

Autonomous Mobile Robots



Turtlebot 2



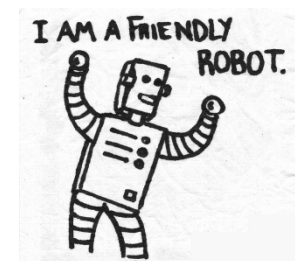
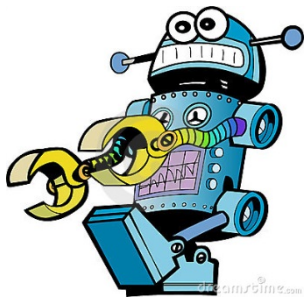
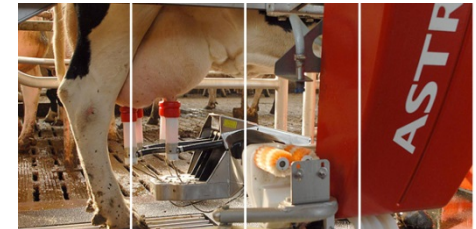
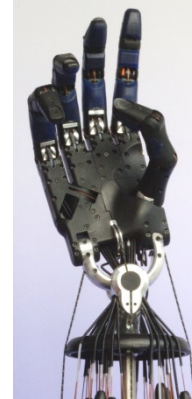
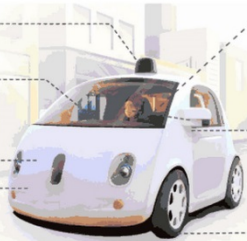
Course Intro

- Lecturers
 - Peter Ahrendt and Mads Dyrmann
- Course content
 - Robotics theory + Matlab exercises
 - Robotic software development + Turtlebot exercises
- Group exam based on final report (passed/not passed)
- 30 min. oral defence (robot demo+presentation+questions)

What is robotics ?

Key facts about the vehicle

- Sensors and hardware components that have been custom-built for self-driving
- New technologies to protect pedestrians, including a flexible windscreen and front made of a foam-like material
- An electric battery
- Speed capped at 25 mph
- Inside:
 - Seats for two passengers and a space for their belongings
 - A button to start or pull over, and an emergency stop button
 - A screen showing the route
- Software designed to drive from point A to point B without requiring any human intervention
- Primary and backup systems for steering and braking



Robots

- One definition :

*"A goal-oriented machine that can
sense, plan and act"*

- Robota – (slave) work, hard work
 - Dirty, Dangerous and Dull (?!)

Industrial robots



Autonomous Mobile Robots



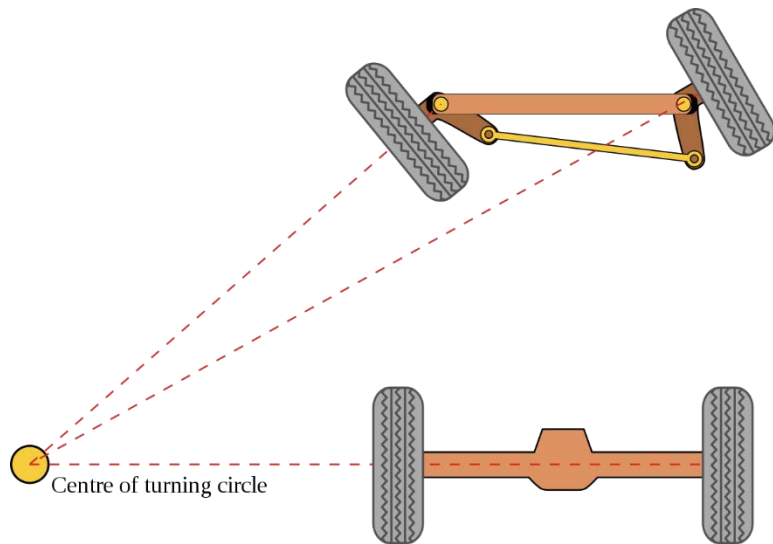
Turtlebot



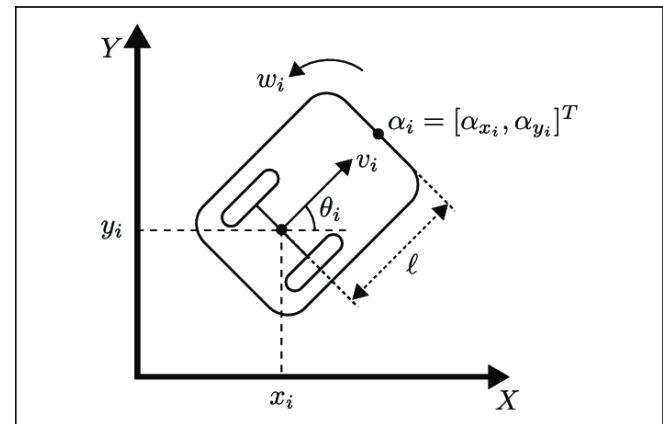
Autonomous Mobile Robots - overview

- Kinematics / dynamics
- Motion planning / navigation
 - Map-based planning / trajectory tracking
 - Motion control
- Localization
- Map building and updating
- Sensors and actuators
 - Vision-based, IMUs, tactile, ..

Kinematic models

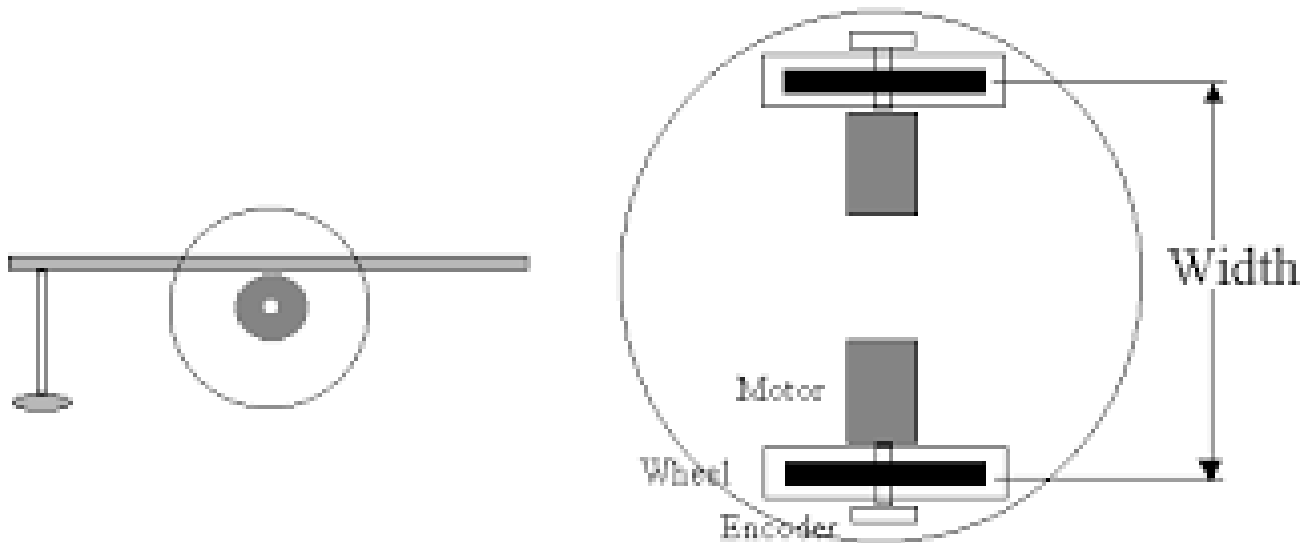


Ackermann steering



Differential drive steering

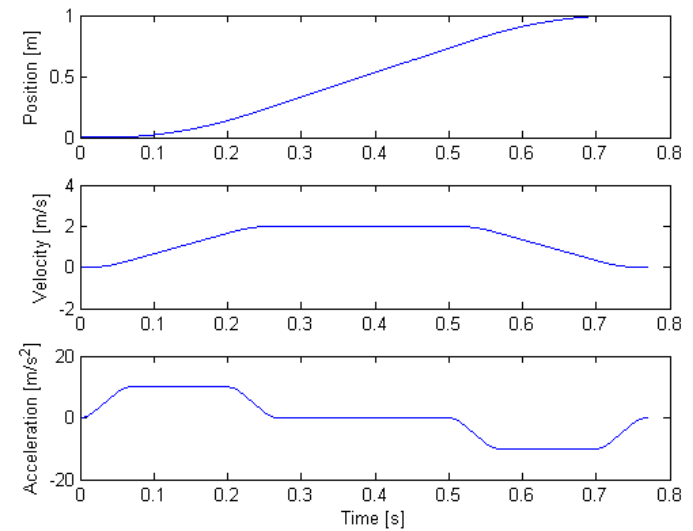
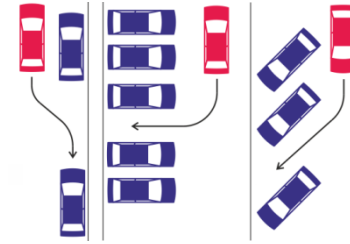
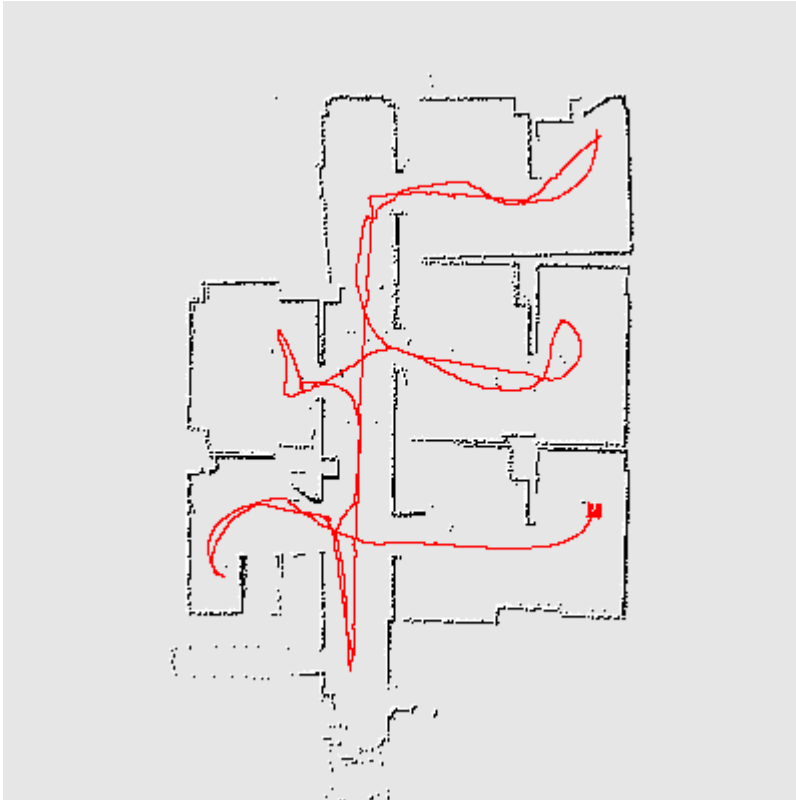
Turtlebot kinematics and dynamics



Path vs. trajectory

- Path - List of positions
 - $[[x_1, y_1], [x_2, y_2], \dots]$
- Trajectory – List of positions with time information
 - $[[t_1, x_1, y_1], [t_2, x_2, y_2], \dots]$
- Note – Pose for turtlebot is $[x, y, \theta]$ (in world coordinates)

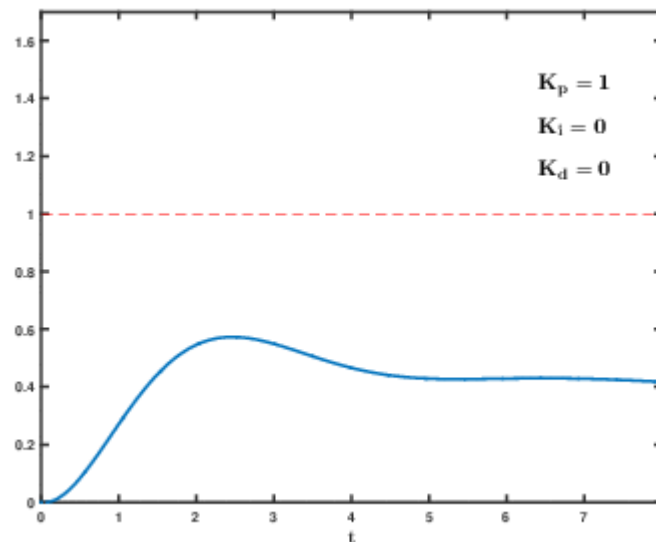
Trajectory generation



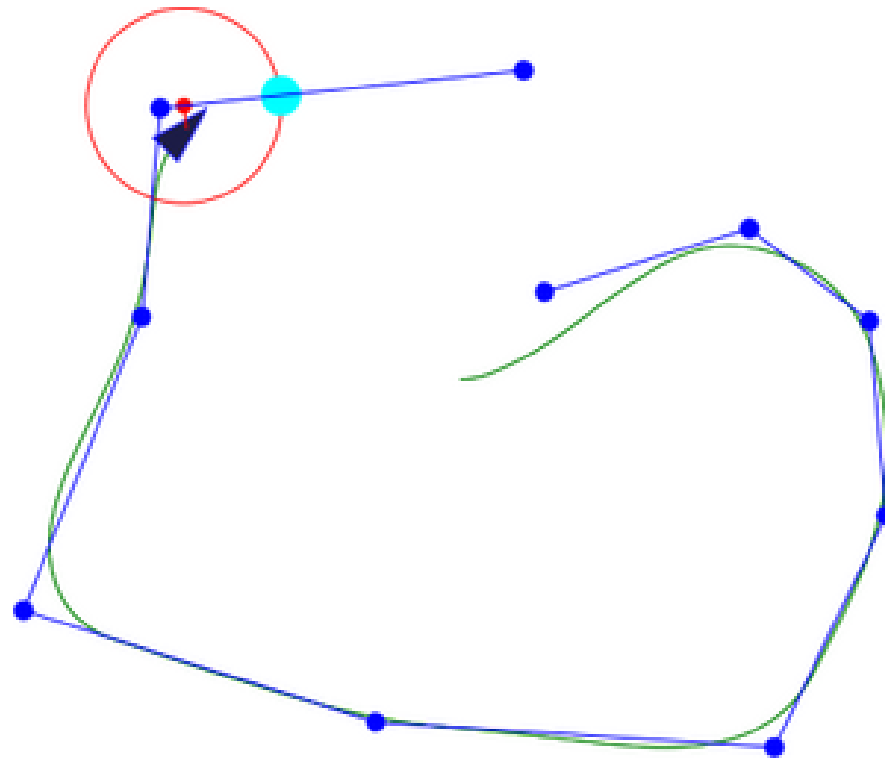
Path tracking – PID basics

- PID = Proportional-Integral-derivative

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$



Pure Pursuit algorithm



<http://se.mathworks.com/help/robotics/ug/pure-pursuit-controller.html>

Exercises

- Form groups of 3-4 persons (ideally multi-disciplinary)
- Download and examine Peter Corke's Robotics Toolbox and Machine Vision Toolbox (<http://www.petercorke.com/Toolboxes.html>)
- Have a look at Mathworks Robotics Systems Toolbox (<http://se.mathworks.com/help/robotics/index.html>)
- BB exercises (including 1 mandatory!)