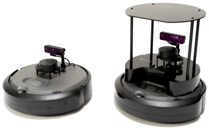
Robotics: Rigid body motion

Vladimír Petrík

vladimir.petrik@cvut.cz

25.09.2023

What is robot?



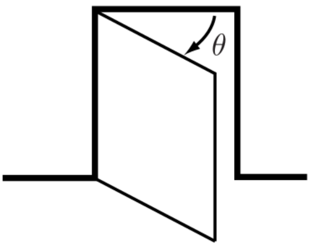
Mobilní robot, UGV - unmanned ground vehicle

Flying robots (*e.g.* drones)

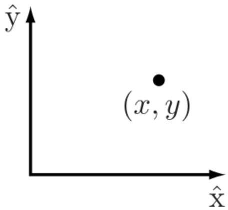
Manipulators (např. Franka Emika Panda)

Robot configuration

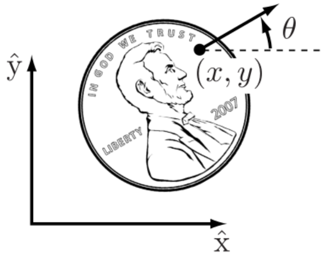
- Complete specification of the position of every point of the robot.



The configuration is described by the angle θ.



Point in plane is described by two coordinates.



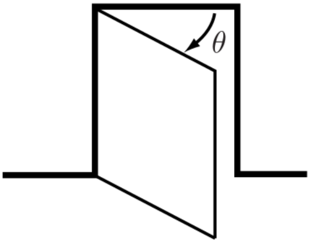
Planar rigid object configuration consists of the position and orientation.

Degrees of freedom (DoF)

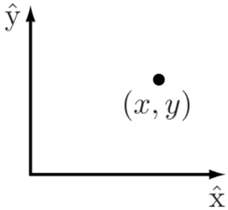
- The minimum number of real-valued coordinates needed to represent the configuration.
  - door: 1
  - planar point: 2
  - planar rigid object: 3
  - manipulators: from 1 (*e.g.* rotating table) to tens (*e.g.* humanoids)
- Determining DoF
  - (sum of freedom of the points) - (number of independent constraints)
  - Rigid objects
    - 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 - $\mathcal{C}$

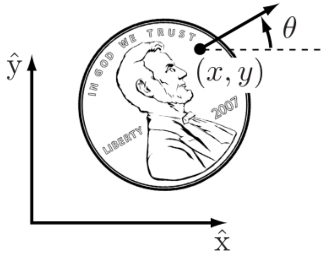
- The N-dimensional space (N correspond to number of DoF)
- Every point of configuration space correspond to one configuration
- Contains all possible configurations of the robot



$\mathcal{C} : \langle 0°, 180° \rangle$ or
$\mathcal{C} : \langle -180°, 180° \rangle$

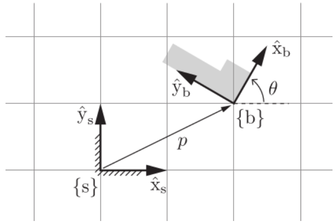


$\mathcal{C} : \mathbb{R}^2$

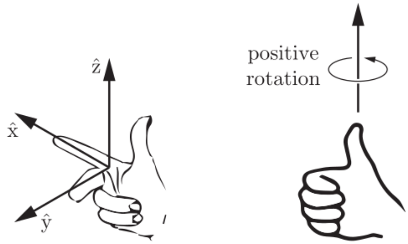


$\mathcal{C} : \mathbb{R}^2 \times \langle 0°, 360°)$

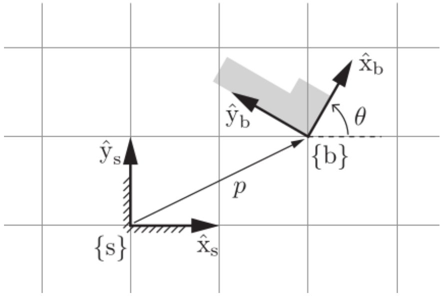
Rigid body motion in plane



- 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

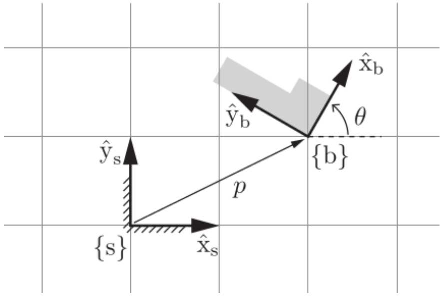


- The configuration of body is given by
  - position of body frame w.r.t. reference frame
  - orientation of body frame w.r.t. reference frame
- Body frame origin
  - $\boldsymbol{p} = p_x \hat{\boldsymbol{x}}_s + p_y \hat{\boldsymbol{y}}_s \in \mathbb{R}^2$
  - If reference frame is clear from the context: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

$SO(2)$

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

$SE(2)$

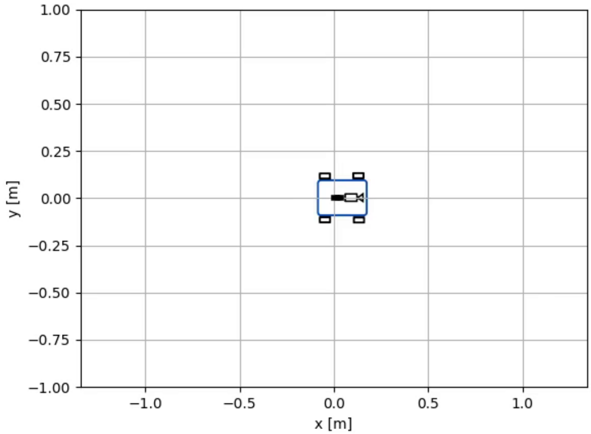


- A pair $(R_{ab}, \boldsymbol{p})$
  - represents pose/configuration of the body
  - changes the reference frame of a vector
    $\boldsymbol{v}_a = R_{ab}\boldsymbol{v}_b + \boldsymbol{p}$
  - moves vector/frame $(R, \boldsymbol{t})$
    $\boldsymbol{R}_{\text{moved}} = R_{ab}R \quad \boldsymbol{t}_{\text{moved}}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

$SE(2)$ example

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

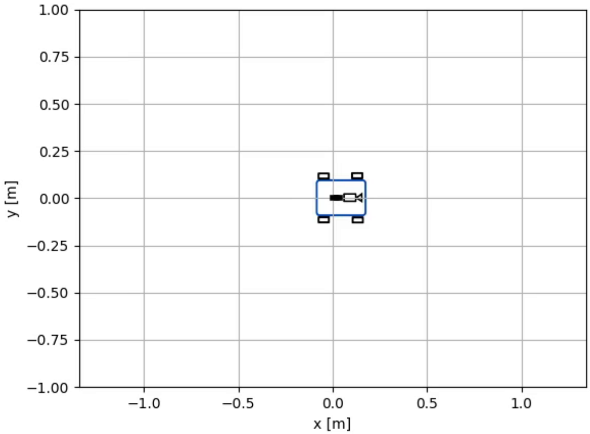
Delta transformations are defined in robot frame.



$SE(2)$ example

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

Delta transformations are defined in reference frame.



$SE(2)$ example camera

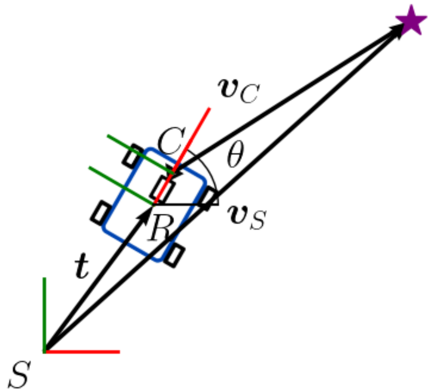
$$T_{SR} = \begin{pmatrix} R(\theta) & \boldsymbol{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

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

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

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

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



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

- $SO(3)$
  - $\det(R) = 1$
  - $RR^\top = I$, *i.e.* $R^{-1} = R^\top$
  - $(R_1R_2)\,R_3 = R_1\,(R_2R_3)$
  - $R_1R_2 ? =? \neq R_2R_1$ obecně
- $SE(3)$
  - $\boldsymbol{v}_a^H = T_{ab}\boldsymbol{v}_b^H$
  - $(T_1T_2)\,T_3 = T_1\,(T_2T_3)$
  - $T_1T_2 \neq T_2T_1$
  - $T^{-1} = \begin{pmatrix} R^\top & -R^\top\boldsymbol{t} \\ \boldsymbol{0}^\top & 1 \end{pmatrix}$

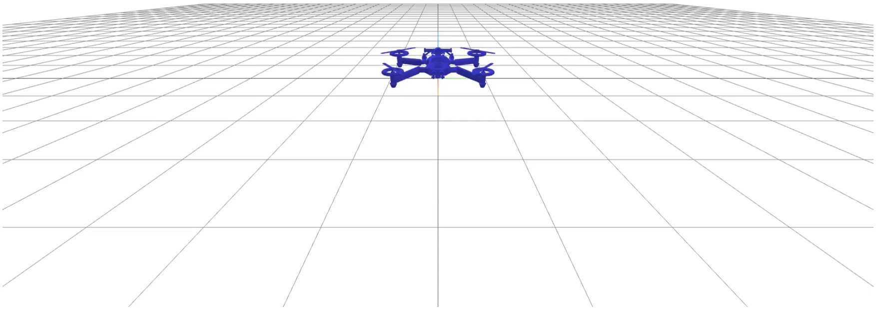
How to compute $R \in SO(3)$?

- 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 $SE(3)$

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

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



Axis-angle representation

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

Exponential coordinates

- A single vector $\boldsymbol{\omega} \in \mathbb{R}^3$
- Also called Euler vector or Euler-Rodrigues parameters
- Mapping to angle-axis representation:
  - $\theta = \|\boldsymbol{\omega}\|$
  - $\hat{\boldsymbol{\omega}} = \frac{\boldsymbol{\omega}}{\theta}$
- Exponential to/from R
  - $R = \exp \boldsymbol{\omega}$: use Rodrigues' formula
  - $\boldsymbol{\omega} = \log R$: use angle axis from R algorithm
- Why exponential?
  - it correspond to matrix exponential/logarithm of $[\boldsymbol{\omega}]$
  - if $\boldsymbol{\omega}$ 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

- $\boldsymbol{q} \in \mathbb{R}^4, \quad \|\boldsymbol{q}\| = 1$
- From axis-angle
  - $q_w = \cos(\theta/2)$
  - $\boldsymbol{q}_{xyz} = \hat{\boldsymbol{\omega}} \sin(\theta/2)$
- From R
  - $q_w = 1/2\sqrt{1 + \operatorname{tr} R}$
  - $\boldsymbol{q}_{xyz} = \frac{1}{4q_w} \begin{pmatrix} r_{32} - r_{23} & r_{13} - r_{31} & r_{21} - r_{12} \end{pmatrix}^\top$
- To R
  - $R = \exp\left(2 \arccos(q_w) \frac{\boldsymbol{q}_{xyz}}{\|\boldsymbol{q}_{xyz}\|}\right)$
  - *i.e.* rotate about $\boldsymbol{q}_{xyz}$ 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
- 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
- Utilities to work with $SO(2)$, $SE(2)$, $SO(3)$, $SE(3)$
  - $\exp(\boldsymbol{\omega})$
  - $\log(R)$
  - R^{-1}
  - $\ldots$
- Preparation
  - Linux and Conda are recommended
  - Install conda
  - Install Python IDE (PyCharm, VSCode)