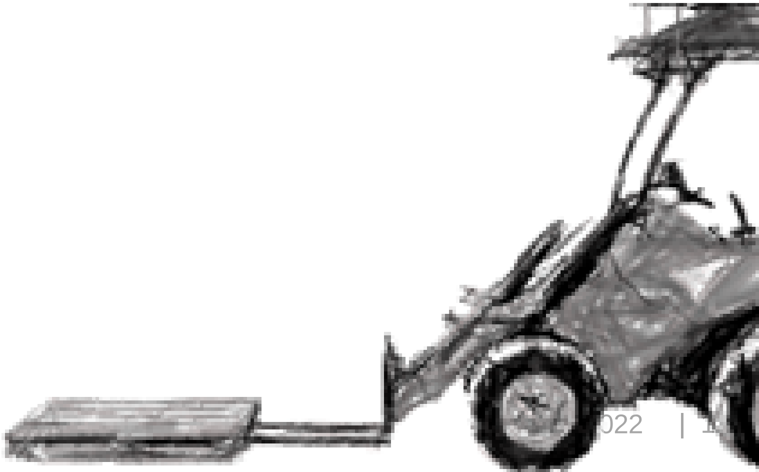
Tampere University

Introduction to and review of some general concepts

Reza Ghabcheloo

AUT 710



Concepts

- Coordinate frames
- Kinematics modelling, extrinsic calibration
- Dynamics systems and state space
- Probabilities and distributions
- Map, world model

Part 1

Mobile Robots vs Non mobile!
AI Robotics vs Industrial robots!

Conventional robotics: Perfect world assumption

http://www.kuka-robotics.com/

Modern needs: Dynamic and uncertain world

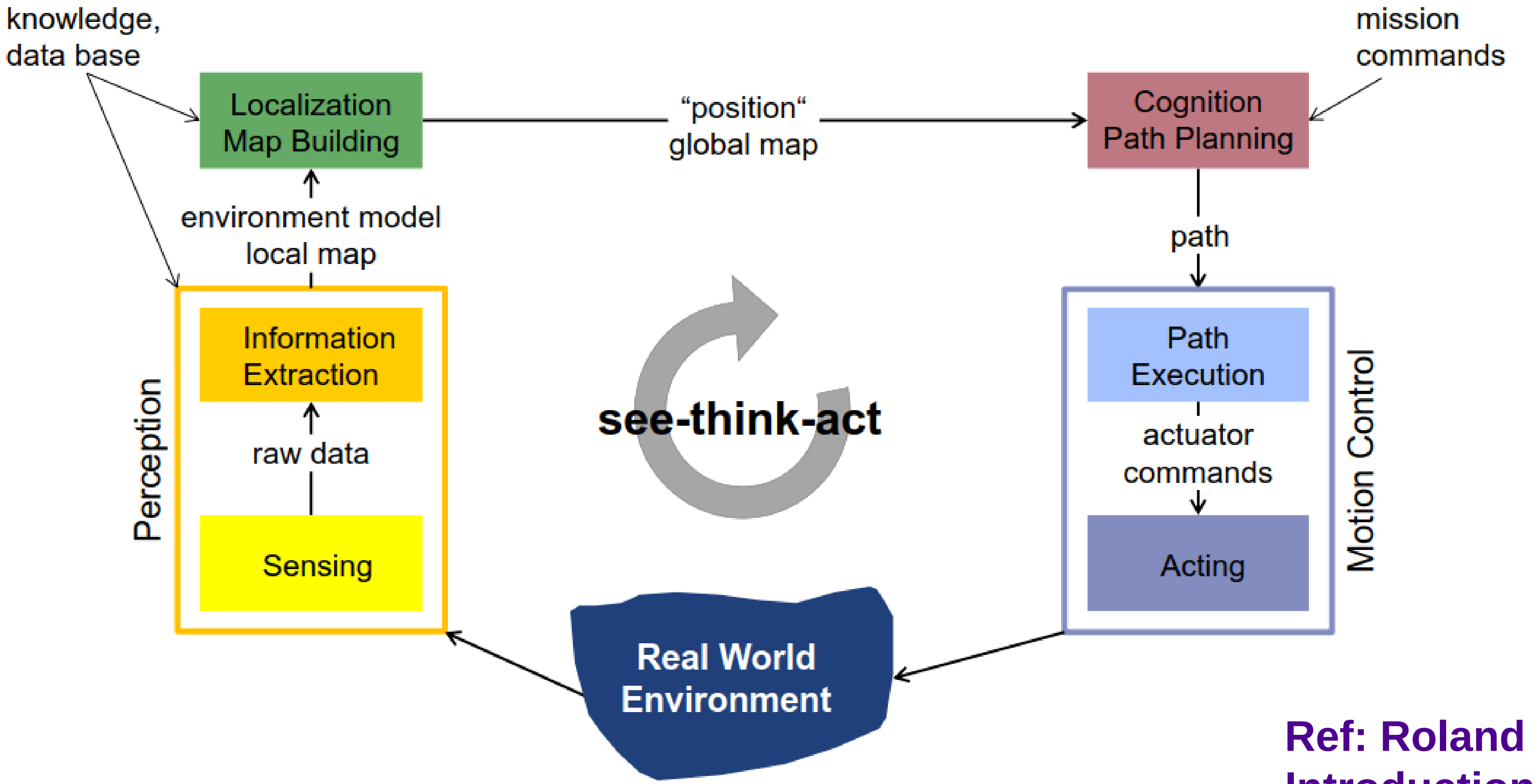
Part of industrial robotics is moving
Away from perfect world assumption

Baxter cobot

Many of things we learn in this course also apply to manipulators. Though most of our examples are related to wheeled mobile robots.

Tampere University

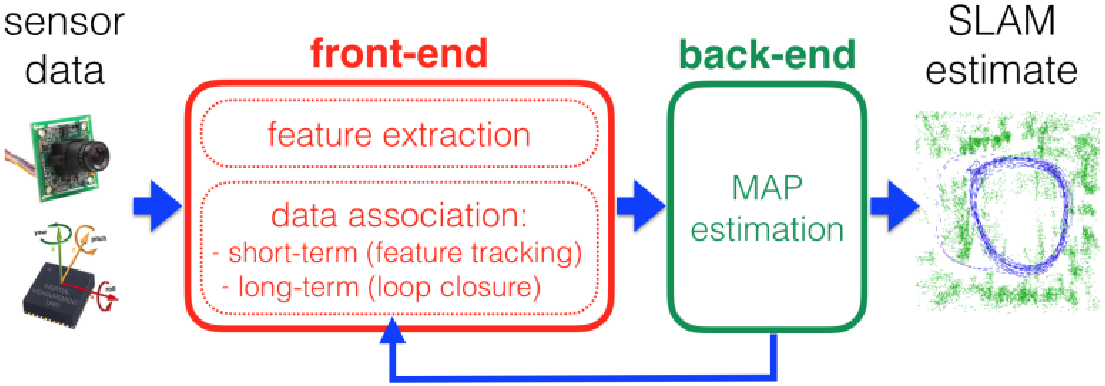
See Think Act control architecture



Ref: Roland Siegward, Introduction to Autonomous Mobile Robots

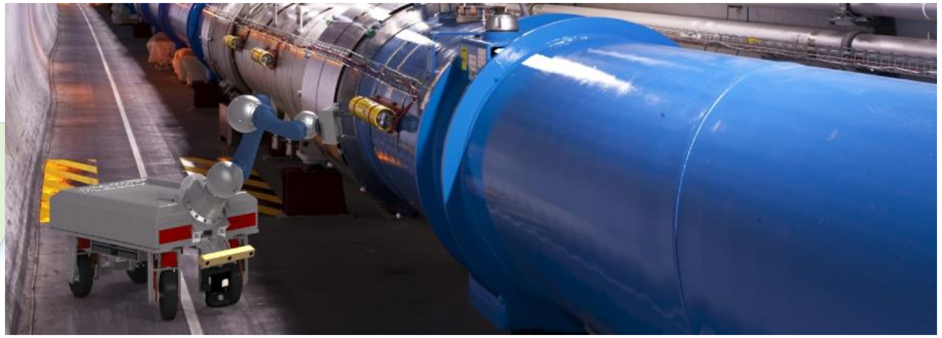
Tampereen yliopisto
Tampere University

Onboard:
Color, texture: vision, IR,…
Line, surface: Lidar, Radar, …
Compliance: IMU, wheel contact,…
Other: radio activity, gas, algae…



Pose

Sensors



World modeling

Decision making
- collision free planning
- Intervention planning
- maintenance planning
- harvesting

Motion control

Third party data

GNSS, UWB,
IMU,
Vision, Lidar,

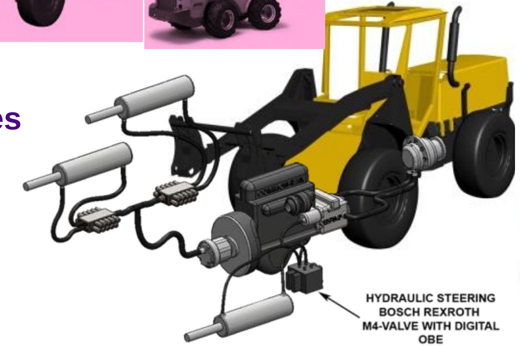
Elevation maps,
other vehicles,
Infra (weather)
sensors

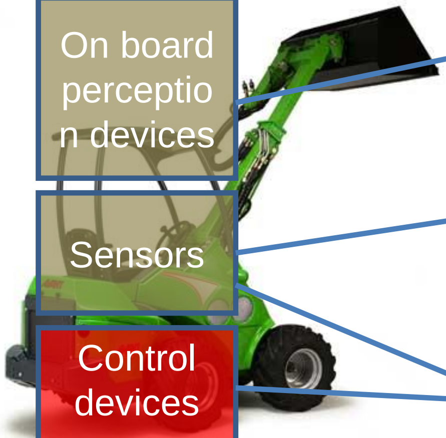
MAP> Maximum a posteriori
Bayes filter (KF, MCF)

Tampereen yliopisto
Tampere University

**TUTrim
2018**

**TAU center for
intelligent working machines**





On board perception devices

Sensors

Control devices

GUI and High level control of worksite (ROS)
-Mission planning , path planning, multi-machine operations, virtual worksite

Stationary perception devices

WLAN

RS

Communication
Router, bridge, firewall

middleware
ROS/ROS2, DDS, Non-time critical decisions, planning

Middle level control
Simulink RT, Linux RT, Motion control, state estimation

Low level control
-diesel, pump, valves control, sensors etc.

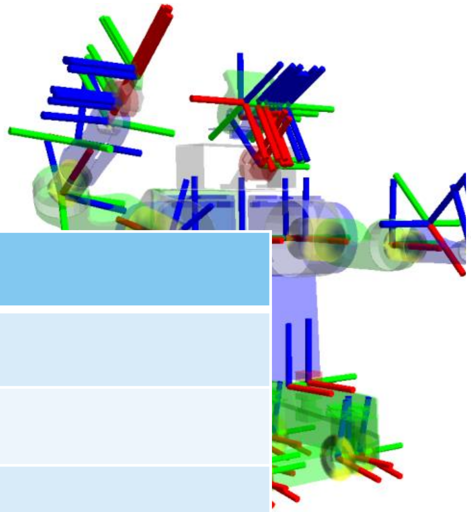
LAN

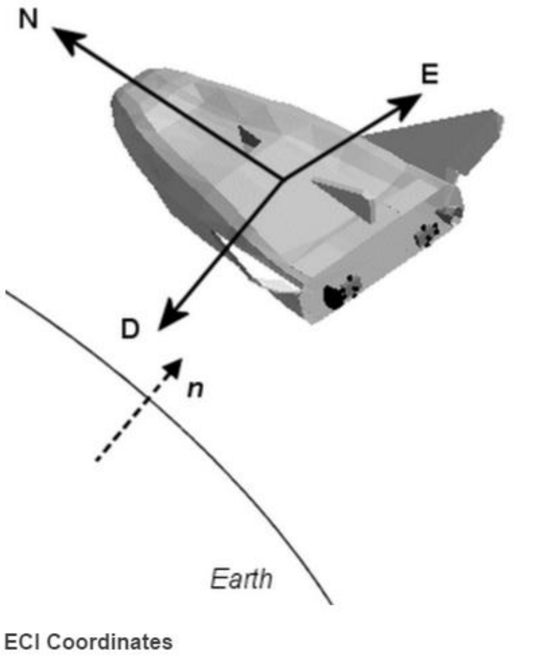
CAN

V / I
PWM

Tampere University

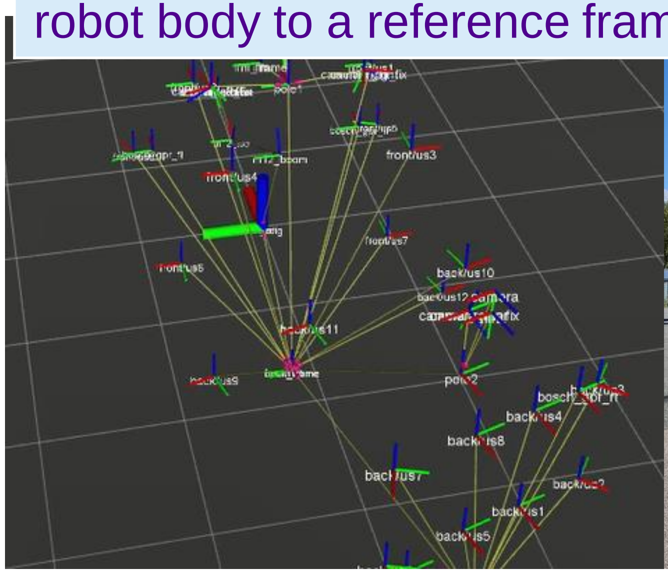
[ROS tf tree](#)





Models and coordinate frames

modelling	term
a sensor to another sensor	Extrinsic calibration
a robot link to another	Kinematics modeling
a sensor to robot body	Extrinsic calibration
a point of the environment to a sensor	Perception, object detection, object pose estimation
robot body to a reference frame	Localization / robot pose estimation



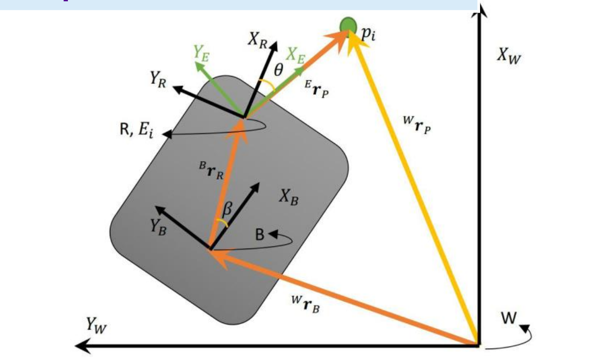
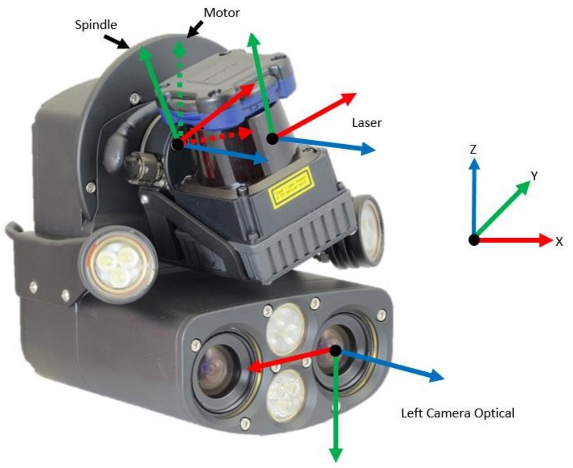
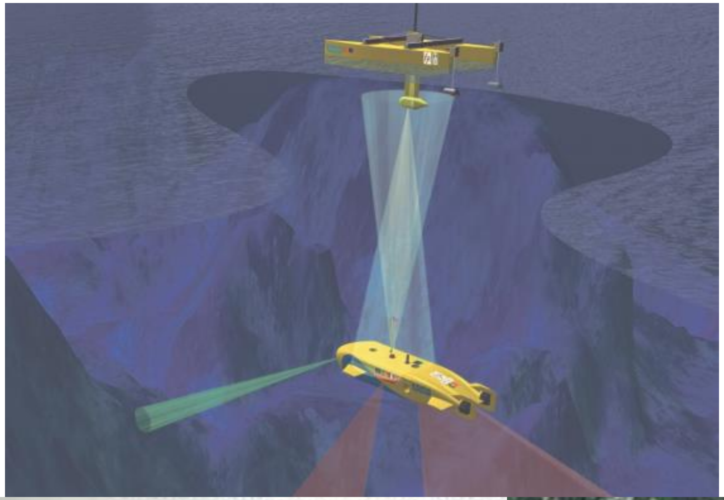
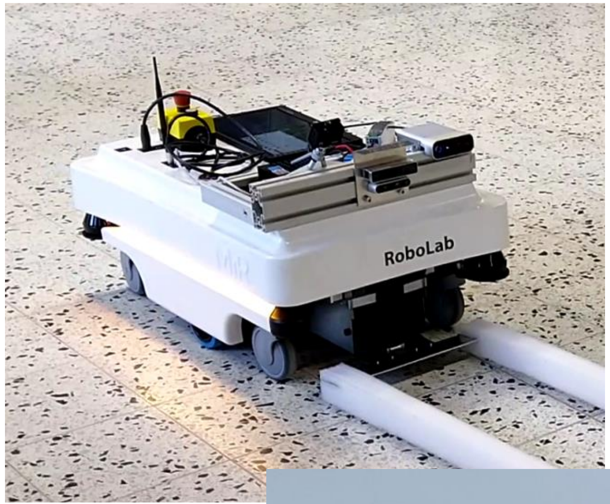


Figure 27: Point $\mathbf{p}_i$ represented in radar frame R, body frame R and world frame W



Mobile robots





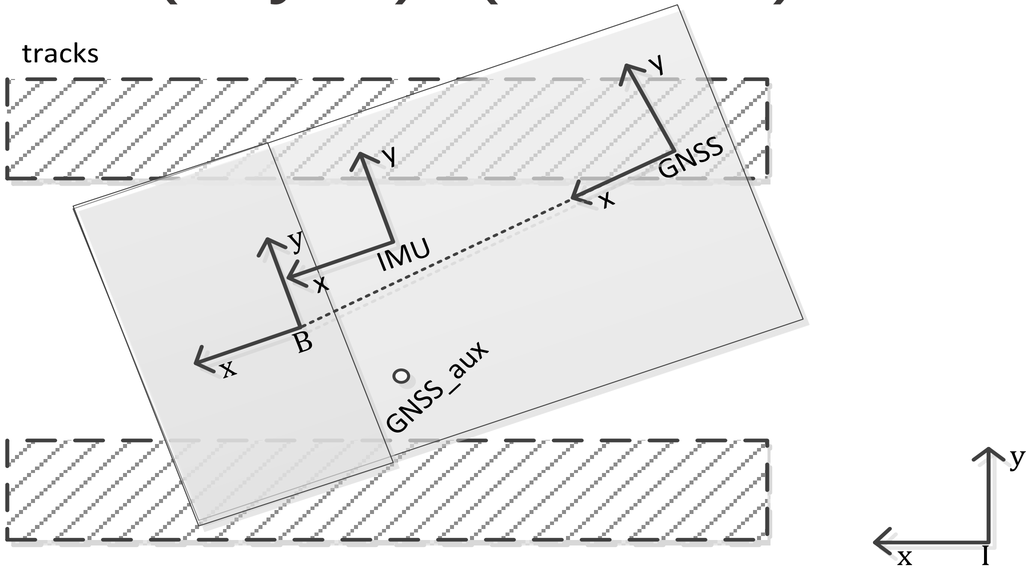
Coordinate systems

- For robots moving in small space, we usually use a simple system: a flat world with a world frame which all others are defined with respect to that or each other
- For robots moving on large field (aircrafts) flat world assumption is not useful

3D space, position

- Flat world
  - Position represented by 3 variables (x, y, z), (N, E, D), etc in distance unit
- Non flat
  - Latitude, longitude, altitude

https://novatron.fi/xsite-pro/



Q: what is the third coordinate in the diagram?

Orientation is more involved

- For frames with limited rotational motions, we may use Euler angles with 3 variables
    - Euler angles (roll, pitch, yaw)
    - For large motions, Euler angles will be ambiguous.
- Otherwise, use
    - quaternions $q = (q_1, q_2, q_3, q_4)$
    - direction cosine matrix (rotation matrix)

$$R \in SO(3), RR^T = I, \det(R) = 1$$

Transformation of points, vectors, orientation, kinematics

- For the purpose of this class, we will focus on 2D world.
- For 3D and more details recall

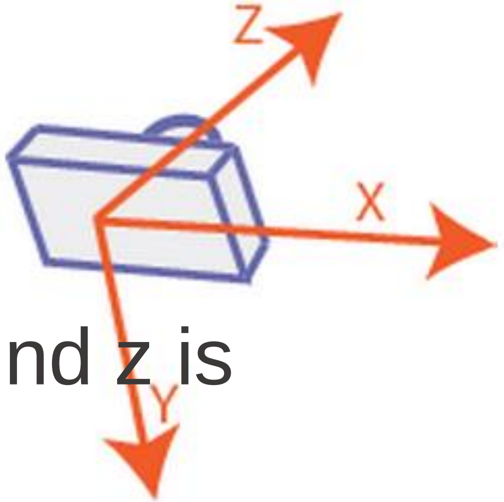
AUT.841 Robot Manipulators: Modeling, Control and Programming (previously ASE-9407)

Earth related coordinate systems

- Matlab/Simulink Aerospace blockset has a very good overview

https://se.mathworks.com/help/aeroblks/about-aerospace-coordinate-systems.html

2D world



- Right hand coordinate system: positive rotation around z is from x to y

Q: my nose pointing at X, my head pointing at Z, what is Y?

- Position: North-East-Down or X-Y-Z

Q: How would you plot NE frames in Matlab?

- Orientation: we will use 'ZYX' order of Euler angles (roll, pitch, yaw); study of robots in 2D world: $roll$=0, $pitch$ =0, ψ=heading or yaw
- In this class, we will typically use XY coordinates, such that Z points away from screen.

Q: what is positive rotation, CW or CCW?

Tampere University

System dynamics and state space

- Motion of the robots are described by **differential equations**.
- Imagine that our robot is mass m moving on a one-dimensional space (e.g. line), which can be controlled by force F. Then, the system dynamics is given by

$$m\ddot{x} = f$$

- We now rewrite this as a set of **first order differential equations** with the help of some auxiliary variables, which basically define the **system state** $(x_1, x_2) = (x, v)$.

$$\dot{x} = v$$

$$\dot{v} = \frac{1}{m} f$$

So, if we know state at initial time $x(t_0), v(t_0)$, for a given input $f(t){:}\, t \in [t_0, t_f]$, we can find state trajectories $x(t), v(t)$ by integrating above set of equations.

- We might be also describing the system dynamics using **difference equations**, which can be found by discretizing the differential equations. Simple Euler method will result in

$$\dot{x} = v \rightarrow \frac{x_{n+1} - x_n}{T_s} = v \rightarrow x_{n+1} = x_n + T_s v$$

By doing this, we only calculate (approximately) the states at $t_0 = t_0, t_1 = t_0 + T_s, t_2 = t_0 + 2T_s, \ldots$ More accurate solution can be found by Runge–Kutta for example.

Notation:

$$\dot{x} = \frac{\mathrm{d}}{\mathrm{d}t} x = v$$

$\ddot{x}$ is the second derivative or acceleration

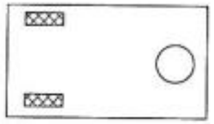
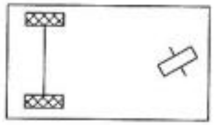
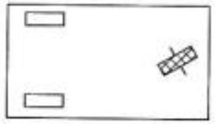
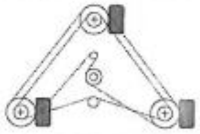
In general:

$$\dot{x}_1 = f_1(x_1, \ldots, u_1, \ldots)$$
$$\dot{x}_2 = f_2(x_1, \ldots, u_1, \ldots)$$
$$\ldots$$

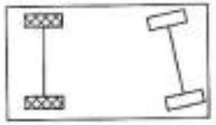
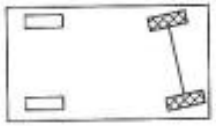
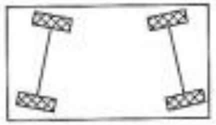
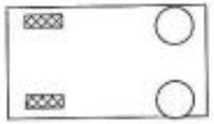
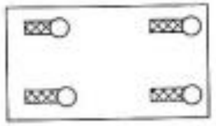
Steering and drive mechanisms

[Link]

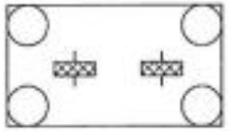
Wheel configurations for rolling vehicles

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bic
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye
3		Two-wheel centered differential drive with a third point of contact	Nor EPF
		Two independently driven wheels in the rear/front, one unpowered omnidirectional wheel in the front/rear	Mar incl Pygi
		Two connected traction wheels (differential) in rear, one steered free wheel in front	Piag
		Two free wheels in rear, one steered traction wheel in front	Nept Univ
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stan Tribo Palm (CM
		Three synchronously motorized and steered wheels; the orientation is not controllable	“Syn Denr Instit Robo

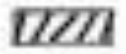
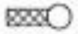
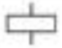
Wheel configurations for rolling vehicles

# of wheels	Arrangement	Description
4		Two motorized wheels in the rear, two steered wheels in the front; steering has to be different for the two wheels to avoid slipping/skidding.
		Two motorized and steered wheels in the front, two free wheels in the rear; steering has to be different for the two wheels to avoid slipping/skidding.
		Four steered and motorized wheels
		Two traction wheels (differential) in rear/front, two omnidirectional wheels in the front/rear
		Four omnidirectional wheels
		Two-wheel differential drive with two additional points of contact
		Four motorized and steered castor wheels

Wheel configurations for rolling vehicles

# of wheels	Arrangement	Description	Typical examples
6		Two motorized and steered wheels aligned in center, one omnidirectional wheel at each corner	First
		Two traction wheels (differential) in center, one omnidirectional wheel at each corner	Terregator (Carnegie Mellon University)

Icons for the each wheel type are as follows:

	unpowered omnidirectional wheel (spherical, castor, Swedish)
	motorized Swedish wheel (Stanford wheel)
	unpowered standard wheel
	motorized standard wheel
	motorized and steered castor wheel
	steered standard wheel
	connected wheels

Ref: Roland Siegward, Introduction to Autonomous Mobile Robots

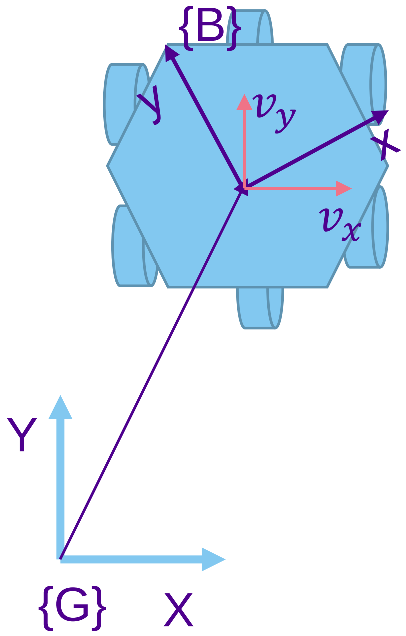
Omni-directional robot kinematics

- If we only concern ourselves with kinematics, then the robot state is described by robot pose (x, y, ψ) in 2D world.
- Omni-directional robot equation of motion

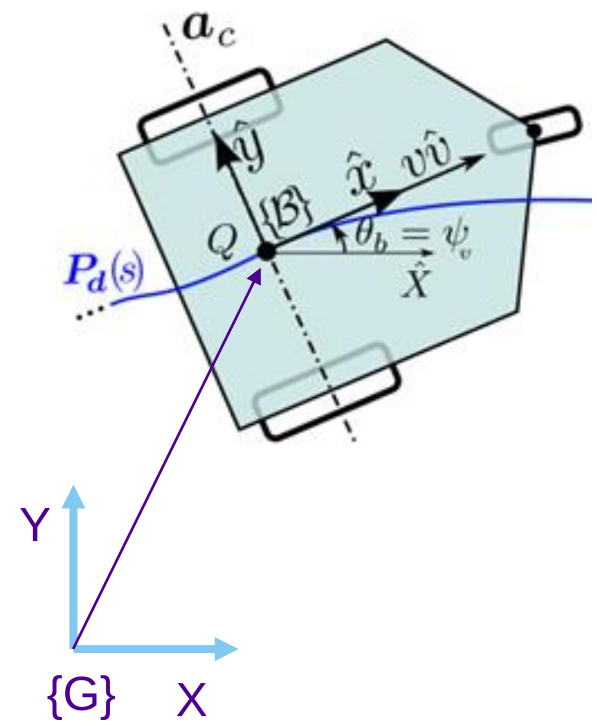
$$\dot{x} = v_{xG}$$
$$\dot{y} = v_{yG}$$
$$\dot{\psi} = \omega_z$$

Three independently controlled servos (v_x, v_y, ω_z), defined in global reference frame {G}.

- Link



Differential drive kinematics (unicycle model)

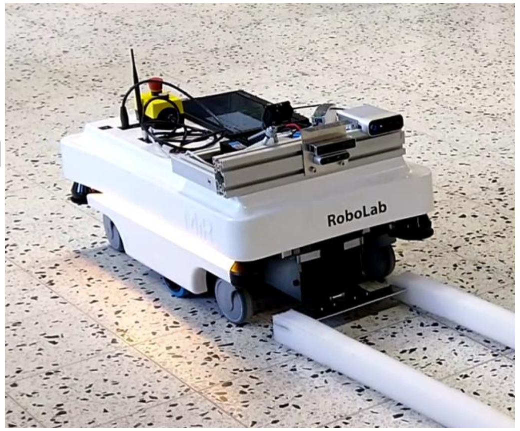


Equations of motion is given by

$$\dot{x} = v_{xB} \cos\psi$$
$$\dot{y} = v_{xB} \sin\psi$$
$$\dot{\psi} = \omega_z$$

where (v_x,ω_z) are independent inputs, defined in body frame {B}.

We will use these equations for both **control** and **localization**



- Control: v_x and ω_z are robot control commands (twist cmd)
- Localization: knowing v_x and ω_z and initial conditions $x_B(0), y_B(0), \psi(0)$, integrate the kinematic equations to get $x_B(t), y_B(t), \psi(t)$ (*dead-reckoning*)