $SO(2)$ group, i.e. $R \in SO(2)$
 - ▶ Special Orthogonal group
 - ▶ $\det(R) = 1$
 - ▶ $RR^T = I$, i.e. $R^{-1} = R^T$
 - ▶ $(R_1 R_2) R_3 = R_1 (R_2 R_3)$
 - ▶ $R_1 R_2 = R_2 R_1$



$SO(2)$

- ▶ R has 4 numbers but only 1 DoF - 3 independent constraints
 - ▶ both columns are unit vectors
 - ▶ columns are orthogonal to each other
- ▶ Set of all rotation matrix is $SO(2)$ group, i.e. $R \in SO(2)$
 - ▶ Special Orthogonal group
 - ▶ $\det(R) = 1$
 - ▶ $RR^T = I$, i.e. $R^{-1} = R^T$
 - ▶ $(R_1 R_2) R_3 = R_1 (R_2 R_3)$
 - ▶ **For $SO(2)$** $R_1 R_2 = R_2 R_1$



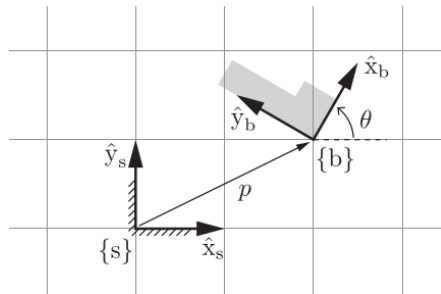
$SO(2)$

- ▶ R has 4 numbers but only 1 DoF - 3 independent constraints
 - ▶ both columns are unit vectors
 - ▶ columns are orthogonal to each other
- ▶ Set of all rotation matrix is $SO(2)$ group, i.e. $R \in SO(2)$
 - ▶ Special Orthogonal group
 - ▶ $\det(R) = 1$
 - ▶ $RR^T = I$, i.e. $R^{-1} = R^T$
 - ▶ $(R_1 R_2) R_3 = R_1 (R_2 R_3)$
 - ▶ **For $SO(2)$** $R_1 R_2 = R_2 R_1$
- ▶ Usage of rotation matrix
 - ▶ to represent an orientation of the frame
 - ▶ to change the reference frame in which a vector is represented
 - ▶ to rotate vector/frame



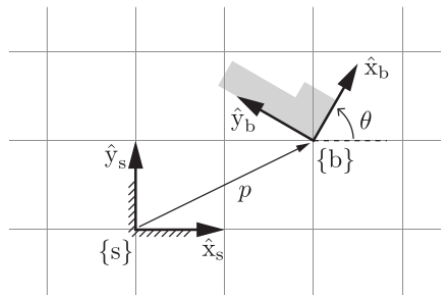
$SE(2)$

- ▶ A pair $(R_{ab}, \mathbf{p})$
 - ▶ represents pose/configuration of the body



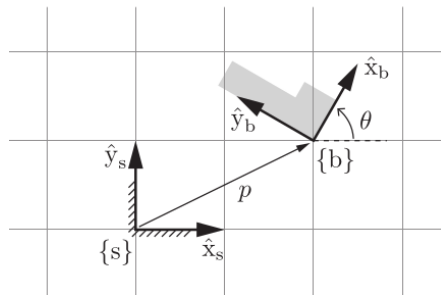
$SE(2)$

- ▶ A pair $(R_{ab}, \mathbf{p})$
 - ▶ represents pose/configuration of the body
 - ▶ changes the reference frame of a vector
- $$\mathbf{v}_a = R_{ab}\mathbf{v}_b + \mathbf{p}$$



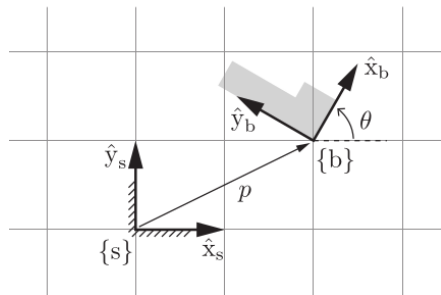
$SE(2)$

- ▶ A pair $(R_{ab}, \mathbf{p})$
 - ▶ represents pose/configuration of the body
 - ▶ changes the reference frame of a vector
$$\mathbf{v}_a = R_{ab}\mathbf{v}_b + \mathbf{p}$$
 - ▶ moves vector/frame $(R, \mathbf{t})$
$$\mathbf{R}_{\text{moved}} = R_{ab}R \quad \mathbf{t}_{\text{moved}} = R_{ab}\mathbf{t} + \mathbf{p}$$



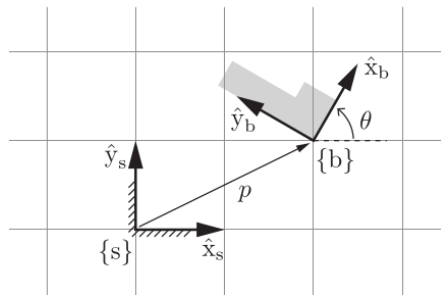
$SE(2)$

- ▶ A pair $(R_{ab}, \mathbf{p})$
 - ▶ represents pose/configuration of the body
 - ▶ changes the reference frame of a vector
$$\mathbf{v}_a = R_{ab}\mathbf{v}_b + \mathbf{p}$$
 - ▶ moves vector/frame $(R, \mathbf{t})$
$$\mathbf{R}_{\text{moved}} = R_{ab}R \quad \mathbf{t}_{\text{moved}} = R_{ab}\mathbf{t} + \mathbf{p}$$
- ▶ Alternatively, in homogeneous coordinates $T_{ab} = \begin{pmatrix} R_{ab} & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{pmatrix} \in SE(2)$
 - ▶ Special Euclidean Group
 - ▶ represents both translation and rotation in a single matrix
 - ▶ $\mathbf{v}_a^H = T_{ab}\mathbf{v}_b^H$
 - ▶ $(T_1T_2)T_3 = T_1(T_2T_3)$
 - ▶ $T_1T_2 \neq T_2T_1$



$SE(2)$

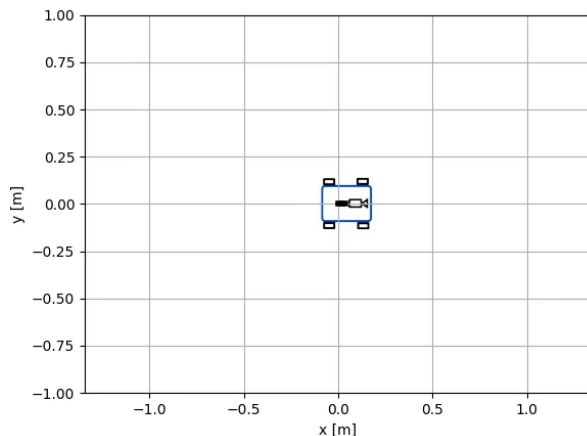
- ▶ A pair $(R_{ab}, \mathbf{p})$
 - ▶ represents pose/configuration of the body
 - ▶ changes the reference frame of a vector
$$\mathbf{v}_a = R_{ab}\mathbf{v}_b + \mathbf{p}$$
 - ▶ moves vector/frame $(R, \mathbf{t})$
$$\mathbf{R}_{\text{moved}} = R_{ab}R \quad \mathbf{t}_{\text{moved}} = R_{ab}\mathbf{t} + \mathbf{p}$$
- ▶ Alternatively, in homogeneous coordinates $T_{ab} = \begin{pmatrix} R_{ab} & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{pmatrix} \in SE(2)$
 - ▶ Special Euclidean Group
 - ▶ represents both translation and rotation in a single matrix
 - ▶ $\mathbf{v}_a^H = T_{ab}\mathbf{v}_b^H$
 - ▶ $(T_1T_2)T_3 = T_1(T_2T_3)$
 - ▶ $T_1T_2 \neq T_2T_1$
 - ▶ Inverse T^{-1}
 - ▶ computing inverse of a matrix is costly
 - ▶ $T^{-1} = \begin{pmatrix} R^\top & -R^\top \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}$



$SE(2)$ example

$$T_{\text{next}} = T_{\text{current}} T_x(\delta_x) \quad T_{\text{next}} = T_{\text{current}} T_\theta(\delta_\theta) \quad T_{\text{next}} = T_{\text{current}} T_x(\delta_x)$$

Delta transformations are defined in robot frame.



$SE(2)$ example

$$T_{\text{next}} = T_x(\delta_x)T_{\text{current}} \quad T_{\text{next}} = T_\theta(\delta_\theta)T_{\text{current}} \quad T_{\text{next}} = T_x(\delta_x)T_{\text{current}}$$

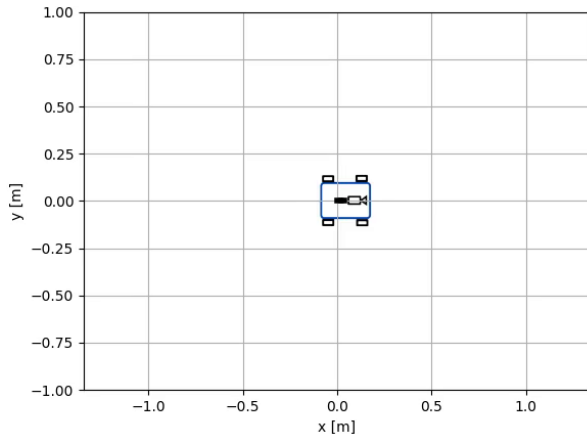
Delta transformations are defined in reference frame.



$SE(2)$ example

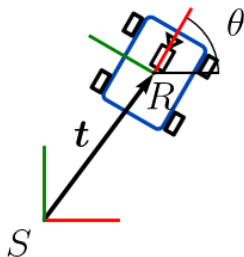
$$T_{\text{next}} = T_x(\delta_x)T_{\text{current}} \quad T_{\text{next}} = T_\theta(\delta_\theta)T_{\text{current}} \quad T_{\text{next}} = T_x(\delta_x)T_{\text{current}}$$

Delta transformations are defined in reference frame.



$SE(2)$ example camera

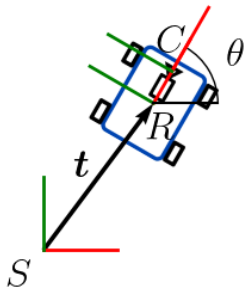
$$T_{SR} = \begin{pmatrix} R(\theta) & t \\ \mathbf{0}^\top & 1 \end{pmatrix}$$



$SE(2)$ example camera

$$T_{SR} = \begin{pmatrix} R(\theta) & t \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

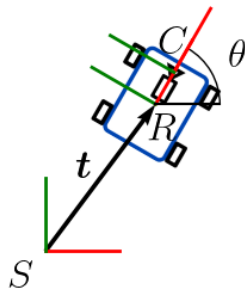
$$T_{RC} = \begin{pmatrix} I & (0.1 \ 0)^\top \\ \mathbf{0}^\top & 1 \end{pmatrix}$$



$SE(2)$ example camera

$$T_{SR} = \begin{pmatrix} R(\theta) & t \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

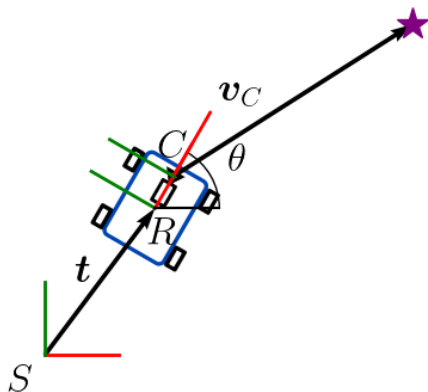
$$T_{RC} = \begin{pmatrix} I & (0.1 \ 0)^\top \\ \mathbf{0}^\top & 1 \end{pmatrix}$$



$SE(2)$ example camera

$$T_{SR} = \begin{pmatrix} R(\theta) & t \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

$$T_{RC} = \begin{pmatrix} I & (0.1 \ 0)^\top \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

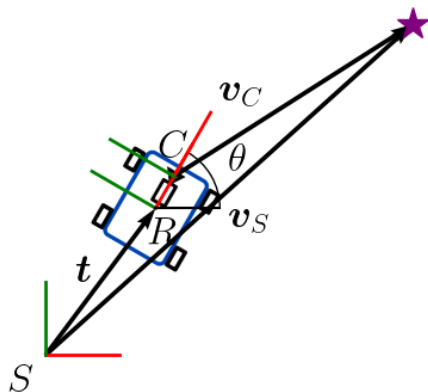


$SE(2)$ example camera

$$T_{SR} = \begin{pmatrix} R(\theta) & t \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

$$T_{RC} = \begin{pmatrix} I & (0.1 \ 0)^\top \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

How to compute v_S ?



$SE(2)$ example camera

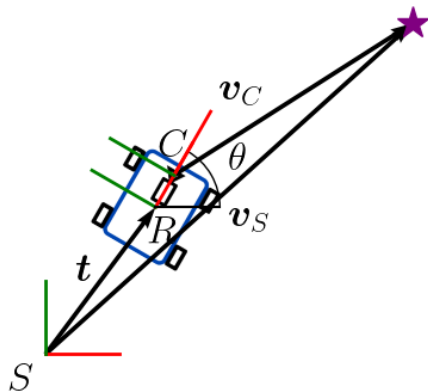
$$T_{SR} = \begin{pmatrix} R(\theta) & t \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

$$T_{RC} = \begin{pmatrix} I & (0.1 \ 0)^\top \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

How to compute $\mathbf{v}_S$?

$$T_{SC} = T_{SR}T_{RC}$$

$$\mathbf{v}_S = T_{SC}\mathbf{v}_C$$



Extending to $SO(3)$ and $SE(3)$

► $SO(3)$

- $\det(R) = 1$
- $RR^\top = I$, i.e. $R^{-1} = R^\top$
- $(R_1 R_2) R_3 = R_1 (R_2 R_3)$
- $R_1 R_2 = R_2 R_1$



Extending to $SO(3)$ and $SE(3)$

► $SO(3)$

- $\det(R) = 1$
- $RR^T = I$, i.e. $R^{-1} = R^T$
- $(R_1 R_2) R_3 = R_1 (R_2 R_3)$
- $R_1 R_2 \neq R_2 R_1$ obecně



Extending to $SO(3)$ and $SE(3)$

► $SO(3)$

- $\det(R) = 1$
- $RR^\top = I$, i.e. $R^{-1} = R^\top$
- $(R_1 R_2) R_3 = R_1 (R_2 R_3)$
- $R_1 R_2 \neq R_2 R_1$ obecně

► $SE(3)$

- $\mathbf{v}_a^H = T_{ab} \mathbf{v}_b^H$
- $(T_1 T_2) T_3 = T_1 (T_2 T_3)$
- $T_1 T_2 \neq T_2 T_1$
- $T^{-1} = \begin{pmatrix} R^\top & -R^\top \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}$



How to compute $R \in SO(3)$?

- ▶ Composing rotations around the x, y, z axes

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

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

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

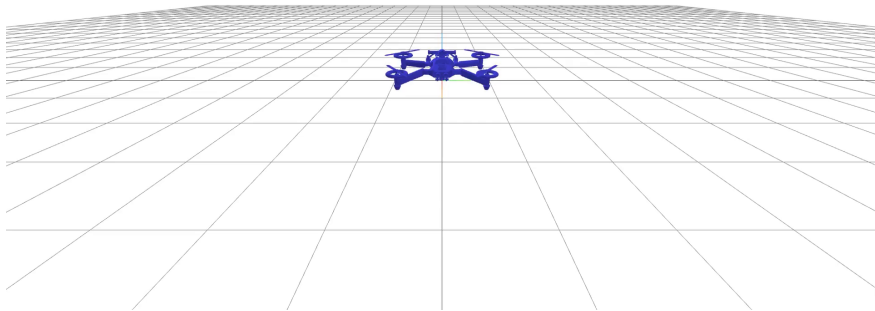
- ▶ From other representations of rotations



Example of $SE(3)$

$$T_{\text{next}} = TT_z(\delta_z) \quad T_{\text{next}} = TR_z(\theta_z) \quad T_{\text{next}} = TR_y(\theta_y) \quad T_{\text{next}} = TT_x(\delta_x)$$

$R_y, R_z \in SE(3)!$



Axis-angle representation

► $\theta \in \mathbb{R}, \quad \hat{\omega} \in \mathbb{R}^3, \quad \|\hat{\omega}\| = 1$



Axis-angle representation

- ▶ $\theta \in \mathbb{R}$, $\hat{\omega} \in \mathbb{R}^3$, $\|\hat{\omega}\| = 1$
- ▶ Axis-angle to R
 - ▶ Rodrigues' formula $R(\hat{\omega}, \theta) = I + \sin \theta [\hat{\omega}] + (1 - \cos \theta) [\hat{\omega}]^2$
 - ▶ Skew-symmetric matrix $[\omega] = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$
 - ▶ Example: compute R_z



Axis-angle representation

- ▶ $\theta \in \mathbb{R}$, $\hat{\omega} \in \mathbb{R}^3$, $\|\hat{\omega}\| = 1$
- ▶ Axis-angle to R
 - ▶ Rodrigues' formula $R(\hat{\omega}, \theta) = I + \sin \theta [\hat{\omega}] + (1 - \cos \theta) [\hat{\omega}]^2$
 - ▶ Skew-symmetric matrix $[\omega] = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$
 - ▶ Example: compute R_z
- ▶ Axis-angle from R algorithm
 - ▶ If $R = I$ then $\theta = 0$ and $\hat{\omega}$ is undefined.



Axis-angle representation

- ▶ $\theta \in \mathbb{R}$, $\hat{\omega} \in \mathbb{R}^3$, $\|\hat{\omega}\| = 1$
- ▶ Axis-angle to R
 - ▶ Rodrigues' formula $R(\hat{\omega}, \theta) = I + \sin \theta [\hat{\omega}] + (1 - \cos \theta) [\hat{\omega}]^2$
 - ▶ Skew-symmetric matrix $[\omega] = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$
 - ▶ Example: compute R_z
- ▶ Axis-angle from R algorithm
 - ▶ If $R = I$ then $\theta = 0$ and $\hat{\omega}$ is undefined.
 - ▶ If $\text{tr } R = -1$ then $\theta = \pi$ and
 - ▶ $\hat{\omega} = \frac{1}{\sqrt{2(1+r_{33})}} (r_{13} \quad r_{23} \quad 1 + r_{33})^\top$ if $r_{33} \neq -1$
 - ▶ $\hat{\omega} = \frac{1}{\sqrt{2(1+r_{22})}} (r_{12} \quad 1 + r_{22} \quad r_{32})^\top$ if $r_{22} \neq -1$
 - ▶ $\hat{\omega} = \frac{1}{\sqrt{2(1+r_{11})}} (1 + r_{11} \quad r_{21} \quad r_{31})^\top$ if $r_{11} \neq -1$



Axis-angle representation

- ▶ $\theta \in \mathbb{R}$, $\hat{\omega} \in \mathbb{R}^3$, $\|\hat{\omega}\| = 1$
- ▶ Axis-angle to R
 - ▶ Rodrigues' formula $R(\hat{\omega}, \theta) = I + \sin \theta [\hat{\omega}] + (1 - \cos \theta) [\hat{\omega}]^2$
 - ▶ Skew-symmetric matrix $[\omega] = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$
 - ▶ Example: compute R_z
- ▶ Axis-angle from R algorithm
 - ▶ If $R = I$ then $\theta = 0$ and $\hat{\omega}$ is undefined.
 - ▶ If $\text{tr } R = -1$ then $\theta = \pi$ and
 - ▶ $\hat{\omega} = \frac{1}{\sqrt{2(1+r_{33})}} (r_{13} \quad r_{23} \quad 1 + r_{33})^\top$ if $r_{33} \neq -1$
 - ▶ $\hat{\omega} = \frac{1}{\sqrt{2(1+r_{22})}} (r_{12} \quad 1 + r_{22} \quad r_{32})^\top$ if $r_{22} \neq -1$
 - ▶ $\hat{\omega} = \frac{1}{\sqrt{2(1+r_{11})}} (1 + r_{11} \quad r_{21} \quad r_{31})^\top$ if $r_{11} \neq -1$
 - ▶ Otherwise $\theta = \arccos(1/2(\text{tr } R - 1))$ and $[\hat{\omega}] = \frac{1}{2\sin \theta}(R - R^\top)$



Exponential coordinates

- ▶ A single vector $\omega \in \mathbb{R}^3$
- ▶ Also called Euler vector or Euler-Rodrigues parameters
- ▶ Mapping to angle-axis representation:
 - ▶ $\theta = \|\omega\|$
 - ▶ $\hat{\omega} = \frac{\omega}{\theta}$



Exponential coordinates

- ▶ A single vector $\omega \in \mathbb{R}^3$
- ▶ Also called Euler vector or Euler-Rodrigues parameters
- ▶ Mapping to angle-axis representation:
 - ▶ $\theta = \|\omega\|$
 - ▶ $\hat{\omega} = \frac{\omega}{\theta}$
- ▶ Exponential to/from R
 - ▶ $R = \exp \omega$: use Rodrigues' formula
 - ▶ $\omega = \log R$: use angle axis from R algorithm



Exponential coordinates

- ▶ A single vector $\omega \in \mathbb{R}^3$
- ▶ Also called Euler vector or Euler-Rodrigues parameters
- ▶ Mapping to angle-axis representation:
 - ▶ $\theta = \|\omega\|$
 - ▶ $\hat{\omega} = \frac{\omega}{\theta}$
- ▶ Exponential to/from R
 - ▶ $R = \exp \omega$: use Rodrigues' formula
 - ▶ $\omega = \log R$: use angle axis from R algorithm
- ▶ Why exponential?
 - ▶ it correspond to matrix exponential/logarithm of $[\omega]$
 - ▶ if ω is angular velocity, its integration for one unit of time leads to exponential and the final orientation is R
 - ▶ numerically sensitive to small angles



Quaternions

- ▶ $\mathbf{q} \in \mathbb{R}^4$, $\|\mathbf{q}\| = 1$
- ▶ From axis-angle
 - ▶ $q_w = \cos(\theta/2)$
 - ▶ $\mathbf{q}_{xyz} = \hat{\boldsymbol{\omega}} \sin(\theta/2)$



Quaternions

- ▶ $\mathbf{q} \in \mathbb{R}^4$, $\|\mathbf{q}\| = 1$
- ▶ From axis-angle
 - ▶ $q_w = \cos(\theta/2)$
 - ▶ $\mathbf{q}_{xyz} = \hat{\boldsymbol{\omega}} \sin(\theta/2)$
- ▶ From R
 - ▶ $q_w = 1/2\sqrt{1 + \text{tr } R}$
 - ▶ $\mathbf{q}_{xyz} = \frac{1}{4q_w} \begin{pmatrix} r_{32} - r_{23} & r_{13} - r_{31} & r_{21} - r_{12} \end{pmatrix}^\top$



Quaternions

- ▶ $\mathbf{q} \in \mathbb{R}^4$, $\|\mathbf{q}\| = 1$
- ▶ From axis-angle
 - ▶ $q_w = \cos(\theta/2)$
 - ▶ $\mathbf{q}_{xyz} = \hat{\boldsymbol{\omega}} \sin(\theta/2)$
- ▶ From R
 - ▶ $q_w = 1/2\sqrt{1 + \text{tr } R}$
 - ▶ $\mathbf{q}_{xyz} = \frac{1}{4q_w} \begin{pmatrix} r_{32} - r_{23} & r_{13} - r_{31} & r_{21} - r_{12} \end{pmatrix}^\top$
- ▶ To R
 - ▶ $R = \exp\left(2 \arccos(q_w) \frac{\mathbf{q}_{xyz}}{\|\mathbf{q}_{xyz}\|}\right)$
 - ▶ i.e. rotate about $\mathbf{q}_{xyz}$ with $\theta = 2 \arccos(q_w)$



Quaternions

- ▶ $\mathbf{q} \in \mathbb{R}^4$, $\|\mathbf{q}\| = 1$
- ▶ From axis-angle
 - ▶ $q_w = \cos(\theta/2)$
 - ▶ $\mathbf{q}_{xyz} = \hat{\boldsymbol{\omega}} \sin(\theta/2)$
- ▶ From R
 - ▶ $q_w = 1/2\sqrt{1 + \text{tr } R}$
 - ▶ $\mathbf{q}_{xyz} = \frac{1}{4q_w} \begin{pmatrix} r_{32} - r_{23} & r_{13} - r_{31} & r_{21} - r_{12} \end{pmatrix}^\top$
- ▶ To R
 - ▶ $R = \exp\left(2 \arccos(q_w) \frac{\mathbf{q}_{xyz}}{\|\mathbf{q}_{xyz}\|}\right)$
 - ▶ i.e. rotate about $\mathbf{q}_{xyz}$ with $\theta = 2 \arccos(q_w)$
- ▶ Quaternions are not unique, two solutions for the same R
- ▶ Numerically stable



Other representations

- ▶ Euler angles
 - ▶ three numbers $\theta_1, \theta_2, \theta_3$
 - ▶ rotation about the x , y , or z axes
 - ▶ e.g. XYX Euler angles correspond to $R = R_x(\theta_1)R_y(\theta_2)R_x(\theta_3)$
 - ▶ computing Euler angles from R is often numerically unstable and requires special algorithm for each triplet of axes



Other representations

- ▶ Euler angles
 - ▶ three numbers $\theta_1, \theta_2, \theta_3$
 - ▶ rotation about the x , y , or z axes
 - ▶ e.g. XYX Euler angles correspond to $R = R_x(\theta_1)R_y(\theta_2)R_x(\theta_3)$
 - ▶ computing Euler angles from R is often numerically unstable and requires special algorithm for each triplet of axes
- ▶ 6D representation of rotation
 - ▶ represented by the first two columns of R
 - ▶ smooth representation
 - ▶ used in machine-learning (e.g. output of neural network)



Summary

- ▶ Configuration, Configuration Space $\mathcal{C}$, DoF
- ▶ Planar rigid body motion $SO(2)$, $SE(2)$
- ▶ Spatial rigid body motion $SO(3)$, $SE(3)$
- ▶ Properties of rotation matrix in $SO(2)$ and $SO(3)$
- ▶ Representation of spatial rotations
 - ▶ rotation matrix
 - ▶ axis-angle
 - ▶ exponential coordinates
 - ▶ quaternions
 - ▶ Euler angles
 - ▶ 6D representation



Laboratories goal

- ▶ Start implementing robotics toolbox
- ▶ <https://robotics-labs.readthedocs.io/>
- ▶ Utilities to work with $SO(2)$, $SE(2)$, $SO(3)$, $SE(3)$
 - ▶ $\exp(\omega)$
 - ▶ $\log(R)$
 - ▶ R^{-1}
 - ▶ ...



Laboratories goal

- ▶ Start implementing robotics toolbox
- ▶ <https://robotics-labs.readthedocs.io/>
- ▶ Utilities to work with $SO(2)$, $SE(2)$, $SO(3)$, $SE(3)$
 - ▶ $\exp(\omega)$
 - ▶ $\log(R)$
 - ▶ R^{-1}
 - ▶ ...
- ▶ Preparation
 - ▶ Linux and MiniConda are recommended
 - ▶ Install Miniconda
 - ▶ Install Python IDE (PyCharm, VSCode)

