上海科技大学
ShanghaiTech University

CS283: Robotics Spring 2025: Kinematics

Sören Schwertfeger / 师泽仁

ShanghaiTech University

ADMIN

Project

- 2 credit points!
- Work in groups, min 2 students, max 3 students!
- Next lecture: Topics will be proposed…
  - You can also do your own topic, but only after approval of Prof. Schwertfeger
    - Prepare a short, written proposal till next Tuesday!

- Topic selection: Next Thursday!
  - One member writes an email for the whole group to Bowen: zhangyq12023 (at)shanghaitech.edu.cn ; Put the other group members on CC
  - Subject: [Robotics] Group Selection

- One graduate student from my group will co-supervise your project
- Weekly project meetings!

- Oral ”exams” to evaluate the contributions of each member
- No work on project => bad grade of fail

Grading

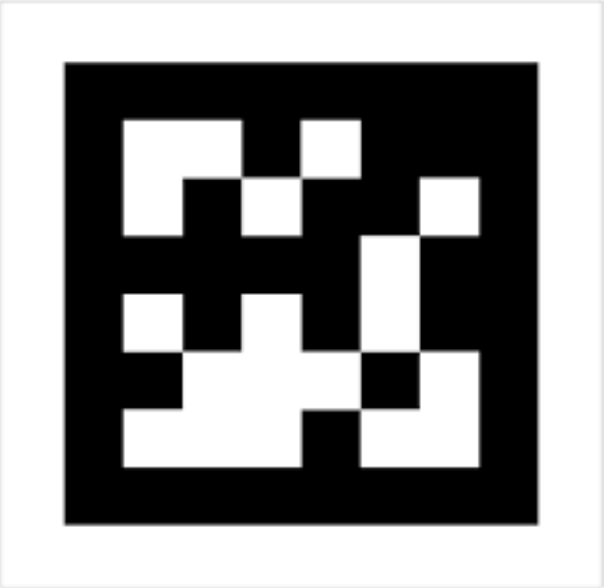
- Grading scheme is not 100% fixed
- Approximately:
  - Lecture: 50%
    - Quizzes during lecture (reading assignments): 4%
    - Homework: 18%
    - Midterm: 8%
    - Final: 20%
  - Project: 50%
    - Paper Presentation: 5%
    - Project Proposal: 5%
    - Intermediate Report: 5%
    - Weekly project meetings: 10%
    - Final Report: 10%
    - Final Demo: 10%
    - Final Webpage: 5%

Campus Autonomy: High Speed Navigation

- Use ROS move_base and TEB planner for high speed robot control.
- Include robot dynamics (mass, acceleration, …)
- Use 3D LRF to detect and predict motion of obstacles (open source software available)
- High-speed navigation through light crowds of students.

- Difficulty: medium
- Requite: good demo
- Supervisor: Yongqi

Ground Truth Localization via AprilTags



- Print (very big) AprilTags – and distribute in scenario (e.g. underground parking)
- Use Faro 3D scanner to (semi-) automatically detect and locate AprilTags ->
- Build ground truth 3D map of AprilTag poses
- Write a small program to detect AprilTags in the sensor data
- (If observed with more than one camera, minimize error)
- Generate ground truth trajectories with this
- Difficulty: Medium
- Graduate Supervisor: Bowen Xu

Robot Dog Project

- Reserved for certain students
- Program advanced capabilities for robot
- Difficulty: High
- Graduate Supervisor: Xin Duan

Fetch Robot

- Some nice project with fetch robot

- Difficulty: Advanced
- Supervisor: Yaxun Yang

Cotton Project Revival

- Difficulty: Advanced
- Supervisor: Prof. Schwertfeger
- Big project – 2 teams can share the work:
- Re-do the cotton collection hardware
- Revive the perception and control part

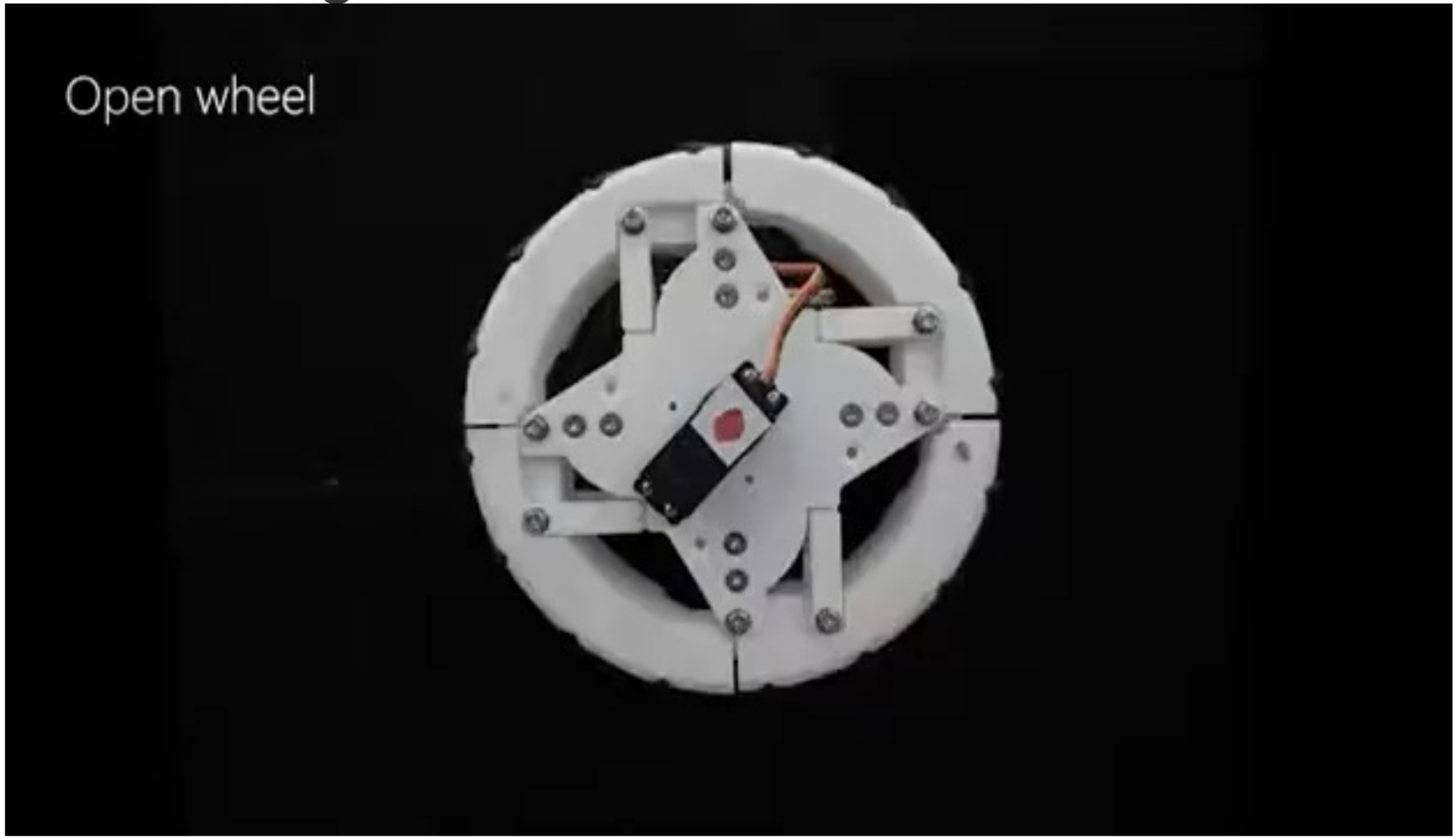
Project Suggestion: Draw with Sand

- Build and program such a robot …
- Quite difficult …
- But cool …
- Bigger group (with sub-tasks) allowed

Finish Omni-Wheel-Leg Journal Paper

- Do final work on journal paper
- Extension of this paper:
- OmniWheg: An Omnidirectional Wheel-Leg Transformable Robot
- Ruixiang Cao; Jun Gu; Chen Yu; Andre Rosendo
- https://ieeexplore.ieee.org/document/9982030

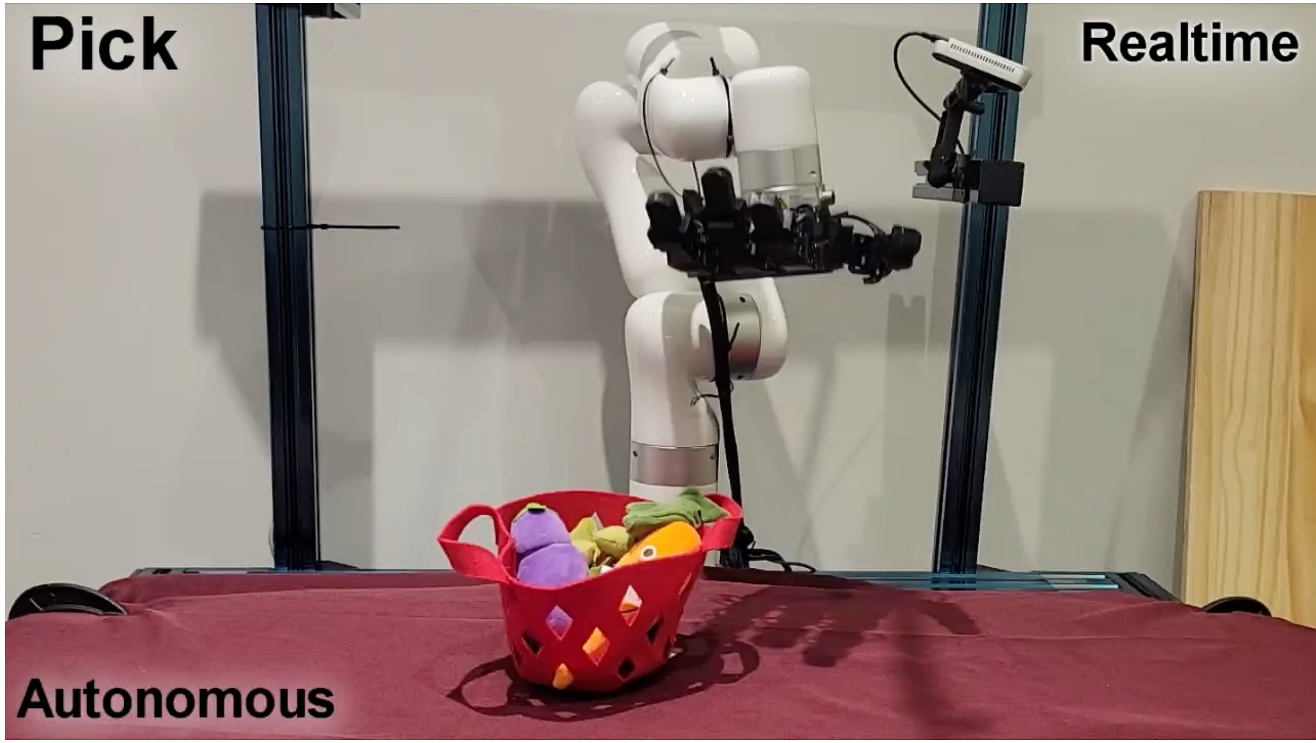
- Advisor: Fujing



Leap Hand

- Install https://v1.leaphand.com/ LeaHand on Kinova Arm
- Get all the software to work well together
- Work together with Yaxun on her paper

- Difficulty medium
- Supervisor: Yaxun Yang



Robot Introspection for LLMs

- Collect all kinds of robot status data, e.g.:
  - Size, height, weight, capabilities, max speed, urdf, …
  - Current speed, current power consumption, current direction, current mission objective, current battery status, current CPU temp, cpu usage, mem usage
  - ROS status, running nodes, available topics & services, joint values, console log, …
  - All kinds of other, robot intrinsic data

- Feed it into an LLM
- Generate a benchmark to test how well the LLM understands the robot

- Supervisor: Prof. Schwertfeger

- Max one group per topic!
- In case of double selection we will discuss alternatives with both groups
- If no one changes it, it will be "First come - First Serve"

			Difficulty:		
	Name	**Advisor**	**Hardware**	**Software**	**Algorithm**
1	Campus Autonomy: High Speed Navigation	Yongqi	low	low	medium +
2	April Tag Localization	Bowen	medium	medium	medium
3	Robot Dog Project	Xin Duan	medium	low	medium
4	Fetch Project	Yaxun Yang	low	medium	medium
5	Cotton Project Revival: Gripping	Prof. Schwertfeger	medium+	low	low
6	Cotton Project Revival: Perception & Autonomy	Prof. Schwertfeger	low	advanced	low
7	Writing Project	Prof. Schwertfeger	advanced	medium	medium
8	OmniWheg Project	Fujing	medium	medium	medium
9	Leap Project	Yaxun Yang	medium	medium	medium
10	Robot Introspection for LLMs	Prof. Schwertfeger	low	medium	medium

KINEMATICS

Motivation

- Autonomous mobile robots move around in the environment. Therefore **ALL** of them:
  - They need to know **where** they **are**.
  - They need to know **where** their **goal** is.
  - They need to know **how** to get there.

- **Odometry!**
- Robot:
  - I know how fast the wheels turned =>
  - I know how the robot moved =>
  - I know where I am ☺

Odometry

- Robot:
  - I know how fast the wheels turned =>
  - I know how the robot moved =>
  - I know where I am ☺

- Marine Navigation: Dead reckoning (using heading sensor)

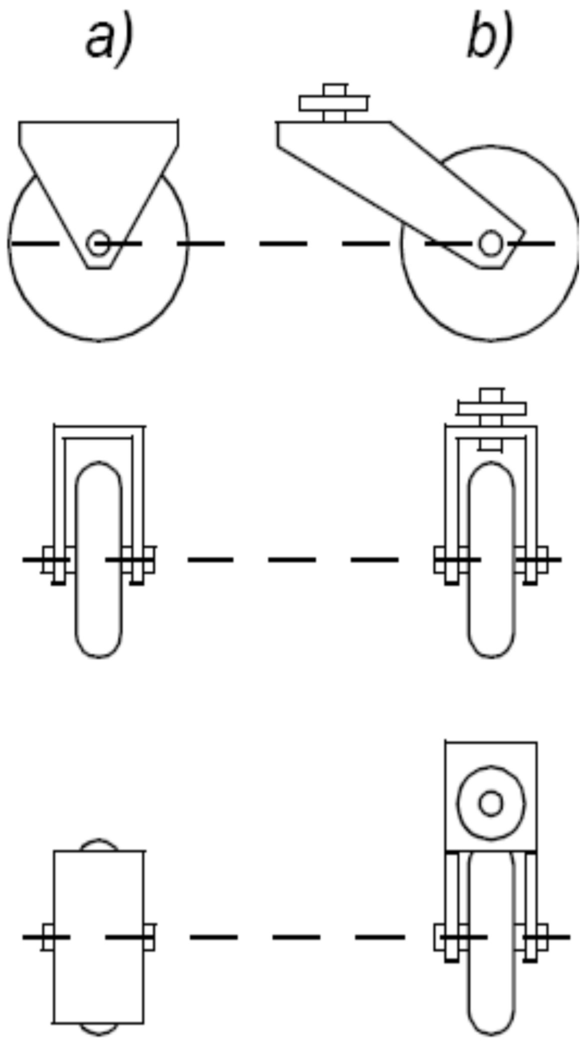
- Sources of error (AMR pages 269 - 270):
  - Wheel slip
    - Uneven floor contact (non-planar surface)
    - Robot kinematic: tracked vehicles, 4 wheel differential drive..
  - Integration from speed to position: Limited resolution (time and measurement)
  - Wheel misalignment
  - Wheel diameter uncertainty
  - Variation in contact point of wheel

Mobile Robots with Wheels

- Wheels are the most appropriate solution for most applications
- Three wheels are sufficient to guarantee stability
- With more than three wheels an appropriate suspension is required
- Selection of wheels depends on the application

The Four Basic Wheels Types I

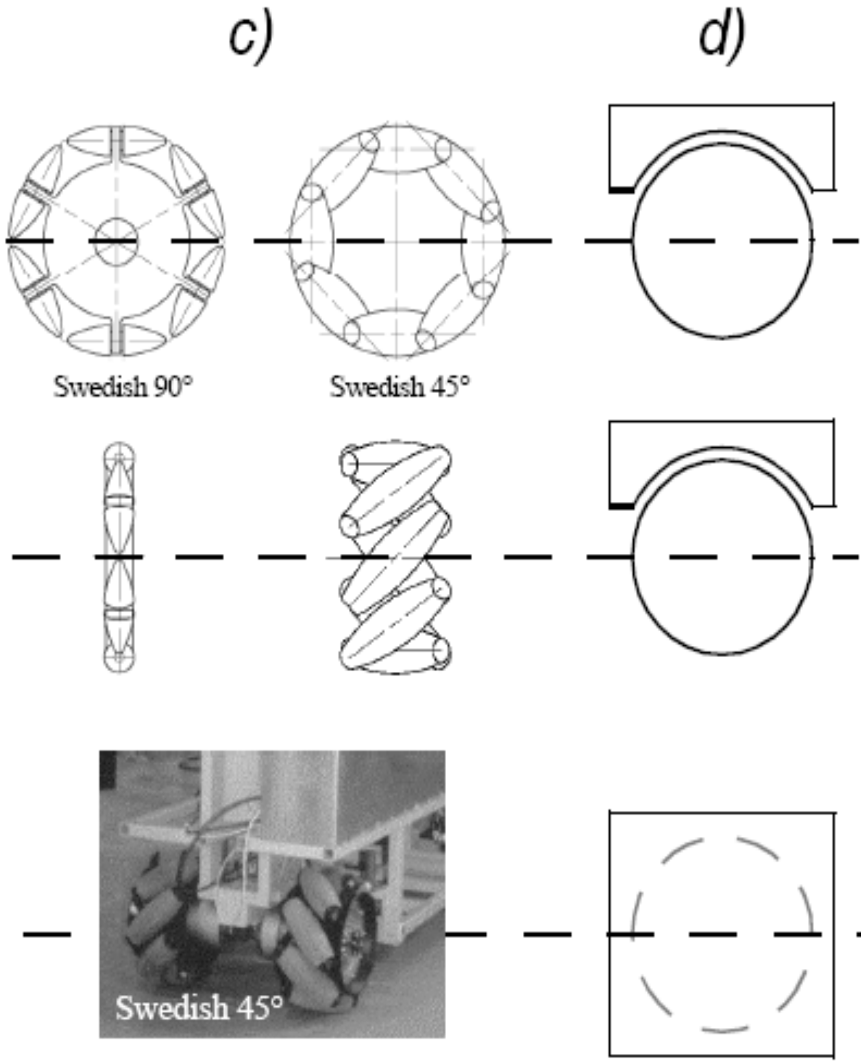
- a) Standard wheel: Two degrees of freedom; rotation around the (motorized) wheel axle and the contact point
- b) Castor wheel: Three degrees of freedom; rotation around the wheel axle, the contact point and the castor axle



The Four Basic Wheels Types II

- c) Swedish wheel: Three degrees of freedom; rotation around the (motorized) wheel axle, around the rollers and around the contact point

- d) Ball or spherical wheel: Suspension technically not solved

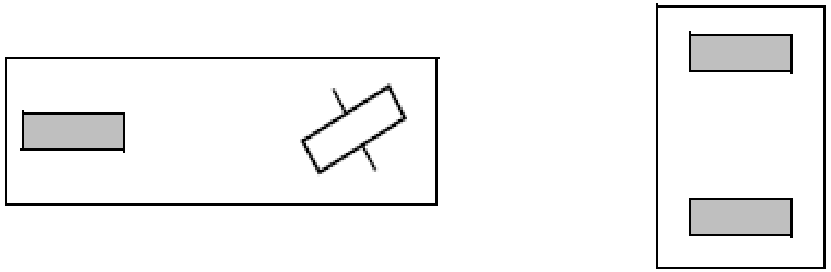


Characteristics of Wheeled Robots and Vehicles

- Stability of a vehicle is be guaranteed with 3 wheels
  - center of gravity is within the triangle with is formed by the ground contact point of the wheels.
- Stability is improved by 4 and more wheel
  - however, this arrangements are hyperstatic and require a flexible suspension system.
- Bigger wheels allow to overcome higher obstacles
  - but they require higher torque or reductions in the gear box.
- Most arrangements are non-holonomic (see chapter 3)
  - require high control effort
- Combining actuation and steering on one wheel makes the design complex and adds additional errors for odometry.

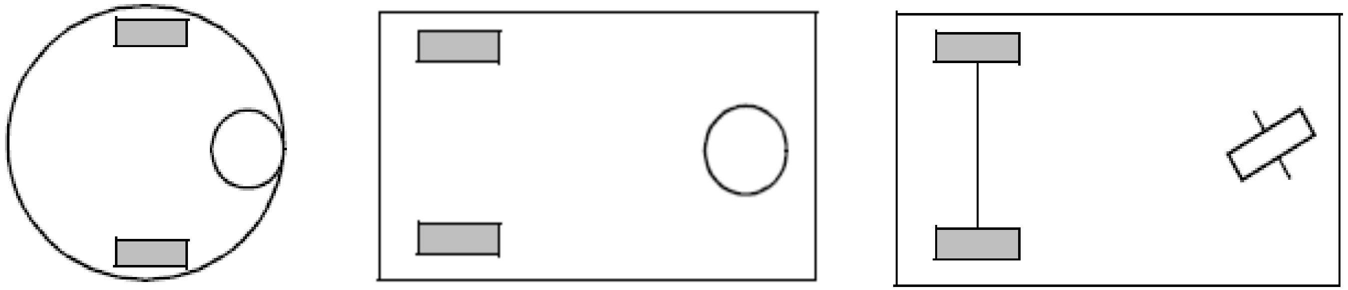
Different Arrangements of Wheels I

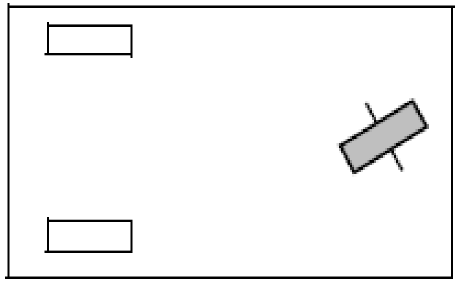
- Two wheels

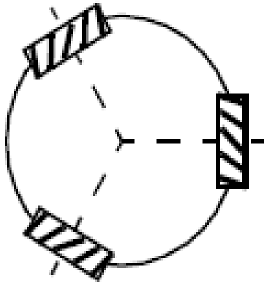


Center of gravity below axle

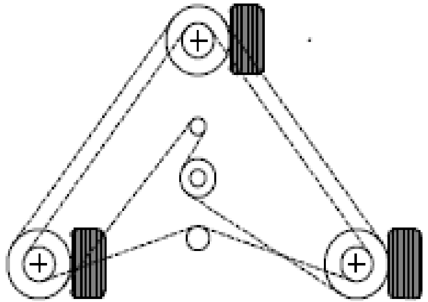
- Three wheels







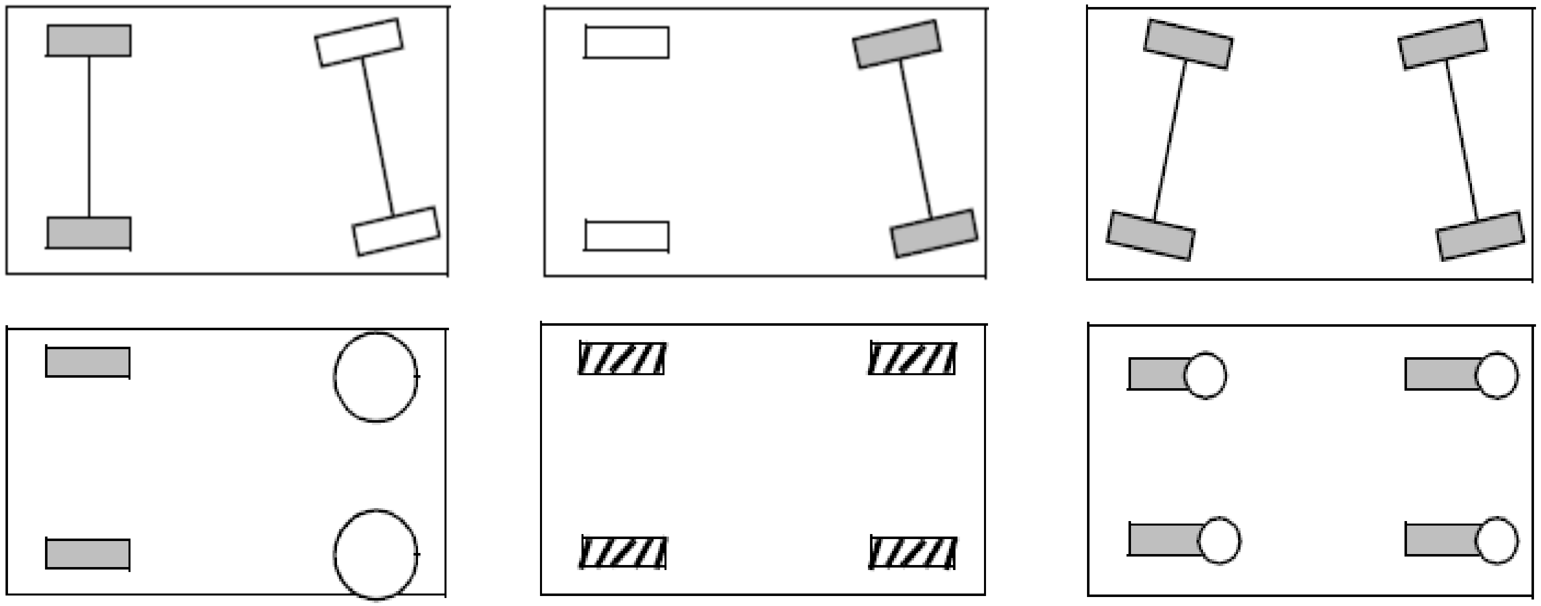
Omnidirectional Drive

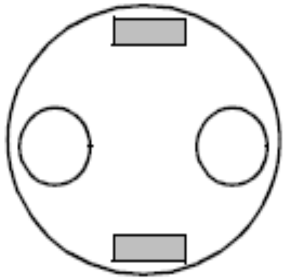


Synchro Drive

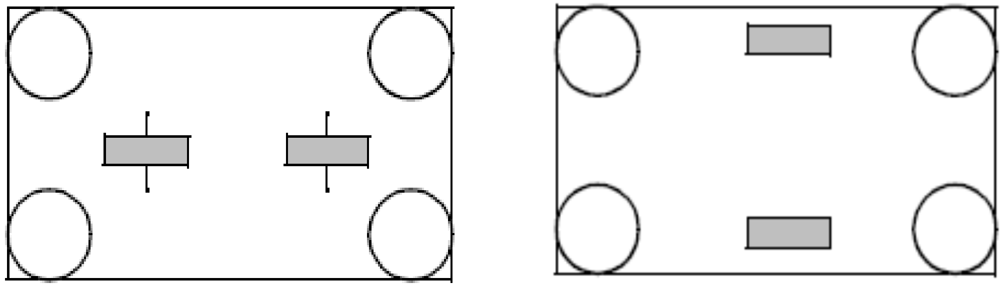
Different Arrangements of Wheels II

- Four wheels



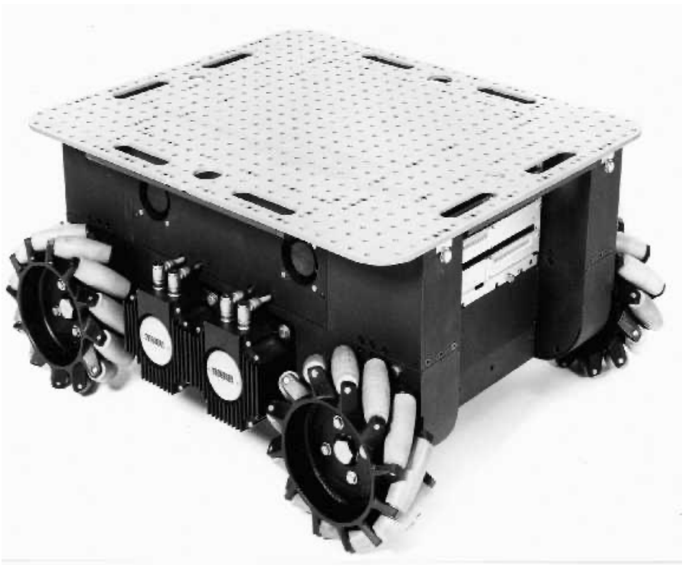


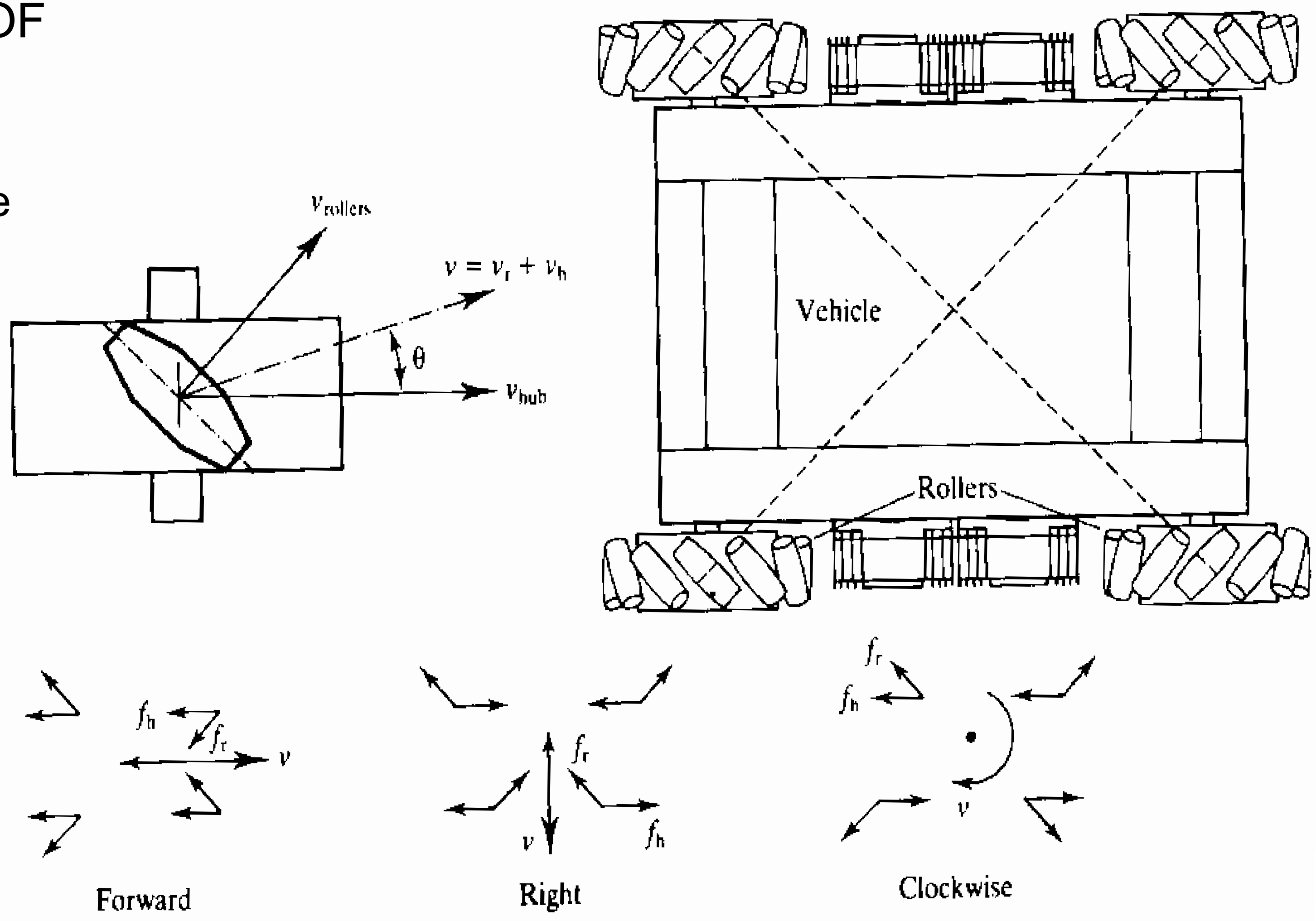
- Six wheels



Uranus, CMU: Omnidirectional Drive with 4 Wheels

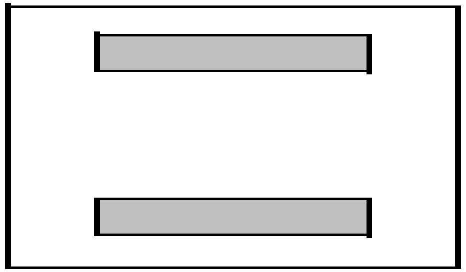
- Movement in the plane has 3 DOF
  - thus only three wheels can be independently controlled
  - It might be better to arrange three swedish wheels in a triangle

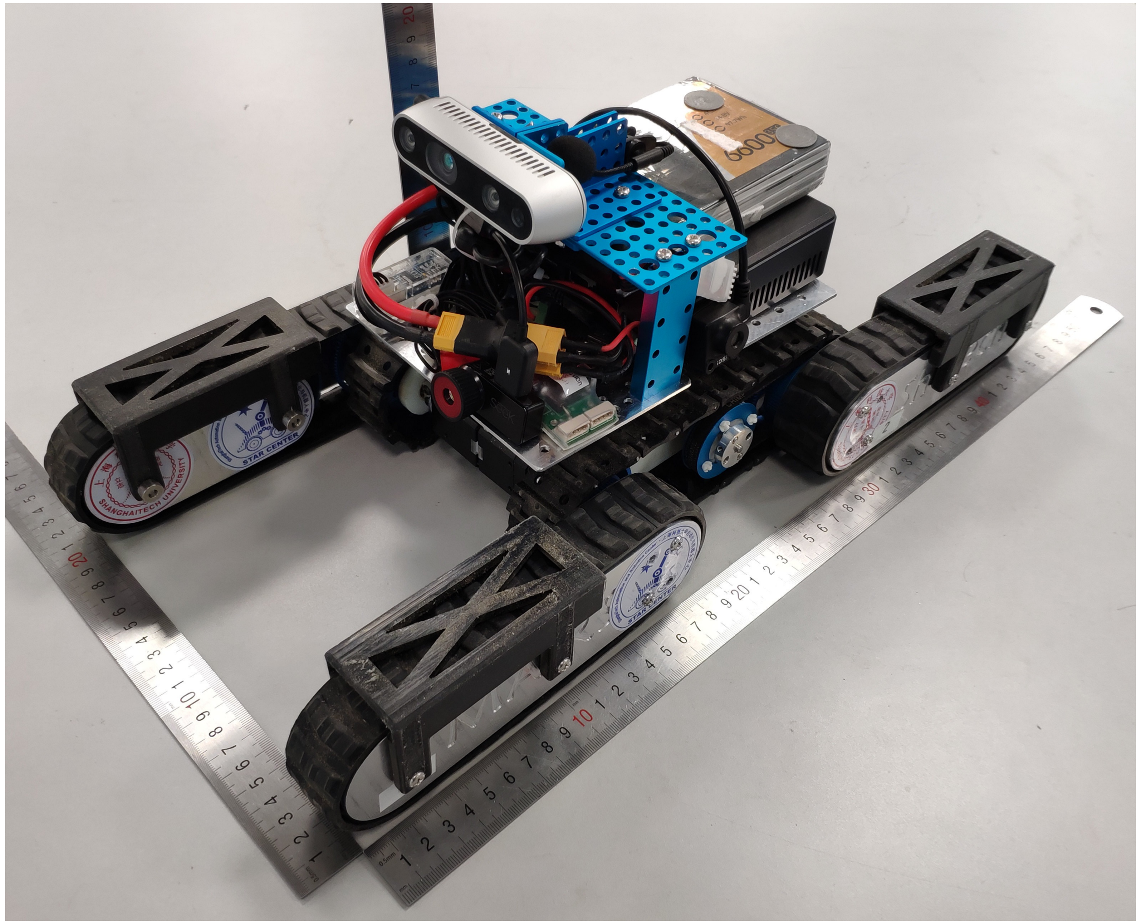




MARS Rescue Robot: Tracked Differential Drive

- Kinematic Simplification:
  - 2 Wheels, located at the center

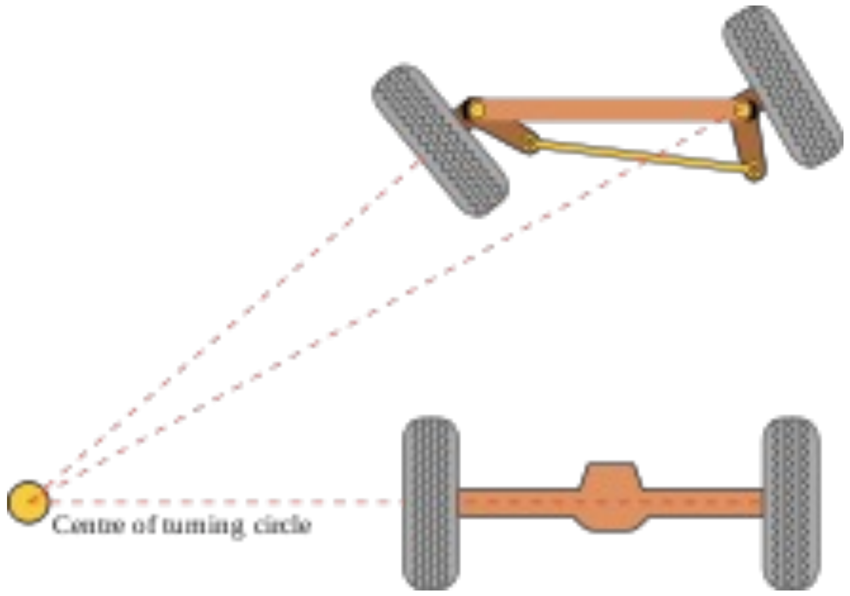




Differential Drive Robots

Ackermann Robot

- No sideways slip than differential drive during turning ☺
- Cannot turn on the spot ☹



Introduction: Mobile Robot Kinematics

- Aim
  - Description of mechanical behavior of the robot for *design* and *control*
  - Similar to robot manipulator kinematics
  - However, mobile robots can move unbound with respect to its environment
    - there is no direct way to measure the robot's position
    - Position must be integrated over time
    - Leads to inaccuracies of the position (motion) estimate
      -> the number 1 challenge in mobile robotics

Kinematics vs. Kinetics

Kinematics:

- Greek origin: “motion”, “moving”
- Describes motion of points and bodies
- Considers position, velocity, acceleration, ..
- Examples: Celestial bodies, particle systems, robotic arm, human skeleton

Kinetics:

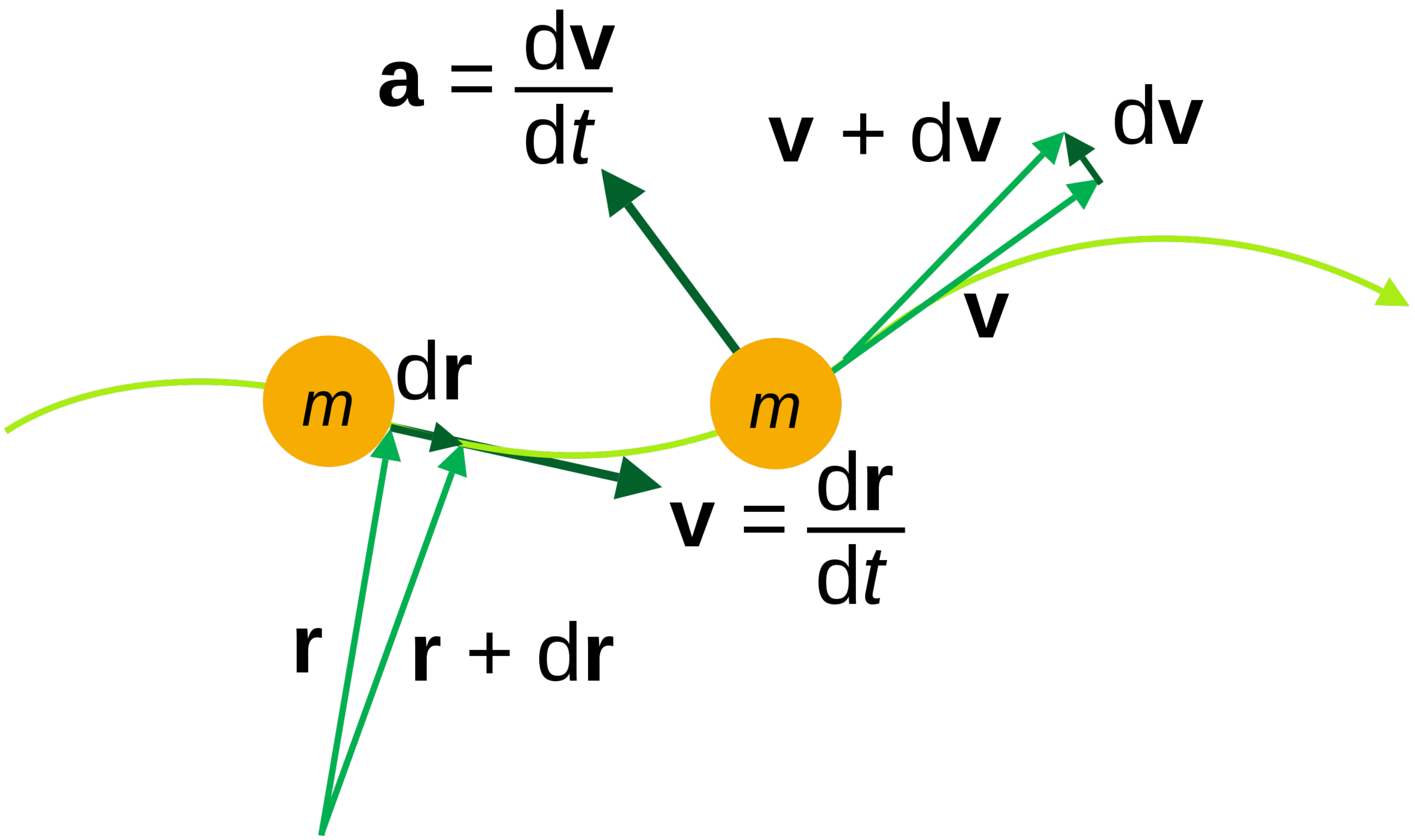
- Describes causes of motion
- Effects of forces/moments
- Newton’s laws, e.g., $F = ma$

Kinematics and Control Slides:
Andreas Geiger
https://uni-tuebingen.de/fakultaeten/mathematisch-naturwissenschaftliche-fakultaet/fachbereiche/informatik/lehrstuehle/autonomous-vision/lectures/self-driving-cars/

What are kinematics?

- Describes the motion of points, bodies (objects), and systems of objects
  - Does not consider the forces that cause them (that would be kinetics)
  - Also known as “the geometry of motion”
- For robotics:
  - Describes the motion of the vehicle
  - Puts position/orientation in relation with translational/angular velocities and accelerations
  - Used for regularization, prediction, etc.

What are kinematics?

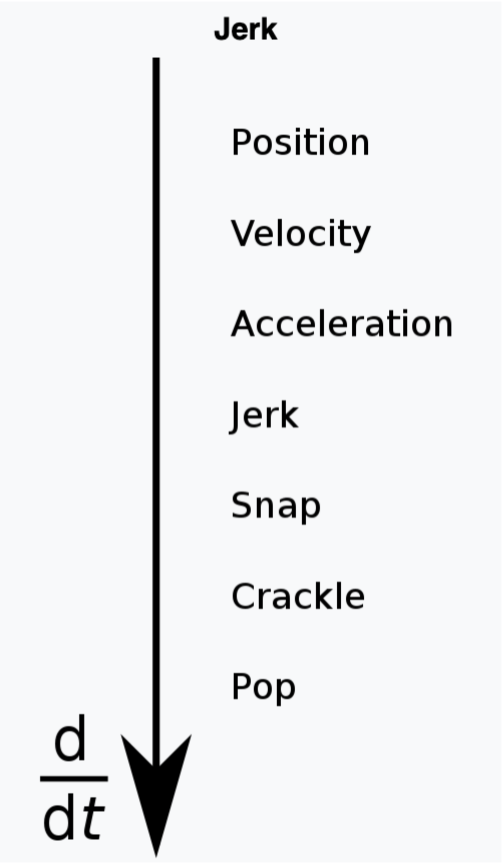


What are kinematics?

- It does not stop at acceleration, but theory involves an arbitrarily high number of derivatives:

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{a}_0 t^2 + \frac{1}{3}\mathbf{j} t^3 + \dots$$

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}_0 + \boldsymbol{\omega}_0 t + \frac{1}{2}\boldsymbol{\alpha}_0 t^2 + \frac{1}{3}\zeta t^3 + \dots$$



Jerk equations: minimal setting for solutions showing chaotic behavior!

In practice

- Often we use finite models to simplify/smoothify the system
  - Locally constant acceleration

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{a}_0 t^2$$

  - Locally constant velocity

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0 t$$

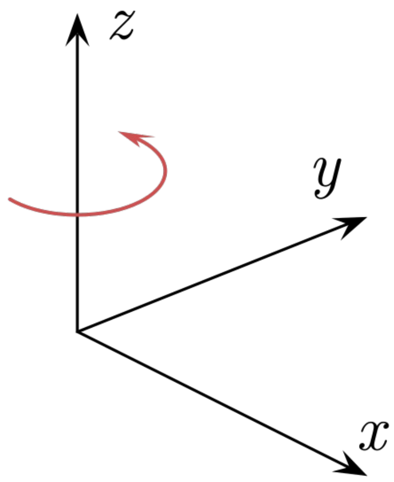
Why do we want to introduce kinematic models

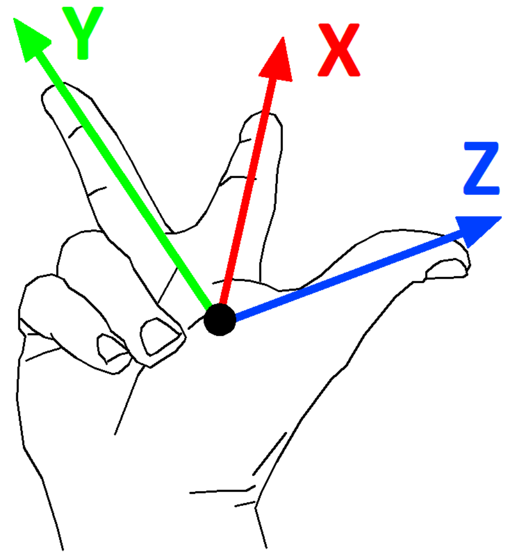
- For prediction
  - E.g.: If we have an initial estimate, we can use a kinematic model to generate a prior pose at a later point
- For smoothness
  - E.g.: If we estimate poses, we may constrain their difference to be consistent with some prior or measured velocity
- To impose constraints
  - E.g.: The motion may be more specific and include kinematic constraints
- For control
  - E.g.: Knowledge of how the system is moving is beneficial for reaching the goal pose

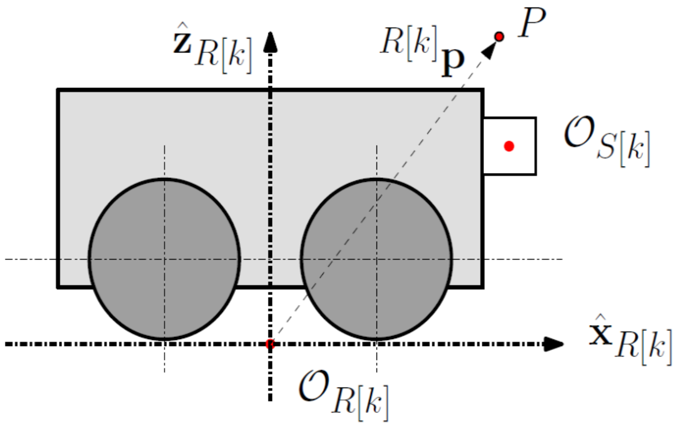
COORDINATE SYSTEM

Right Hand Coordinate System

- Standard in Robotics
- Positive rotation around X is anti-clockwise
- Right-hand rule mnemonic:
  - Thumb: z-axis
  - Index finger: x-axis
  - Second finger: y-axis
  - Rotation: Thumb = rotation axis, positive rotation in finger direction
- Robot Coordinate System:
  - X front
  - Z up (Underwater: Z down)
  - Y ???

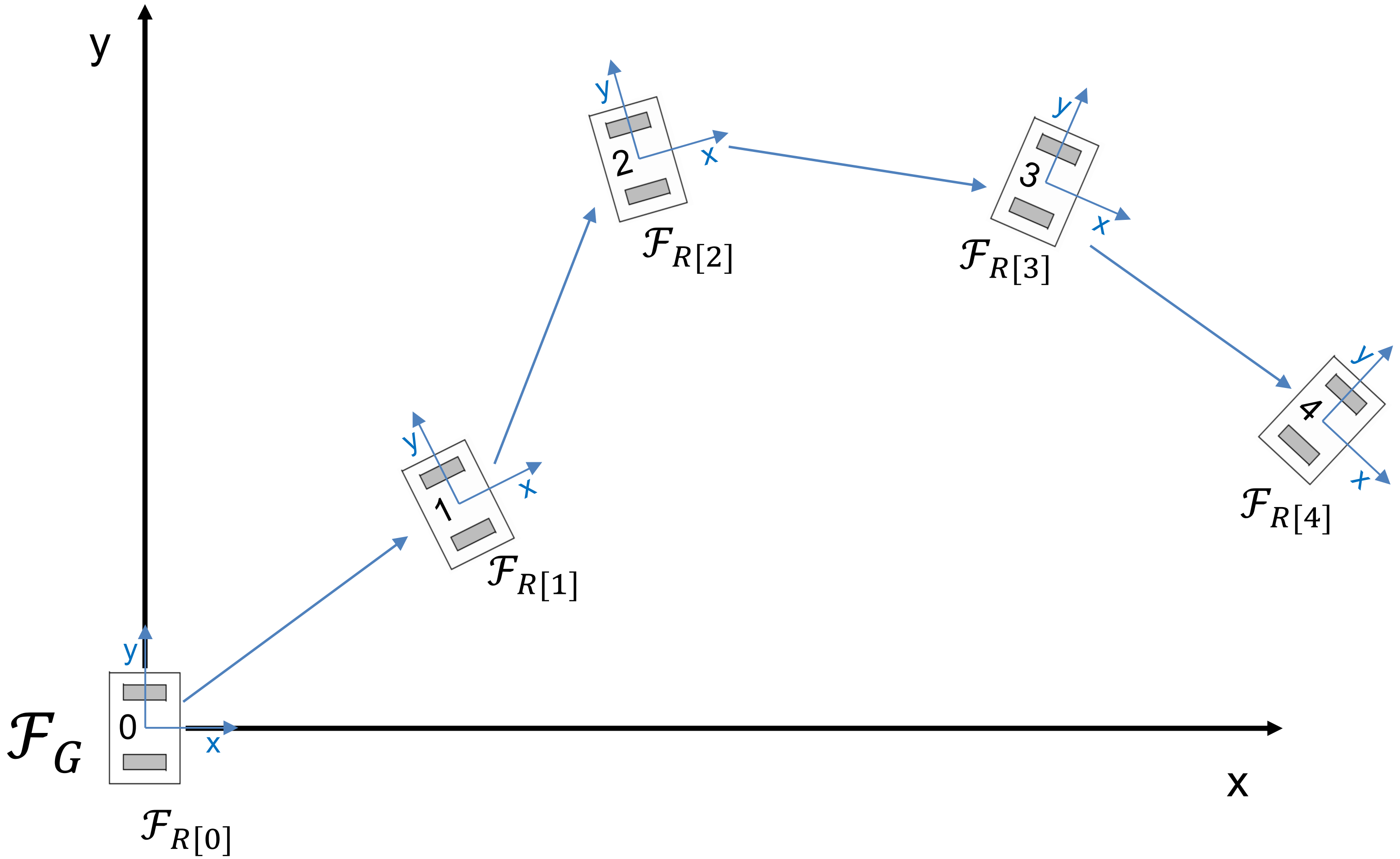






Right Hand Rule http://en.wikipedia.org/wiki/Right-hand_rule

Odometry



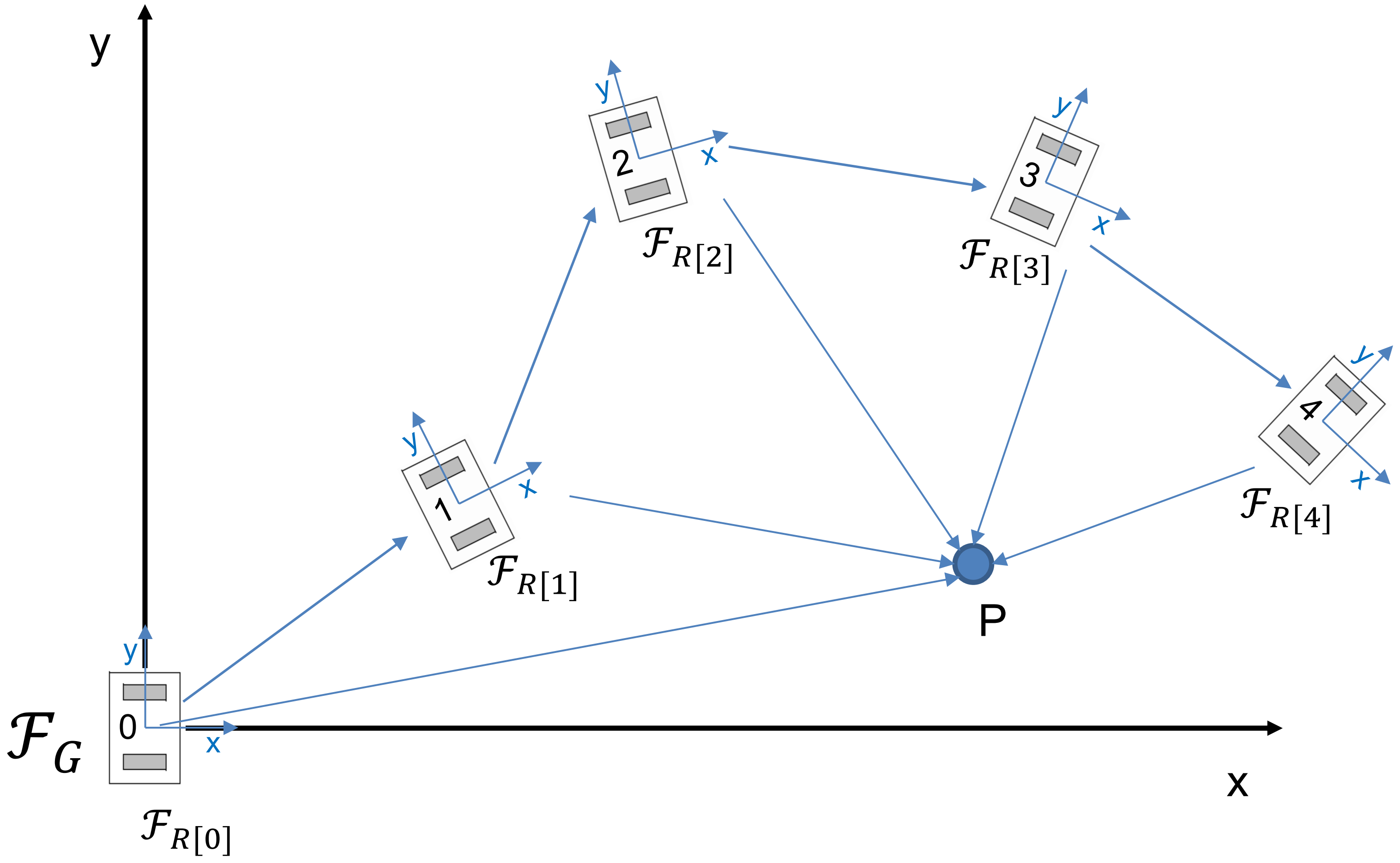
With respect to the robot start pose:
Where is the robot now?

Two approaches – same result:

- Geometry (easy in 2D)
- Transforms (better for 3D)

$\mathcal{F}_{R[X]}$: The ***F***rame of reference (the local coordinate system) of the ***R***obot at the time ***X***

Use of robot frames $\mathcal{F}_{R[X]}$

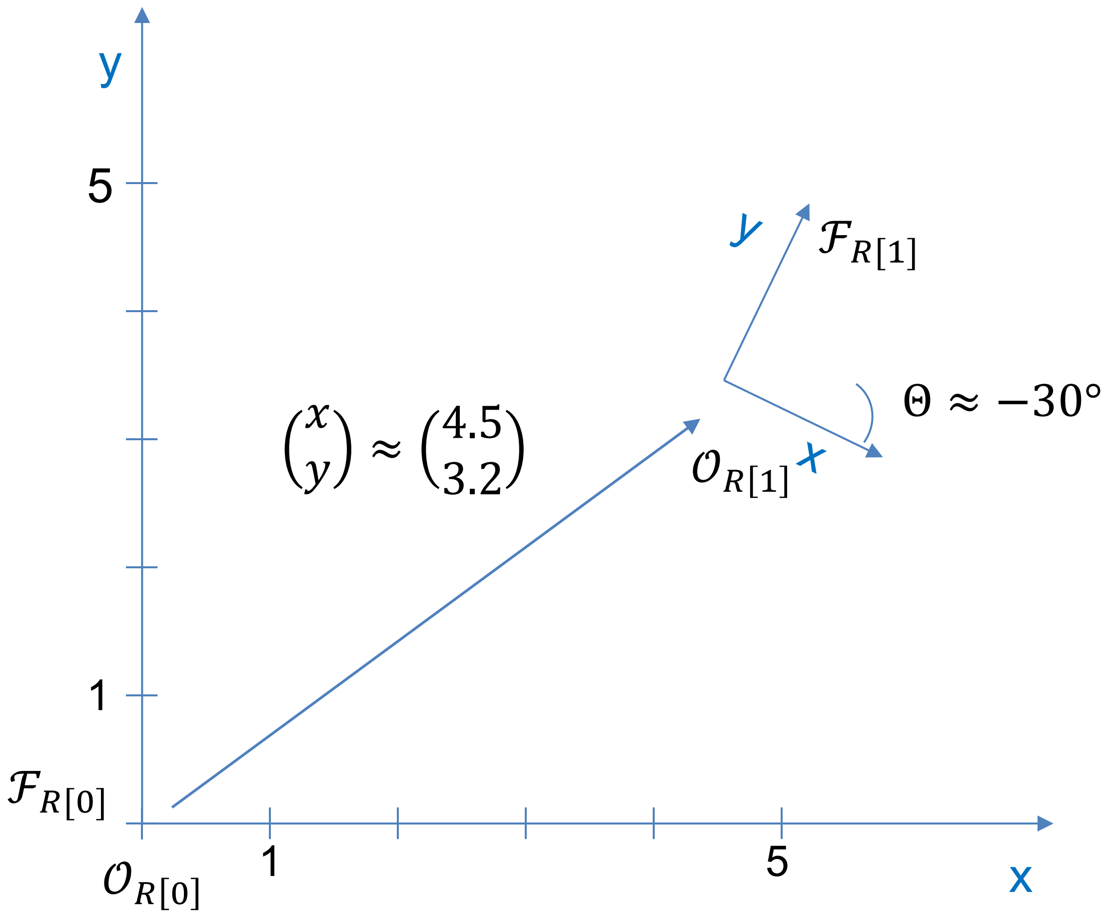


$\mathcal{O}_{R[X]}$: Origin of $\mathcal{F}_{R[X]}$ (coordinates (0, 0)

$\overrightarrow{\mathcal{O}_{R[X]}\mathrm{P}}$: position vector from $\mathcal{O}_{R[X]}$ to point P - $\binom{x}{y}$

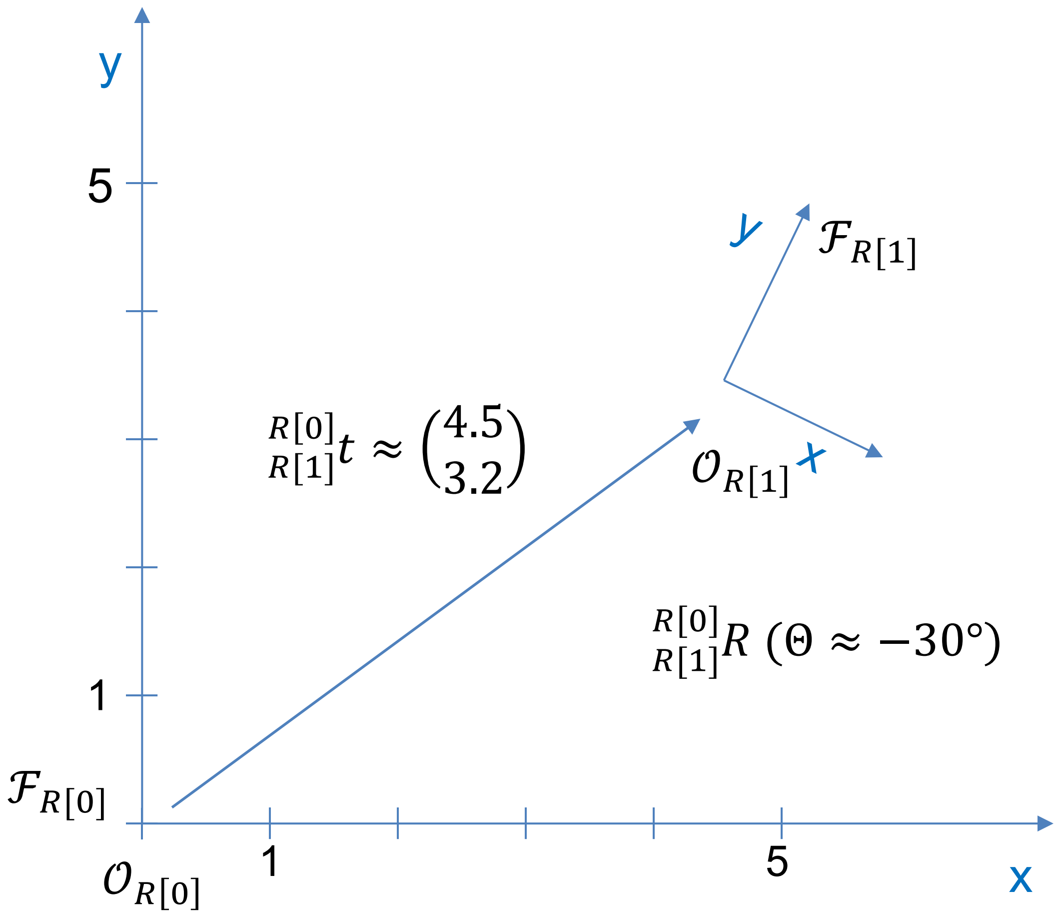
- Object P is observed at times 0 to 4
- Object P is static (does not move)
- The Robot moves (e.g. $\mathcal{F}_{R[0]} \neq \mathcal{F}_{R[1]}$)
- => (x, y) coordinates of P are different in all frames, for example:
  - $\overrightarrow{\mathcal{O}_{R[0]}\mathrm{P}} \neq \overrightarrow{\mathcal{O}_{R[1]}\mathrm{P}}$

Position, Orientation & Pose



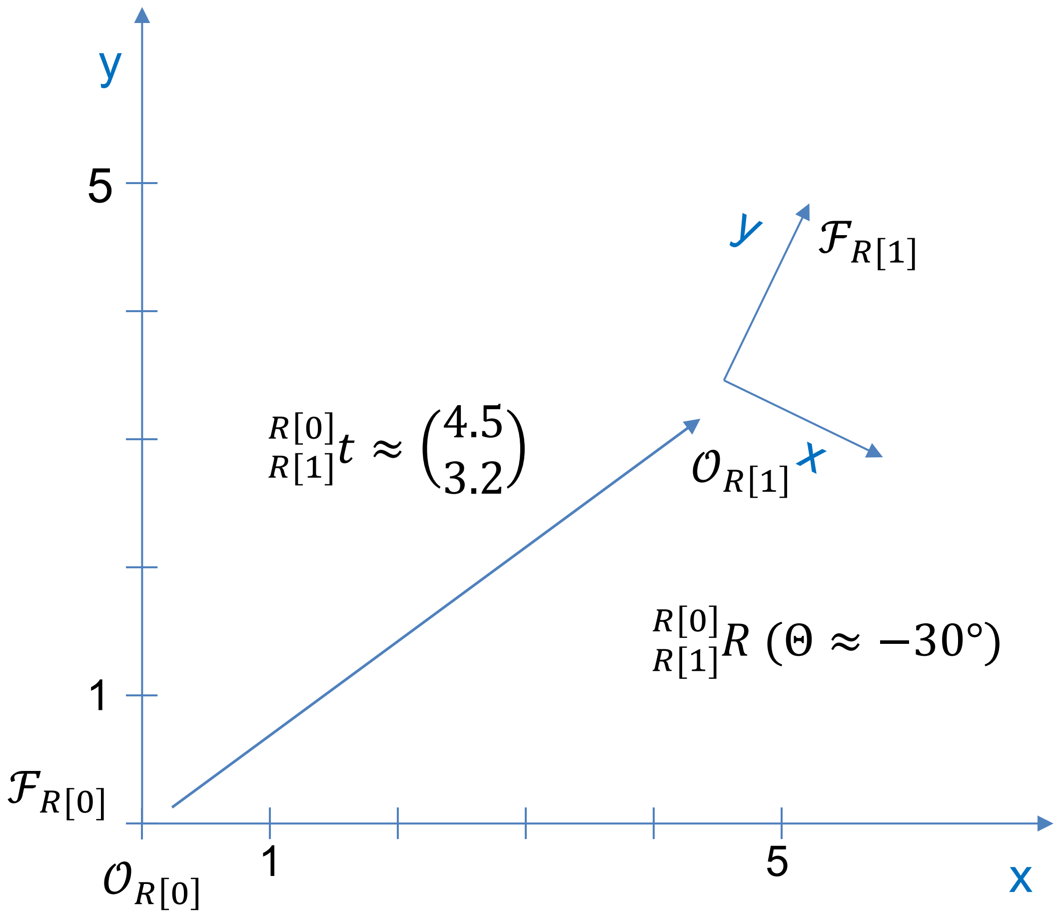
- **Position**:
  - $\begin{pmatrix} x \\ y \end{pmatrix}$ coordinates of any object or point (or another frame)
  - with respect to (wrt.) a specified frame
- **Orientation**:
  - (Θ) angle of any oriented object (or another frame)
  - with respect to (wrt.) a specified frame
- **Pose**:
  - $\begin{pmatrix} x \\ y \\ \Theta \end{pmatrix}$ position and orientation of any oriented object
  - with respect to (wrt.) a specified frame

Translation, Rotation & Transform



- **Translation**:
  - $\begin{pmatrix} x \\ y \end{pmatrix}$ difference, change, motion from one reference frame to another reference frame
- **Rotation**:
  - (Θ) difference in angle, rotation between one reference frame and another reference frame
- **Transform**:
  - $\begin{pmatrix} x \\ y \\ \Theta \end{pmatrix}$ difference, motion between one reference frame and another reference frame

Position & Translation, Orientation & Rotation



- $\mathcal{F}_{R[X]}$: Frame of reference of the robot at time X
- Where is that frame $\mathcal{F}_{R[X]}$?
  - Can only be expressed with respect to (wrt.) another frame (e.g. global Frame $\mathcal{F}_G$) =>
  - Pose of $\mathcal{F}_{R[X]}$ wrt. $\mathcal{F}_G$

- $\mathcal{O}_{R[X]}$: Origin of $\mathcal{F}_{R[X]}$
  - $\overrightarrow{\mathcal{O}_{R[X]}\mathcal{O}_{R[X+1]}}$: **Position** of $\mathcal{F}_{R[X+1]}$ wrt. $\mathcal{F}_{R[X]}$

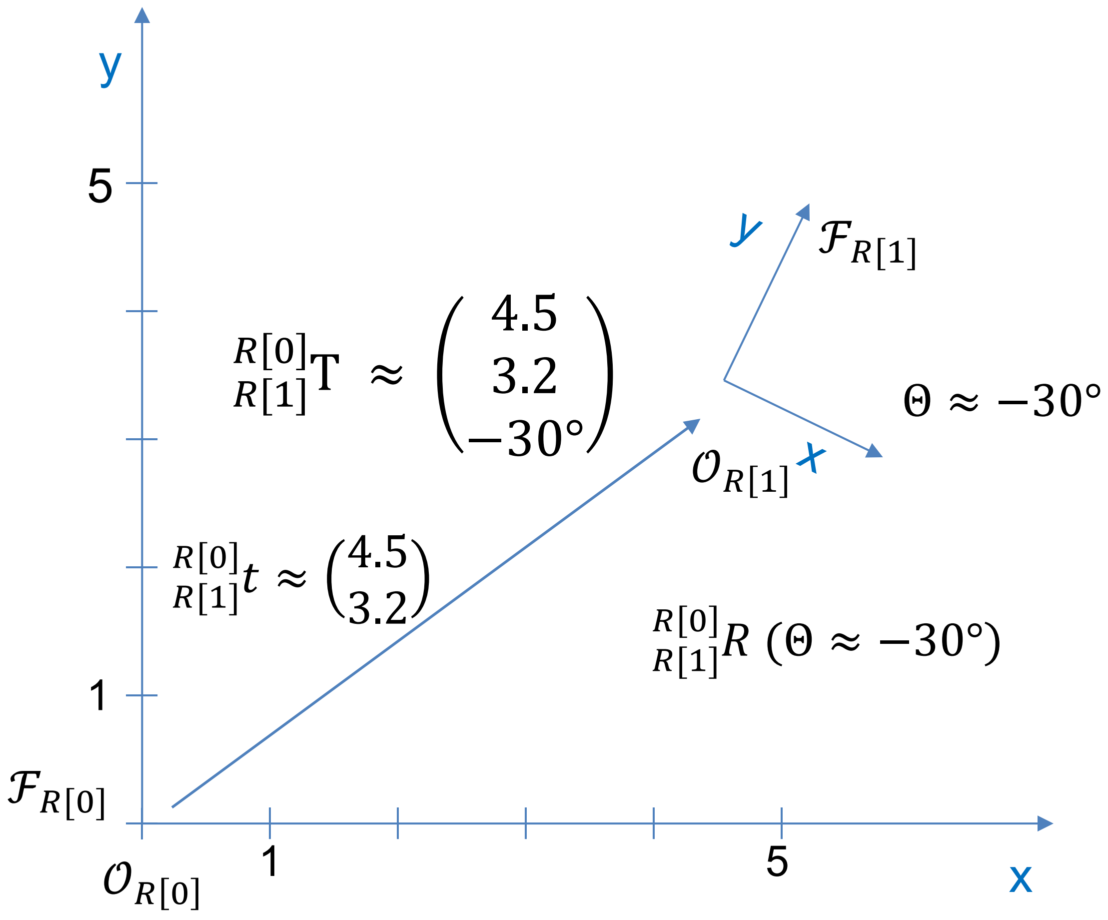
    so $\mathcal{O}_{R[X+1]}$ wrt. $\mathcal{F}_{R[X]}$

    $\triangleq \ {}^{R[X]}_{R[X+1]}t$: **Translation**

- The angle Θ between the x-Axes:
  - **Orientation** of $\mathcal{F}_{R[X+1]}$ wrt. $\mathcal{F}_{R[X]}$

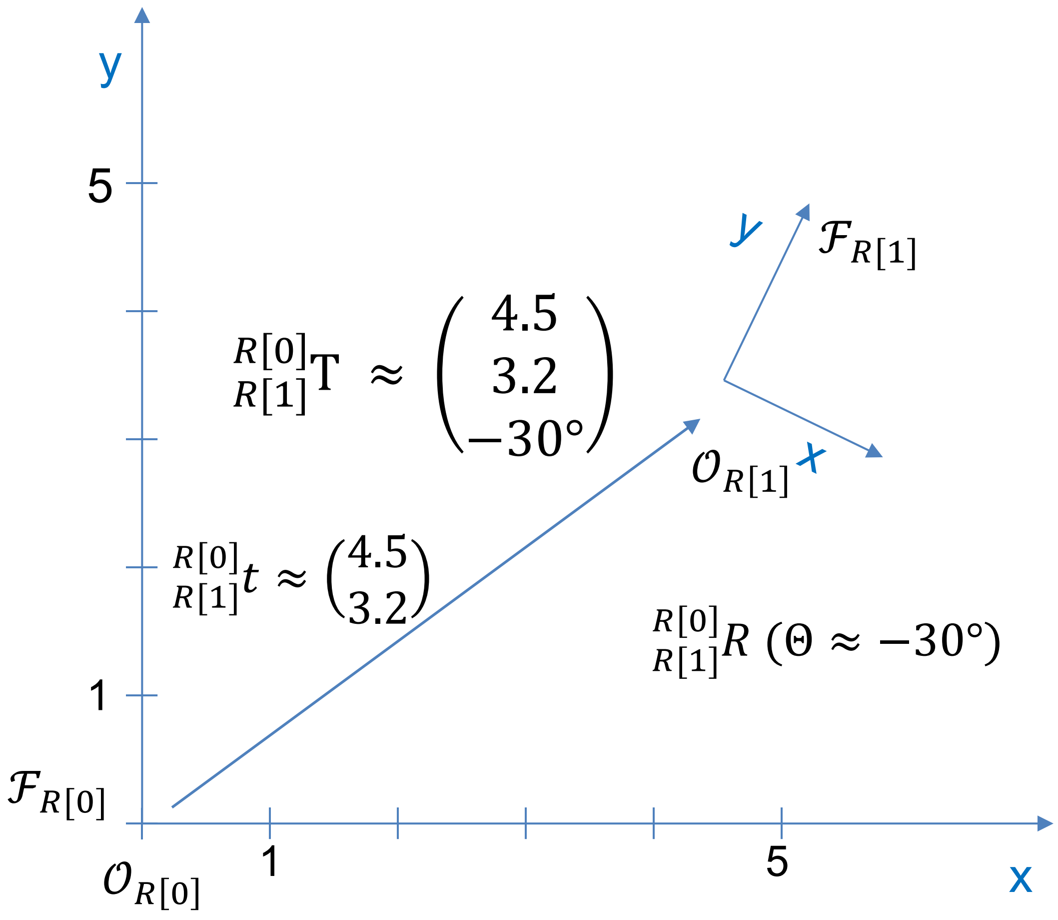
    $\triangleq \ {}^{R[X]}_{R[X+1]}R$: **Rotation** of $\mathcal{F}_{R[X+1]}$ wrt. $\mathcal{F}_{R[X]}$

Transform



- $^{R[X]}_{R[X+1]}t$: **Translation**
  - Position vector (x, y) of $R[X+1]$ wrt. $R[X]$
- $^{R[X]}_{R[X+1]}R$: **Rotation**
  - Angle (Θ) of $R[X+1]$ wrt. $R[X]$
- **Transform:** $^{R[X]}_{R[X+1]}\mathrm{T} \equiv \begin{Bmatrix} ^{R[X]}_{R[X+1]}t \\ ^{R[X]}_{R[X+1]}R \end{Bmatrix}$

Geometry approach to Odometry



We want to know:

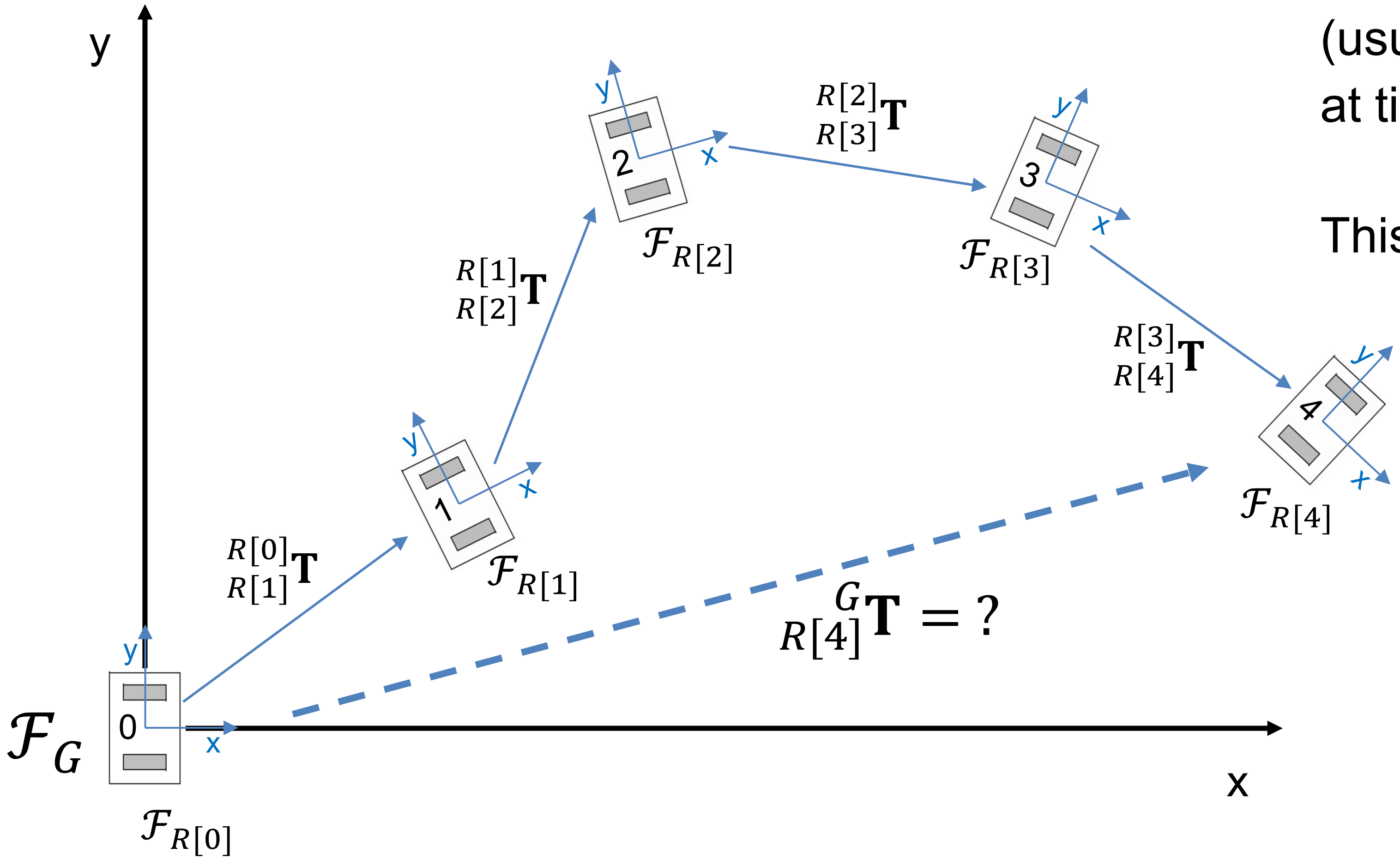
- Position of the robot (x, y)
- Orientation of the robot (Θ)
- => together: Pose $\begin{pmatrix} x \\ y \\ \Theta \end{pmatrix}$

With respect to (wrt.) $\mathcal{F}_G$: The global frame; global coordinate system

$$\mathcal{F}_{R[0]} = \mathcal{F}_G \Rightarrow \ {}^{G}\mathcal{F}_{R[0]} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$${}^{G}\mathcal{F}_{R[1]} = {}^{R[0]}_{R[1]}\mathrm{T} \approx \begin{pmatrix} 4.5 \\ 3.2 \\ 30° \end{pmatrix}$$

Mathematical approach: Transforms



Where is the Robot now?

The pose of $\mathcal{F}_{R[X]}$ with respect to $\mathcal{F}_G$ (usually = $\mathcal{F}_{R[0]}$) is the pose of the robot at time X.

This is equivalent to $^{G}_{R[X]}\mathbf{T}$

Chaining of Transforms

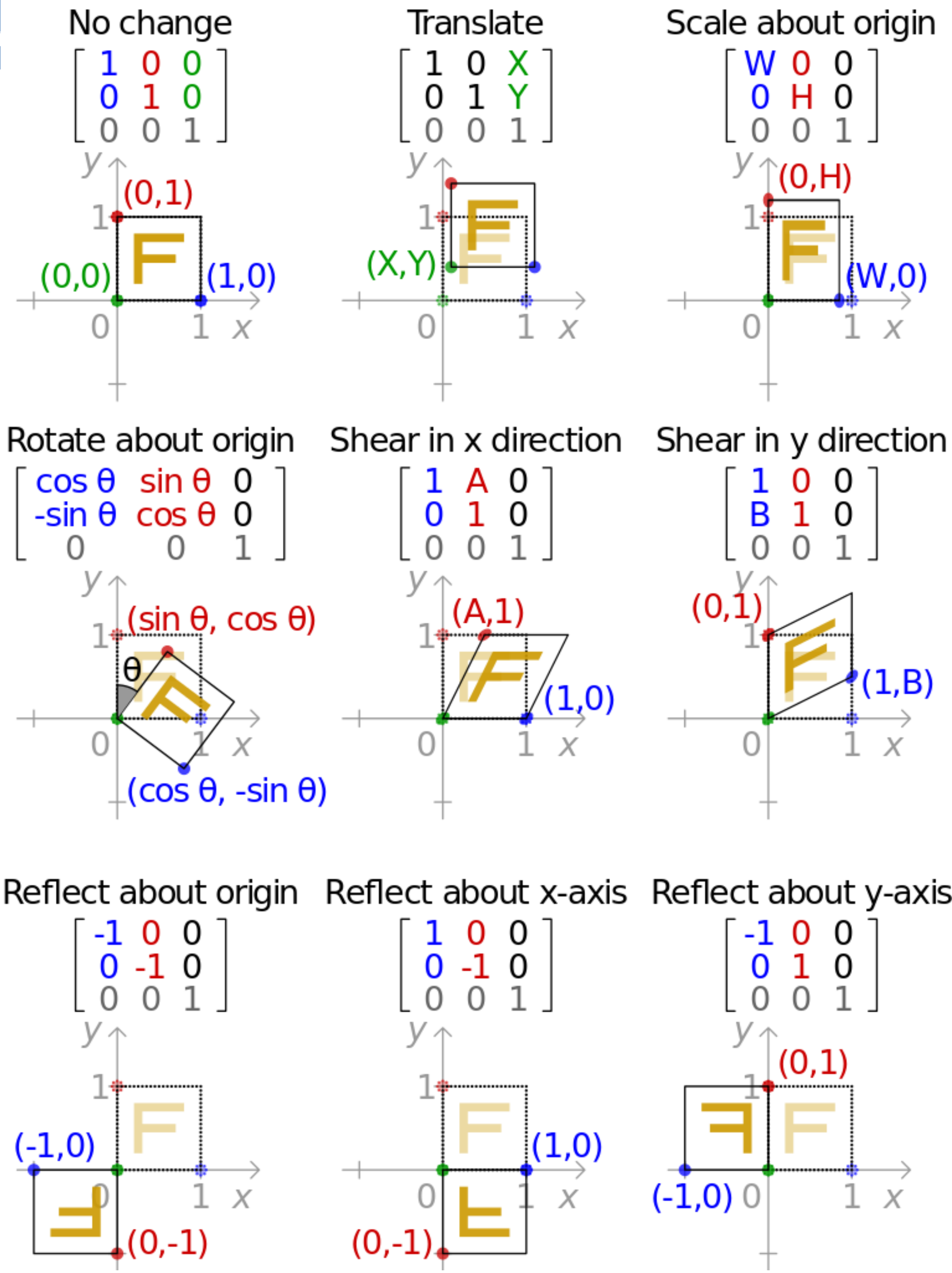
$$^{G}_{R[X+1]}\mathbf{T} = {}^{G}_{R[X]}\mathbf{T} \; {}^{R[X]}_{R[X+1]}\mathbf{T}$$

often: $\mathcal{F}_G \equiv \mathcal{F}_{R[0]} \Rightarrow {}^{G}_{R[0]}\mathbf{T} = id$

TRANSFORMS & STUFF ☺

Affine Transformation

- Function between affine spaces. Preserves:
  - points,
  - straight lines
  - planes
  - sets of parallel lines remain parallel
- Allows:
  - Interesting for Robotics: translation, rotation, (scaling), and chaining of those
  - Not so interesting for Robotics: reflection, shearing, homothetic transforms
- Rotation and Translation: $\begin{bmatrix} \cos\theta & \sin\theta & X \\ -\sin\theta & \cos\theta & Y \\ 0 & 0 & 1 \end{bmatrix}$



Math: Rigid Transformation

- Geometric transformation that preserves Euclidean distance between pairs of points.
- Includes reflections (i.e. change from right-hand to left-hand coorinate system and back)
- Just rotation & translation: rigid motions or proper rigid transformations:
  - Decomposed to rotation and translation
  - => subset of Affine Transofrmations

- In Robotics: Just use term **Transform** or **Transformation** for rigid motions (without reflections)

Lie groups for transformations

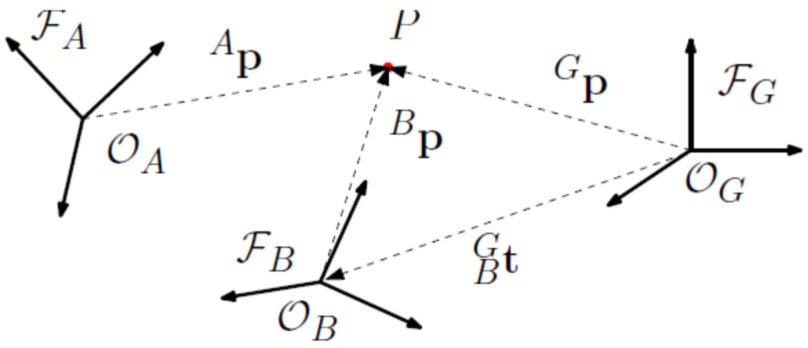
- Smoothly differentiable Group
- No singularities
- Good interpolation

- SO: Special Orthorgonal group
- SE: Special Euclidian group
- Sim_ilarity transform group

Group	Description	Dim.	Matrix Representation
SO(3)	3D Rotations	3	3D rotation matrix
SE(3)	3D Rigid transformations	6	Linear transformation on homogeneous 4-vectors
SO(2)	2D Rotations	1	2D rotation matrix
SE(2)	2D Rigid transformations	3	Linear transformation on homogeneous 3-vectors
Sim(3)	3D Similarity transformations (rigid motion + scale)	7	Linear transformation on homogeneous 4-vectors

http://ethaneade.com/lie.pdf

Transform



Notation	Meaning
$\mathcal{F}_{\mathrm{R}[k]}$	Coordinate frame attached to object 'R' (usually the robot) at sample time-instant k.
$\mathcal{O}_{\mathrm{R}[k]}$	Origin of $\mathcal{F}_{\mathrm{R}[k]}$.
${}^{\mathrm{R}[k]}\mathbf{p}$	For any general point P, the position vector $\overrightarrow{\mathcal{O}_{\mathrm{R}[k]}P}$ resolved in $\mathcal{F}_{\mathrm{R}[k]}$.
${}^{\mathrm{H}}\hat{\mathbf{x}}_{\mathrm{R}}$	The x-axis direction of $\mathcal{F}_{\mathrm{R}}$ resolved in $\mathcal{F}_{\mathrm{H}}$. Similarly, ${}^{\mathrm{H}}\hat{\mathbf{y}}_{\mathrm{R}}$, ${}^{\mathrm{H}}\hat{\mathbf{z}}_{\mathrm{R}}$ can be defined. Obviously, ${}^{\mathrm{R}}\hat{\mathbf{x}}_{\mathrm{R}} = \hat{\mathbf{e}}_1$. Time indices can be added to the frames, if necessary.
${}^{\mathrm{R}[k]}_{\mathrm{S}[k']}\mathbf{R}$	The rotation-matrix of $\mathcal{F}_{\mathrm{S}[k']}$ with respect to $\mathcal{F}_{\mathrm{R}[k]}$.
${}^{\mathrm{R}}_{\mathrm{S}}\mathbf{t}$	The translation vector $\overrightarrow{\mathcal{O}_{\mathrm{R}}\mathcal{O}_{\mathrm{S}}}$ resolved in $\mathcal{F}_{\mathrm{R}}$.

Transform between two coordinate frames

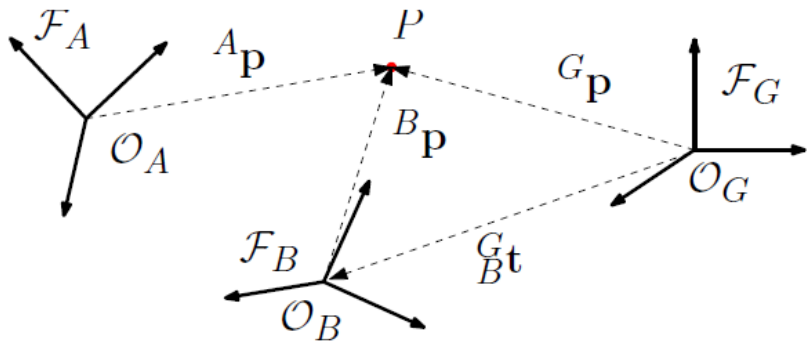
$${}^{\mathrm{G}}_{\mathrm{A}}\mathbf{t} \triangleq \overrightarrow{\mathcal{O}_{\mathrm{G}}\mathcal{O}_{\mathrm{A}}} \text{ resolved in } \mathcal{F}_{\mathrm{G}}$$

$$\begin{pmatrix} {}^{\mathrm{G}}\mathbf{p} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{R} & {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{t} \\ \mathbf{0}_{1\times[2,3]} & 1 \end{pmatrix} \begin{pmatrix} {}^{\mathrm{A}}\mathbf{p} \\ 1 \end{pmatrix} \qquad {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{T} \equiv \left\{ \begin{matrix} {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{t} \\ {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{R} \end{matrix} \right\}$$

$$\begin{aligned} {}^{\mathrm{G}}\mathbf{p} &= {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{R}\; {}^{\mathrm{A}}\mathbf{p} + {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{t} \\ &\triangleq {}^{\mathrm{G}}_{\mathrm{A}}\mathbf{T}({}^{\mathrm{A}}\mathbf{p}). \end{aligned}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & {}^{\mathrm{G}}_{\mathrm{A}}t_x \\ \sin\theta & \cos\theta & {}^{\mathrm{G}}_{\mathrm{A}}t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Transform: Operations



Transform between two coordinate frames (chaining, compounding):

$${}^{G}_{B}\mathbf{T} = {}^{G}_{A}\mathbf{T}\,{}^{A}_{B}\mathbf{T} \equiv \begin{Bmatrix} {}^{G}_{A}\mathbf{R}\,{}^{A}_{B}\mathbf{t} + {}^{G}_{A}\mathbf{t} \\ {}^{G}_{A}\mathbf{R}\,{}^{A}_{B}\mathbf{R} \end{Bmatrix}$$

Inverse of a Transform :

$${}^{B}_{A}\mathbf{T} = {}^{A}_{B}\mathbf{T}^{-1} \equiv \begin{Bmatrix} -{}^{A}_{B}\mathbf{R}^{\mathsf{T}}\,{}^{A}_{B}\mathbf{t} \\ {}^{A}_{B}\mathbf{R}^{\mathsf{T}} \end{Bmatrix}$$

Relative (Difference) Transform :

$${}^{B}_{A}\mathbf{T} = {}^{G}_{B}\mathbf{T}^{-1}\,{}^{G}_{A}\mathbf{T}$$

See: **Quick Reference to Geometric Transforms in Robotics** by Kaustubh Pathak on the webpage!

Chaining :

$$ {}^{G}_{R[X+1]}\mathbf{T} = {}^{G}_{R[X]}\mathbf{T}\ {}^{R[X]}_{R[X+1]}\mathbf{T} \equiv \begin{Bmatrix} {}^{G}_{R[X]}\mathrm{R}\ {}^{R[X]}_{R[X+1]}t + {}^{G}_{R[X]}t \\ {}^{G}_{R[X]}\mathrm{R}\ {}^{R[X]}_{R[X+1]}\mathrm{R} \end{Bmatrix} = \begin{Bmatrix} {}^{G}_{R[X+1]}t \\ {}^{G}_{R[X+1]}\mathrm{R} \end{Bmatrix} $$

In 2D Translation:

$$ \begin{bmatrix} {}^{G}_{R[X+1]}t_x \\ {}^{G}_{R[X+1]}t_y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos {}^{G}_{R[X]}\theta & -\sin {}^{G}_{R[X]}\theta & {}^{G}_{R[X]}\mathrm{t_x} \\ \sin {}^{G}_{R[X]}\theta & \cos {}^{G}_{R[X]}\theta & {}^{G}_{R[X]}\mathrm{t_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{R[X]}_{R[X+1]}\mathrm{t_x} \\ {}^{R[X]}_{R[X+1]}t_y \\ 1 \end{bmatrix} $$

In 2D Rotation:

$$ {}^{G}_{R[X+1]}\mathrm{R} = \begin{bmatrix} \cos {}^{G}_{R[X+1]}\theta & -\sin {}^{G}_{R[X+1]}\theta \\ \sin {}^{G}_{R[X+1]}\theta & \cos {}^{G}_{R[X+1]}\theta \end{bmatrix} = \begin{bmatrix} \cos {}^{G}_{R[X]}\theta & -\sin {}^{G}_{R[X]}\theta \\ \sin {}^{G}_{R[X]}\theta & \cos {}^{G}_{R[X]}\theta \end{bmatrix} \begin{bmatrix} \cos {}^{R[X]}_{R[X+1]}\theta & -\sin {}^{R[X]}_{R[X+1]}\theta \\ \sin {}^{R[X]}_{R[X+1]}\theta & \cos {}^{R[X]}_{R[X+1]}\theta \end{bmatrix} $$

In 2D Rotation (simple):

$$ {}^{G}_{R[X+1]}\theta = {}^{G}_{R[X]}\theta + {}^{R[X]}_{R[X+1]}\theta $$

In ROS: nav_2d_msgs/Pose2DStamped

- First Message at time 97 : G
- Message at time 103 : X
- Next Message at time 107 : X+1

$$ {}^{G}_{R[X+1]}\mathbf{T} = {}^{G}_{R[X]}\mathbf{T} \; {}^{R[X]}_{R[X+1]}\mathbf{T} $$

$$ {}^{R[X]}_{R[X+1]}t_x $$

$$ {}^{R[X]}_{R[X+1]}t_y $$

$$ {}^{R[X]}_{R[X+1]}\Theta $$

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
geometry_msgs/Pose2D pose2D
  float64 x
  float64 y
  float64 theta
```