München, 2019-04-25

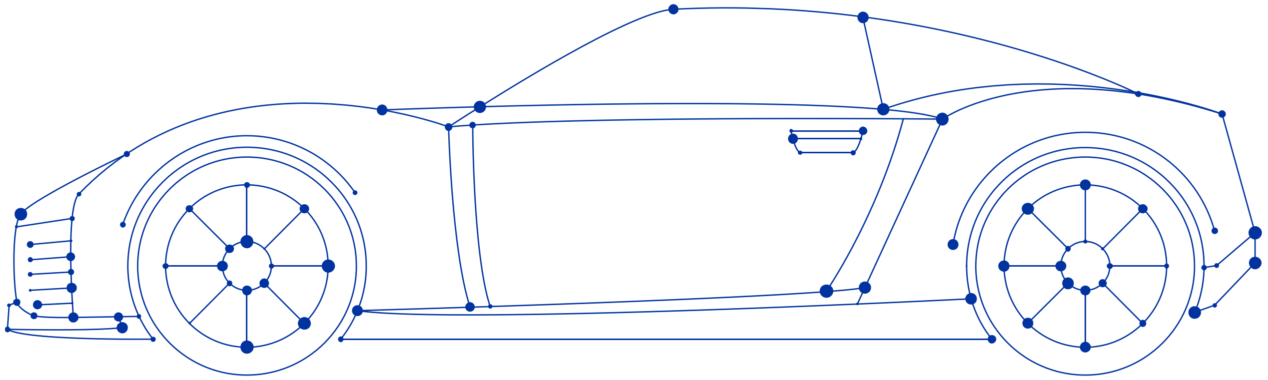
Autonomous Parking

Overview + 1st Sprint

Praktikum, Summer Semester 2019

Hernan Ponce de Leon, Sudeep Kanav, Marco Volpe

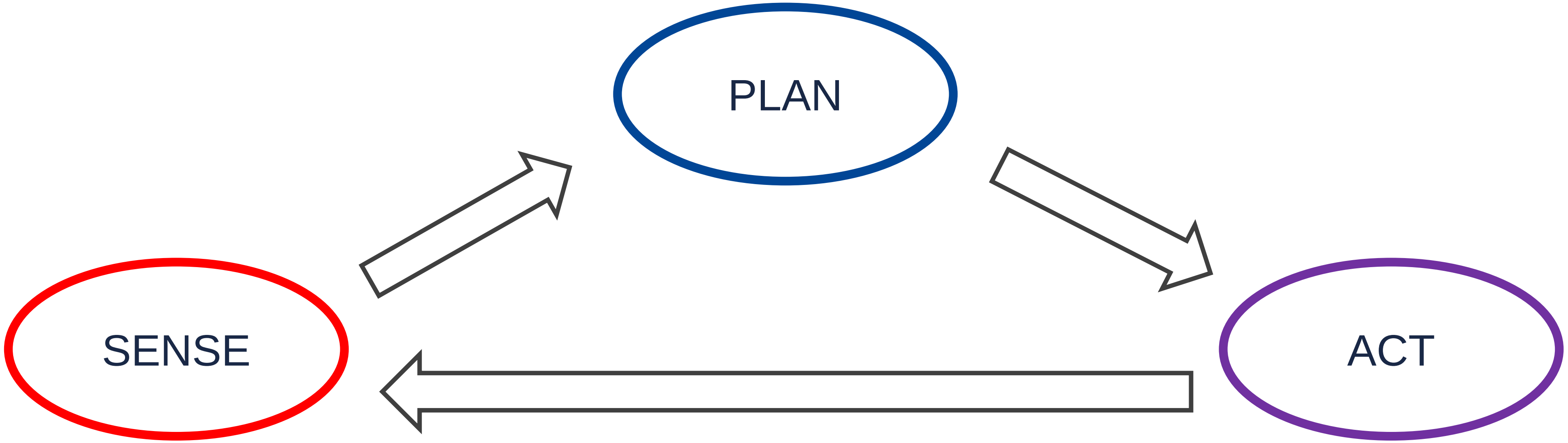
Thomas Böhm, Martin Eisenmann



Overall Goals

- Enable rover’s **autonomous parking**:
    1. **Sense** in the environment:
        - ➢ an appropriate parking location
        - ➢ possible obstacles
    2. **Plan** a path:
        - ➢ reaching the parking location
        - ➢ avoiding possible obstacles
    3. **Act** in the environment:
        - ➢ by moving according to the path planned

Robot control phases

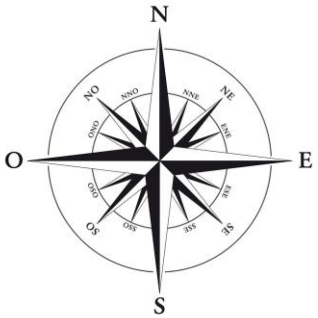


Our Components

environment model

driving functions

SENSE → PLAN → ACT



Sense

Our Toolbox: sensors at our disposal

- Wide angle camera
  - 200 degrees angle
  - Image resolution: 2952x1944
  - Used for reading QR code
  - Used for lane detection
  - More information here
- LIDAR: distance sensor
  - Used for sensing distance of the obstacle straight ahead.
- Ultrasound
  - Used for sensing obstacle distance on the sides
- LIDAR: 360 laser range scanner
  - Obstacle detection
  - We provide a c file which reads the values
  - More info here

Act

Our Toolbox

- Rover
- AF3 model
- Current implementation

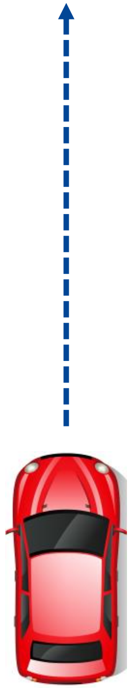
1^{st} Sprint

A working prototype in a basic environment

- Goals:
  - Define interface between *sense* and *plan* (environment model)
  - *Sense*
    - Integrate all sensors
    - Use input datas from cameras and LIDAR to define a simple environment model
  - *Plan*
    - Plan in a simple environment model
    - Communicate with low-level driving functions

Test Scenario 1

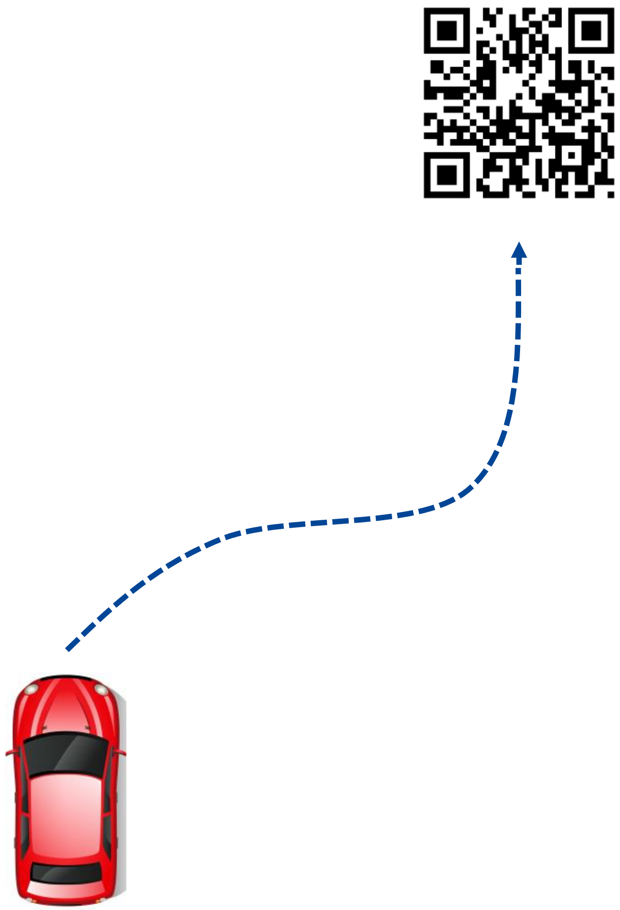
Move straight to the goal location



- Expected tasks
  - Define interface *sense*/*plan*
  - Define the physical scenario
  - Integrate data from camera for QR
    - Improve existing code
    - Support higher resolution for longer distance
  - Detect goal location
  - Create environment model
  - Support basic planning to reach the goal
  - Transmit output data to low-level driving functions

Test Scenario 2

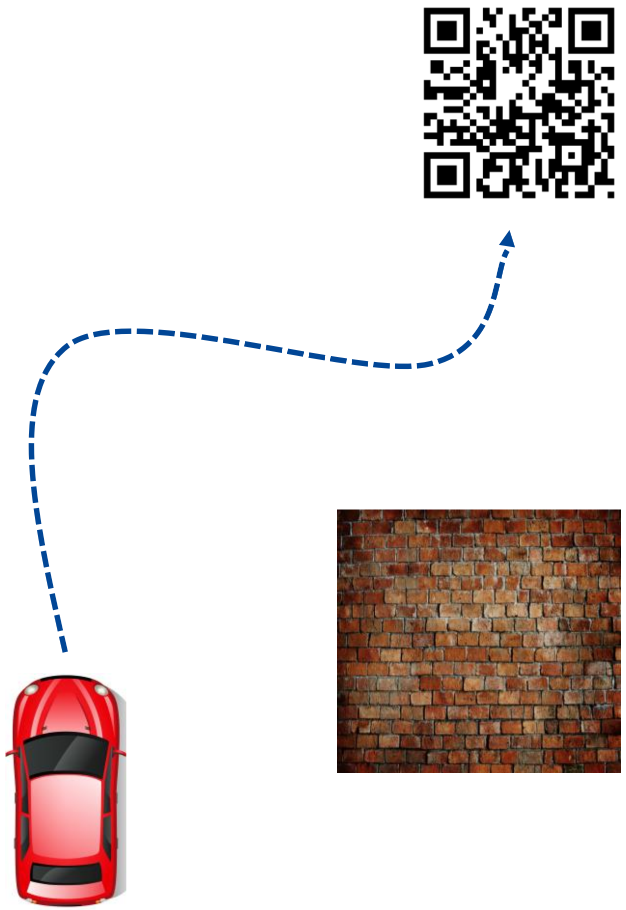
Curve to reach the goal location



- Expected tasks
  - Define the physical scenario
  - Integrate data from camera for QR
    - Angle is also relevant here
  - Detect goal location
    - Even if not in front of the camera
  - Support basic planning to reach the goal
    - Change of direction is also relevant
  - Transmit output data to low-level driving functions
    - Change of direction is also relevant

Test Scenario 3

Avoid (at least) one obstacle



- Expected tasks
  - Define the physical scenario
  - Integrate data from LIDAR
  - Detect obstacles
  - Support planning to reach the goal
    - By avoiding obstacles

Open questions

- How do we model the environment?
  - Will we need a more advanced model for localization in next sprints?
- Do we need a kinematic model of the rover in this sprint?
- Which algorithms to use for path planning?
  - Need for replanning?
- Which programming languages/tools shall we use?