Universität Bremen

Institute for Artificial Intelligence

Faculty 03

Mathematics &

Computer Science

Robot Programming with ROS

6. Navigation

Arthur Niedźwiecki, Stefan Eirich

30th Nov. 2023

Outline

Autonomous Driving (2005)

https://youtu.be/7a6GrKqOxeU

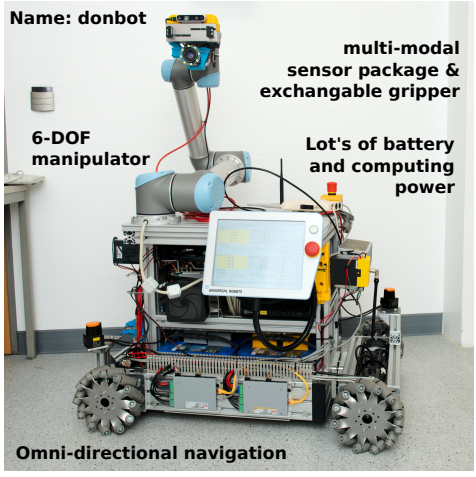

22
22
https://youtu.be/7a6GrKqOxeU

Mobile Manipulation (2012)


3x
0:44 / 2:58
3D

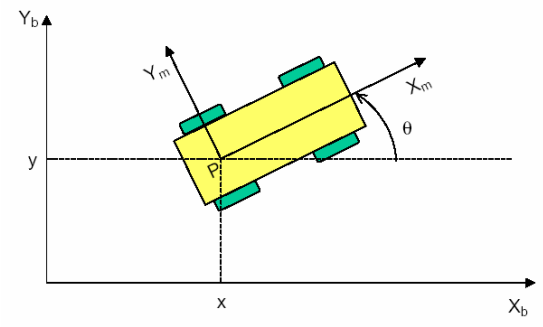
Outline

Robot Capabilities



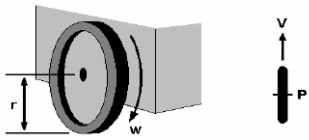
Robot Locomotion - Wheeled Locomotion

Goal: Bring the robot to a desired pose (x, y, θ):
$\Rightarrow$ 3 DOF (typically, with **non-holonomic constraints**)

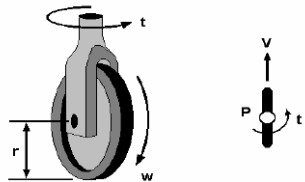


Robot Locomotion - Wheel Types

Fixed wheel

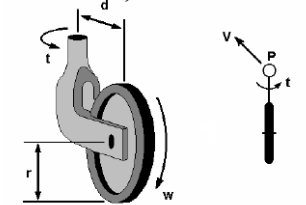


Centered orientable wheel

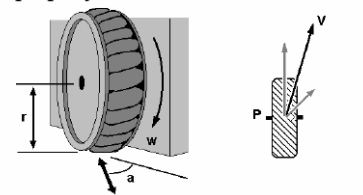


Off-centered orientable wheel

(Castor wheel)

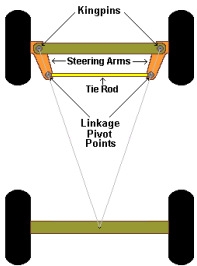


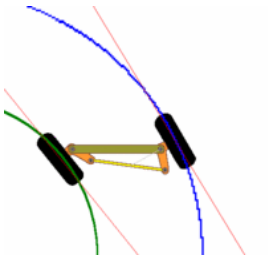
Swedish wheel: omnidirectional property



Robot Locomotion - Ackerman steering

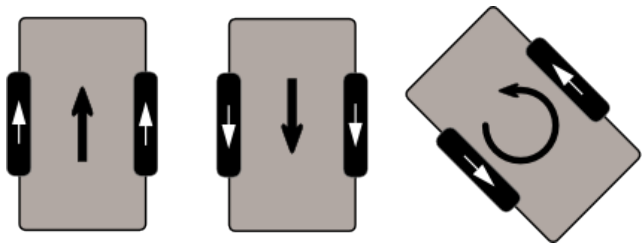
- Car-like steering
+ Robust
+ Outer wheels moves on a circle of different radius than inner wheel
– But hard to control (parking!)





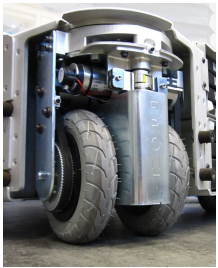
Robot Locomotion - Differential-Drive

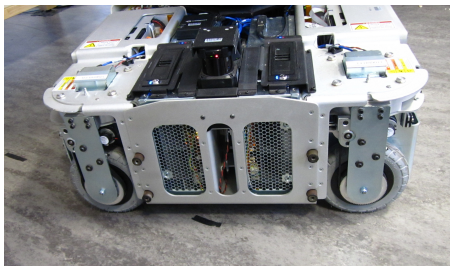
+ Turns on spot
+ Good choice for round robots
+ Parking is easier
– Cannot move sidewards



Robot Locomotion - Turnable wheels

+ Omnidirectional (can drive forwards, sideways and turn)
– On change of direction, requires 'reconfiguration' of its wheels.
→ Controllers should not oscillate

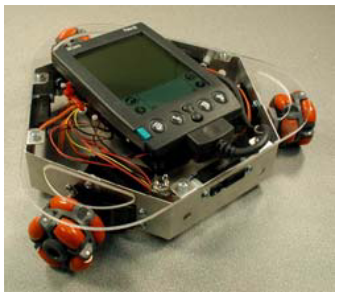


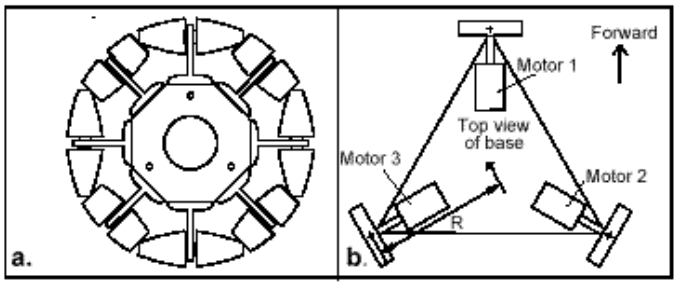


PR2: Double wheel construction to reduce friction while turning the wheel

Robot Locomotion - Omniwheels

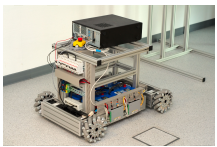
+ Omnidirectional (can drive forwards, sideways and turn)
• Wheels have free rollers at **90^o**
+ Three wheels are enough
– Hard to make them run smooth

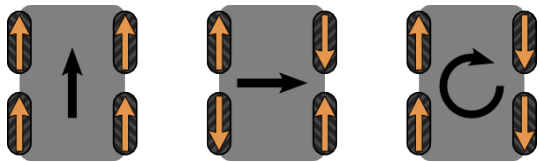




Robot Locomotion - Mecanum-Wheels

+ Omnidirectional (can drive forwards, sideways and turn)
• Wheels have free rollers at **45**^{o}
+ No reconfiguration is involved
– Depending on wheels, requires flat ground





Linearity $\Rightarrow$
A (linear) combination of cartesian movements can be achieved with the linear combination of the respective wheel velocities.

Robot Locomotion - Issue: Dead Reckoning

Dead Reckoning

”In navigation, dead reckoning is the process of calculating one’s current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course.”
https://en.wikipedia.org/wiki/Dead_reckoning

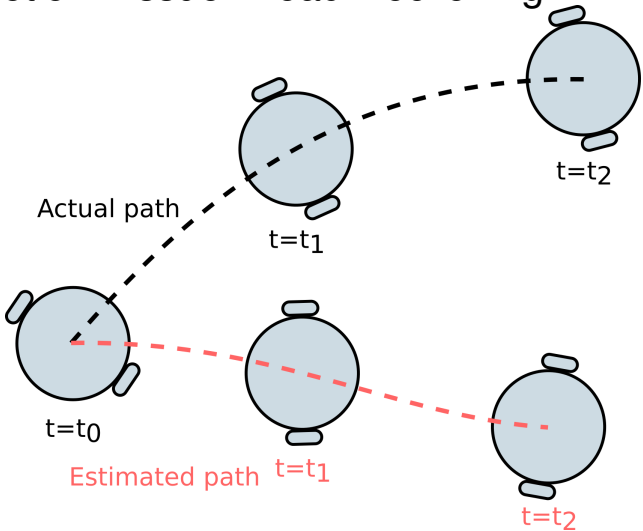
Robot Locomotion - Issue: Dead Reckoning

Dead Reckoning

”In navigation, dead reckoning is the process of calculating one’s current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course.”
https://en.wikipedia.org/wiki/Dead_reckoning

tl;dr: calculating position based on estimating direction and distance traveled (instead of using landmarks)

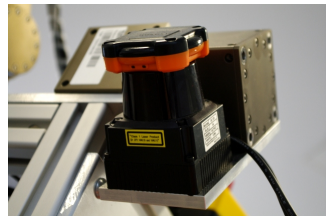
Robot Locomotion - Issue: Dead Reckoning



Robot Sensing - Time-of-Flight Sensors

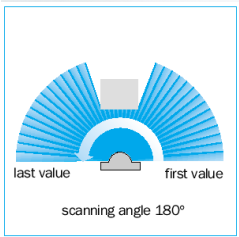
- Measurement principle: send out wave pulses, wait for the echo, and compute distance by *time of flight*
- *Same principle used by bats, dolphins, RADAR and the police...*

Robot Sensing - Laser scanners

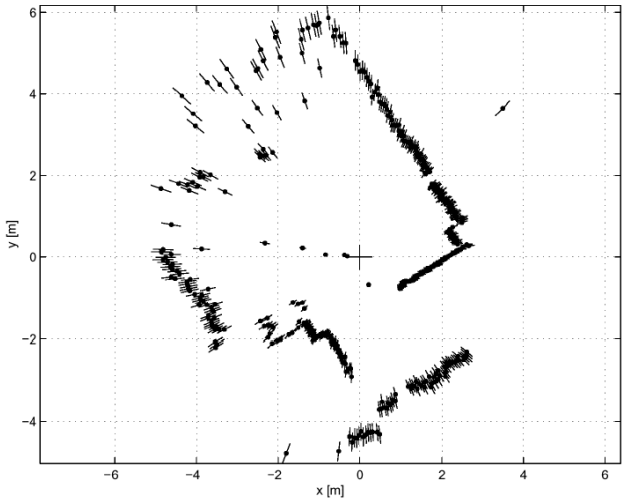
- Principle: send beam of light, beam hits target, measure time between beam transmission and reception of backscatter
- Rotating mirror deflects beam $\rightarrow$ 2D Scanner
- If the round trip time is t, the distance is $d = (c \cdot t)/2$.
- Time t is very short $\rightarrow$ use phase difference instead

Robot Sensing - Typical Laser Range Scanner



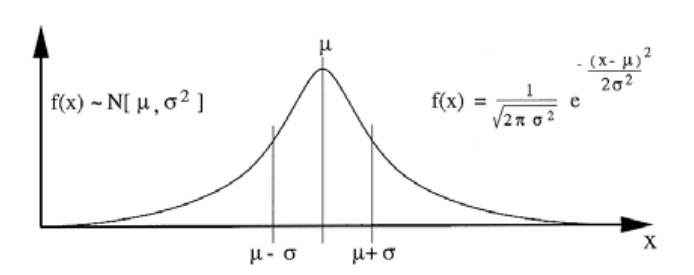
- Scanning angle: 180 degrees
- Resolution: 0.25deg, 0.5deg, or 1deg.
- Typical detection range: 30m (max. 80m)
- Data Received: Angle + Distance
- Normally used for:
  - Making maps of the environment
  - Localization of the robot
  - Tracking of objects or people
- In some circles, it is known as LIDAR (LIght Detection And Ranging)

Robot Sensing



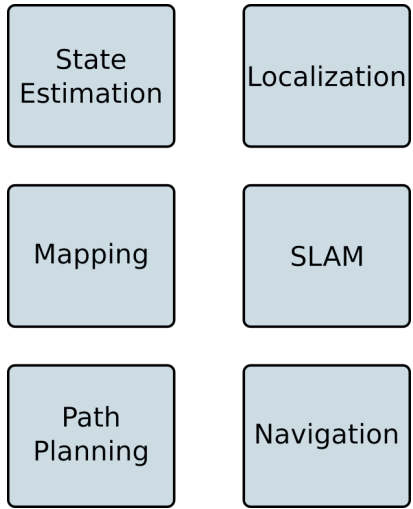
Robot Sensing - Issue: Sensor Noise

- Anything that obscures a signal.
- External noise
  - Part of the environment, e.g. temperature, electromagnetic interference, sun light, gravitational flux, or ...
- Internal noise
  - White noise (uniform), e.g. thermal noise
- Often estimated with a Normal distribution

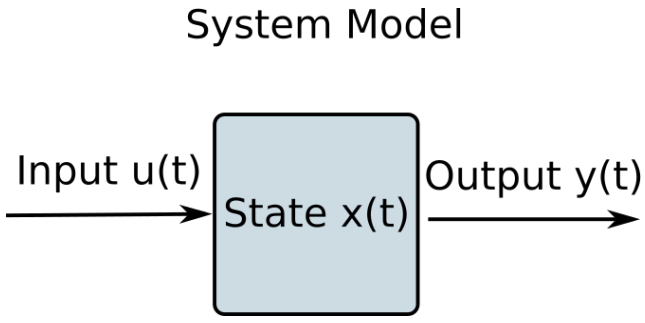


Outline

Concepts


State Estimation
Localization
Mapping
SLAM
Path Planning
Navigation

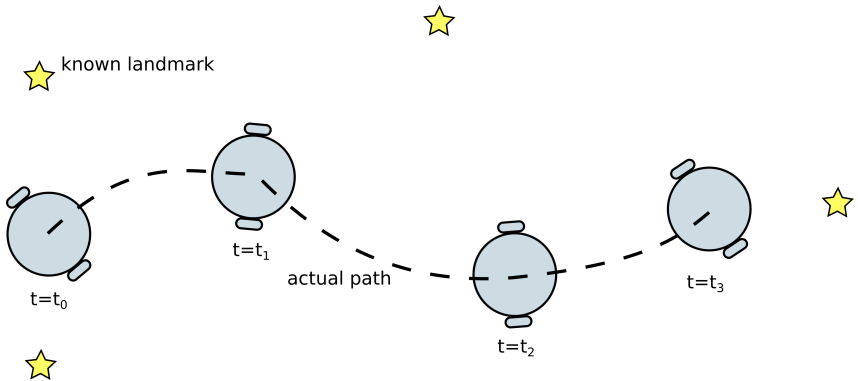
Concepts - State Estimation



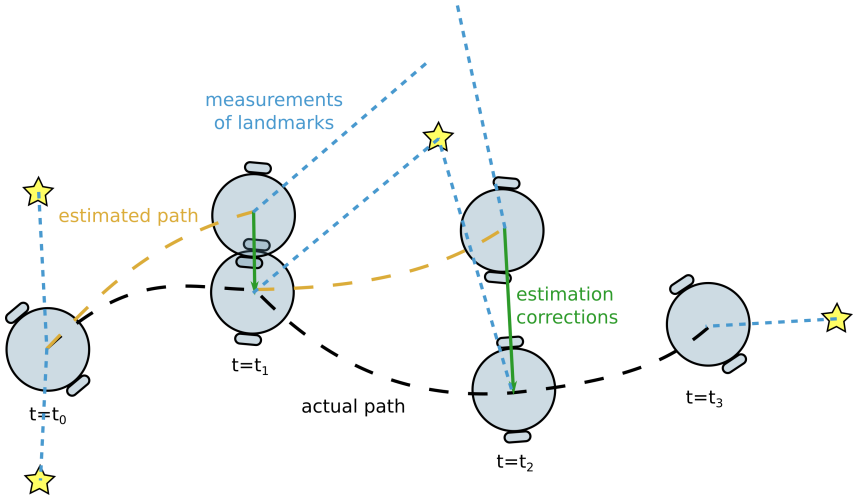
State Estimation

Computation of an estimate $\hat{x}(t)$ of the internal state $x(t)$ of a system from observations of the system's inputs $u(t)$ and outputs $y(t)$.

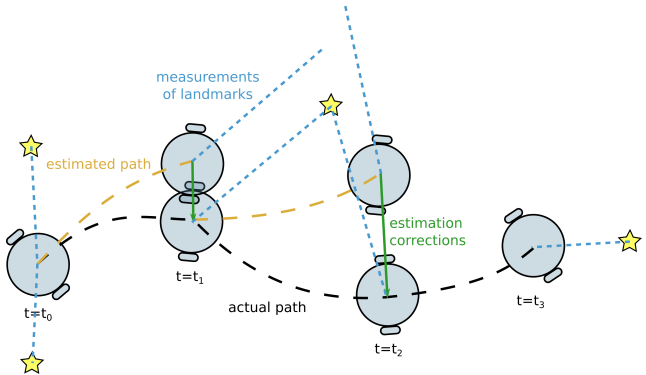
Concepts - Localization



Concepts - Localization



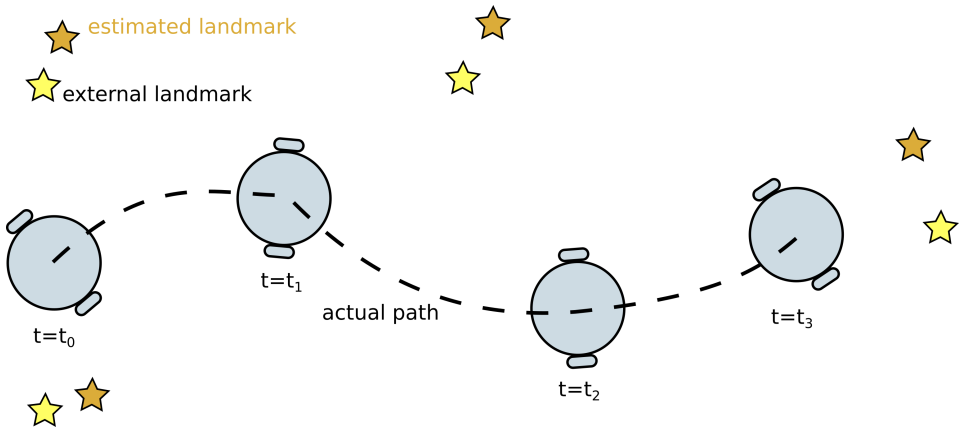
Concepts - Localization



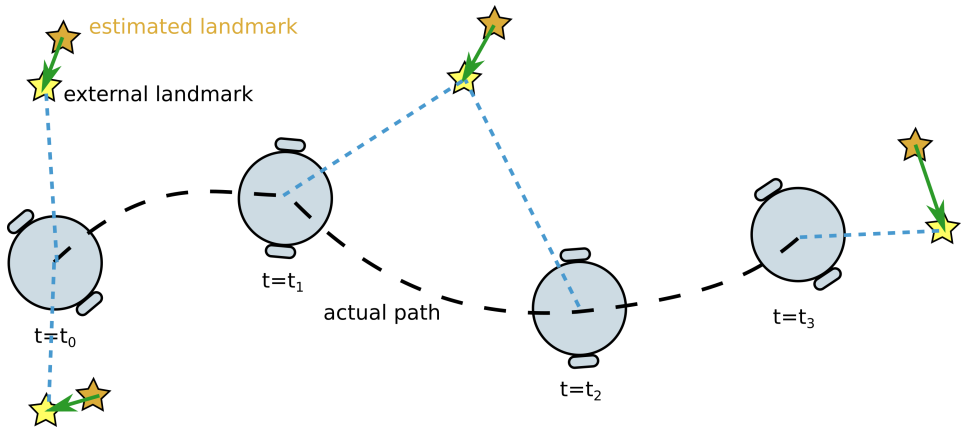
Localization

Estimation of the robot's location in the world, given some known external landmarks.

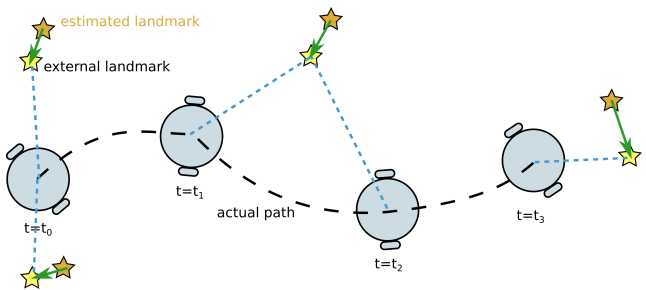
Concepts - Mapping



Concepts - Mapping



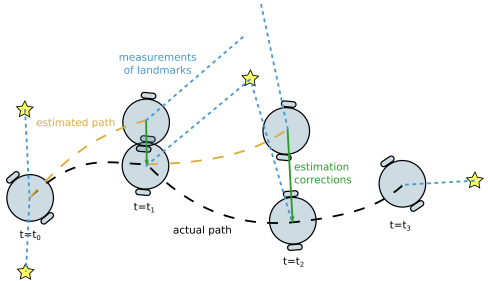
Concepts - Mapping

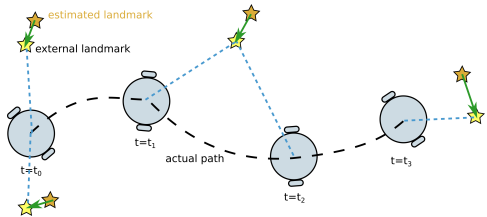


Mapping

Estimation of external landmarks in the world, given the robot's location is known.

Concepts - SLAM

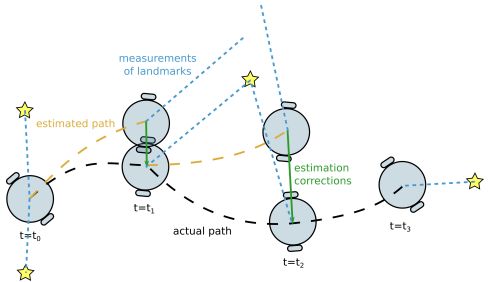


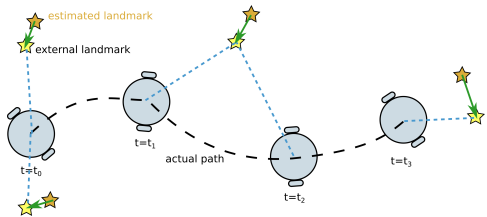


Simultaneous Localization and Mapping

Estimation of the locations of the robot and the external landmarks, at the same time.

Concepts - SLAM





SLAM is a **chicken-or-egg** problem

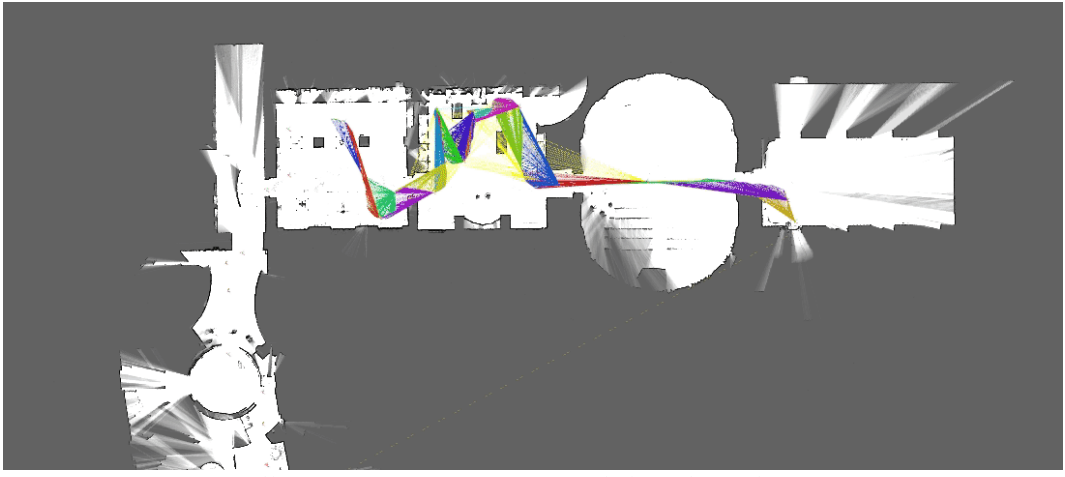
- Known landmarks are needed for localization
- Known robot is needed for mapping

Concepts - SLAM

Problem Definition of SLAM

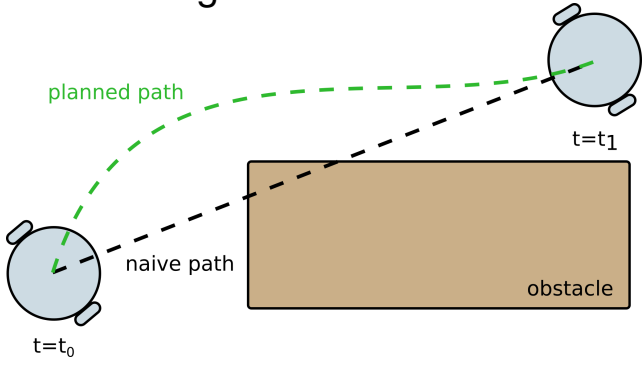
- Given:
  - The robot's control inputs: $u_{1:T} = \{u(1), u(2), \ldots, u(T)\}$
  - The robot's measurements: $y_{1:T} = \{y(1), y(2), \ldots, y(T)\}$
- Wanted:
  - Environment map m
  - The robot's path $x_{1:T} = \{x(1), x(2), \ldots, x(T)\}$

Concepts - SLAM



https://google-cartographer-ros.readthedocs.io/en/latest/_images/demo_2d.gif

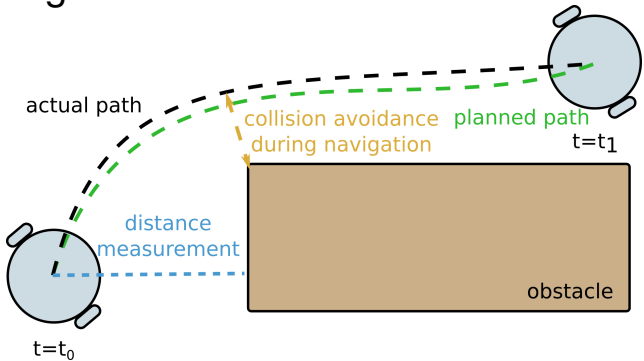
Concepts - Path Planning



Path Planning

Compute a sequence of valid configurations that moves the robot from the source to destination. https://en.wikipedia.org/wiki/Motion_planning

Concepts - Navigation

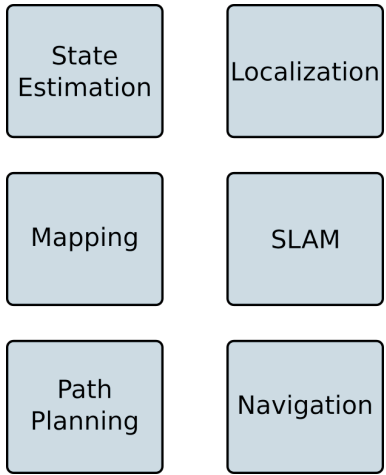


Navigation

Navigation is a field of study that focuses on the process of monitoring and controlling the movement of a craft or vehicle from one place to another.

https://en.wikipedia.org/wiki/Navigation

Concepts


State Estimation
Localization
Mapping
SLAM
Path Planning
Navigation

Outline

Software in ROS

- Mapping
  - Gmapping: `http://wiki.ros.org/gmapping`
- Localization
  - AMCL: `http://wiki.ros.org/amcl`
- SLAM
  - slam toolbox: `http://wiki.ros.org/slam_toolbox`
  - Carthographer: `https://google-cartographer-ros.readthedocs.io`
- Navigation:
  - move_base: `http://wiki.ros.org/move_base`
  - move_base_flex: `http://wiki.ros.org/move_base_flex`

Courses and Literature

- Another brief overview slide deck:
  `https://www.dis.uniroma1.it/~nardi/Didattica/CAI/matdid/robot-programming-ROS-introduction-to-navigation.pdf`
- Very good course on SLAM from Uni Freiburg:
  `http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/`
- Online programming course: `http://www.theconstructsim.com/construct-learn-develop-robots-using-ros/robotigniteacademy_learnros/ros-courses-library/ros-courses-ros-navigation-in-5-days/`

Autonomous Driving

- Waymo: `https://www.youtube.com/watch?v=hA_-MkUONfw`
- Why autonomous driving stalls: `https://www.youtube.com/watch?v=4sCK-a33Nkk`
- Pros and cons: `https://www.youtube.com/watch?v=G2OU_lzsMdE`

Robot Navigation (2019)

Outline

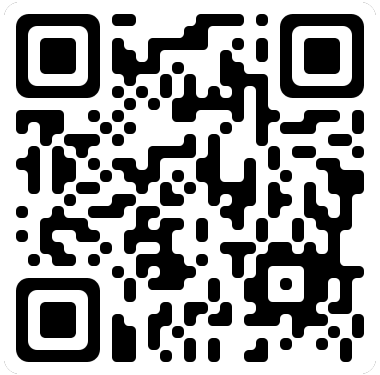
Assignment and dates

- Assignment 6:
  `https://github.com/artnie/rpwr-assignments`
- Grades: 8 points for this assignment
- Due: 06.12., 23:59 AM German time
- Next class: 07.12., 14:00

Evaluation

Thanks for your attention!

Special thanks to the IAI team for the content of this lecture!



https://forms.gle/iZyKqLCxsrwBU3XZ6