TAS – Autonomous Driving and parking roboter

Christopher Zeiser, David Full, Hans Böhme

TAS - Final Presentation/Report

Prof. Dr. Wollherr, Khoi Hoang Dinh

Lehrstuhl für Steuerungs- und Regelungstechnik

Technische Universität München

Table of Contents

Motivation

Interesting facts about autonomous driving:

- 2025 - Law and insurances are ready
- 2035 - 20-35% autonomous cars
- Accident reduction by 15% thanks to car assistant systems in the past 10 years in Europe

Task 1 – Map Saving

1. Record map with hector mapping
   → ‘run1.launch‘

2. Save map with „map_saver“ service
   → ‘run2.launch‘ and ‘run2.sh‘

3. Start necessary nodes
   → ‘run3.launch‘

→ Flexible starting position

Task 1 - Parameter Optimization

Preset parameters for each car

- Forward and backward speed
- Steering angle offset

Parameter server for real time driving optimization

- Additional speed (forward and backward)
- Additional steering angle offset

Local/Global planner parameters

Task 1 – Single-Goal Autonomous Driving

- Reducing the number of waypoints to one
- Independent of map shape and size, only low requirements close to the robot
- Possible to start at nearly every position

→ High flexibility

Task 1 – Single-Goal Autonomous Driving

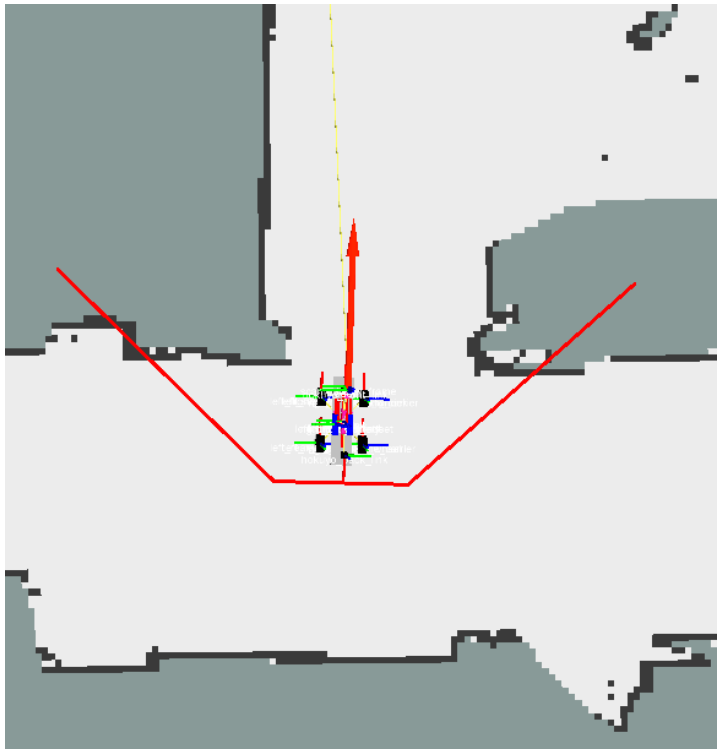
Idea:

Change the map temporarily by faked laser data.

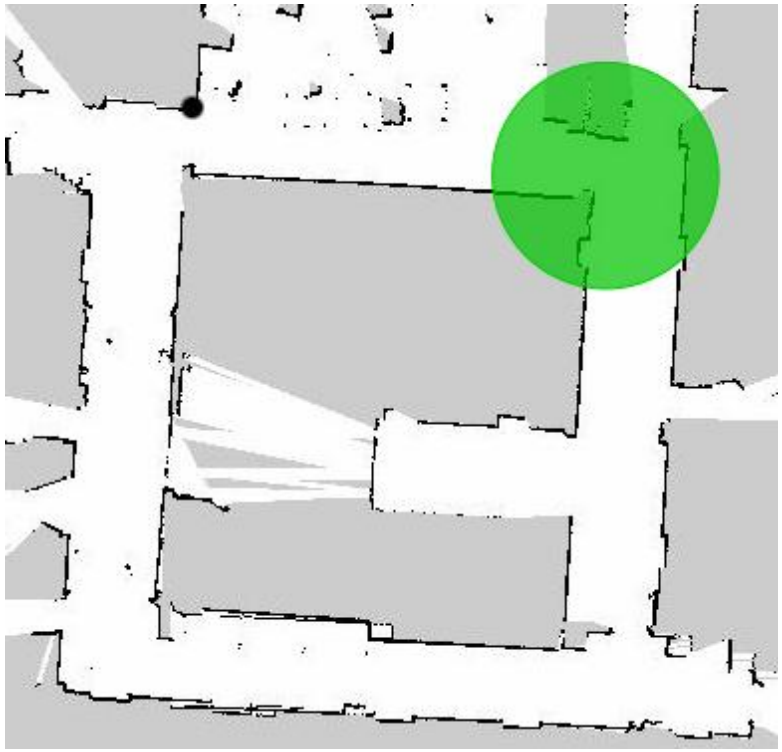
Approach:

1. Wait until wall is published
2. Set goal directly behind the car
3. Start autonomous driving
4. Remove the wall and slightly correct the goal to the starting position

Task 1 – Single-Goal Autonomous Driving



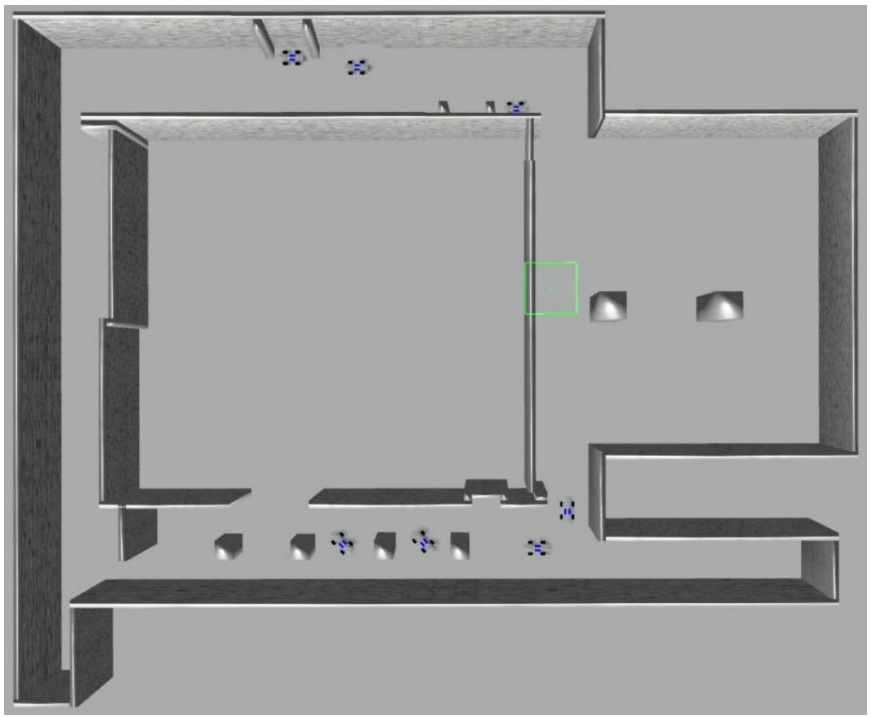
U-shaped wall behind the car



Relevant area in green

Task 2 - Simulation Extension

- Additional parking slot
- Additional slalom course
- 8 different starting positions accessable on simulation startup



- *roslaunch tas_simulator startSimulation.launch* ***pos‘x‘:=1***

Task 2 – Flexible Parking

New parking node for laser (position) + imu (orientation) communication.

Task implemented with final state maschine like approach

1. Start park procedure with pressing the A-button
2. Detect parking slot and drive to starting position
3. 5 binary coded state indicators + 7 states + formulas depending on car indicators, states and parking attempt (left – right, in - out)

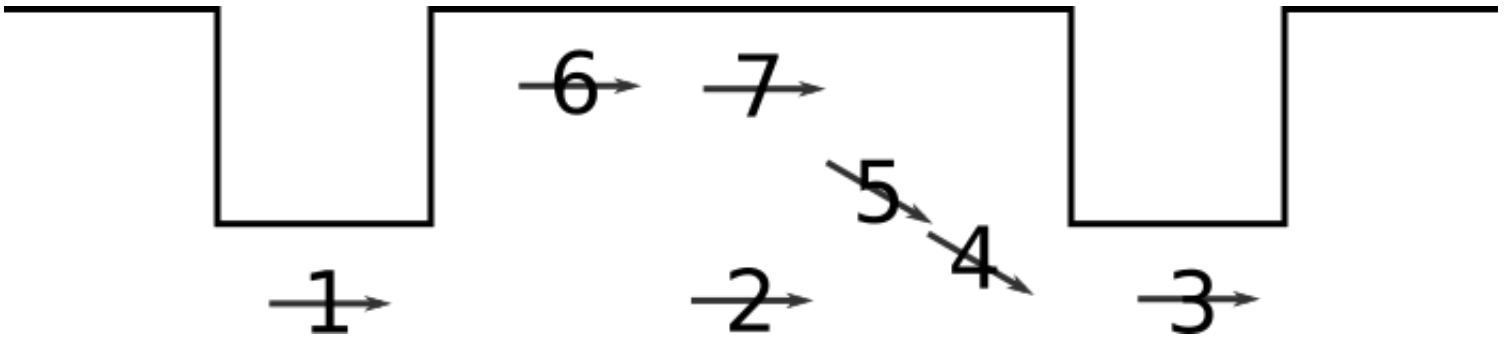
Task 2 – Flexible Parking

LSB

- 0/1 current_orientation == start_orientation
- 0/1 current_roentation == start_orientation $\pm$ 45°
- 0/1 car in parking slot
- 0/1 car near the wall in the parking slot
- 0/1 front_scan == back_scan

MSB

➔ Truth tables for velocity and steering angle



Summary

- Stable autonomous driving + obstacle avoidance
- Parking with flexible side and (un-)parking and partially independence of starting position
- Optimized simulation
- Optimized parameters for all treated application

Problems and feedback

- Hardware issues (LAN, bluetooth, screen)
- Software issues (malfunctioning simulation)
- Car issues (battery, steering parameters)

Thank you for your attention!