Technical Autonomous Systems

Group 3

Fernando Espindola / Hao Xu

Final presentation

Supervisor: Khoi Hoang Dinh / Zengjie Zhang

Chair of Automatic Control Engineering

Technical University of Munich

Overview

- Task 1: Autonomous driving on a pre-defined path
  - Parameter calibration
  - Setting waypoints
- Task 2: Autonomous slalom (with trailer)
  - Simulation environment
  - Slalom driving method 1
  - Slalom driving method 2
  - Slalom with trailer
- Summary

Autonomous driving on pre-defined path

Contribution

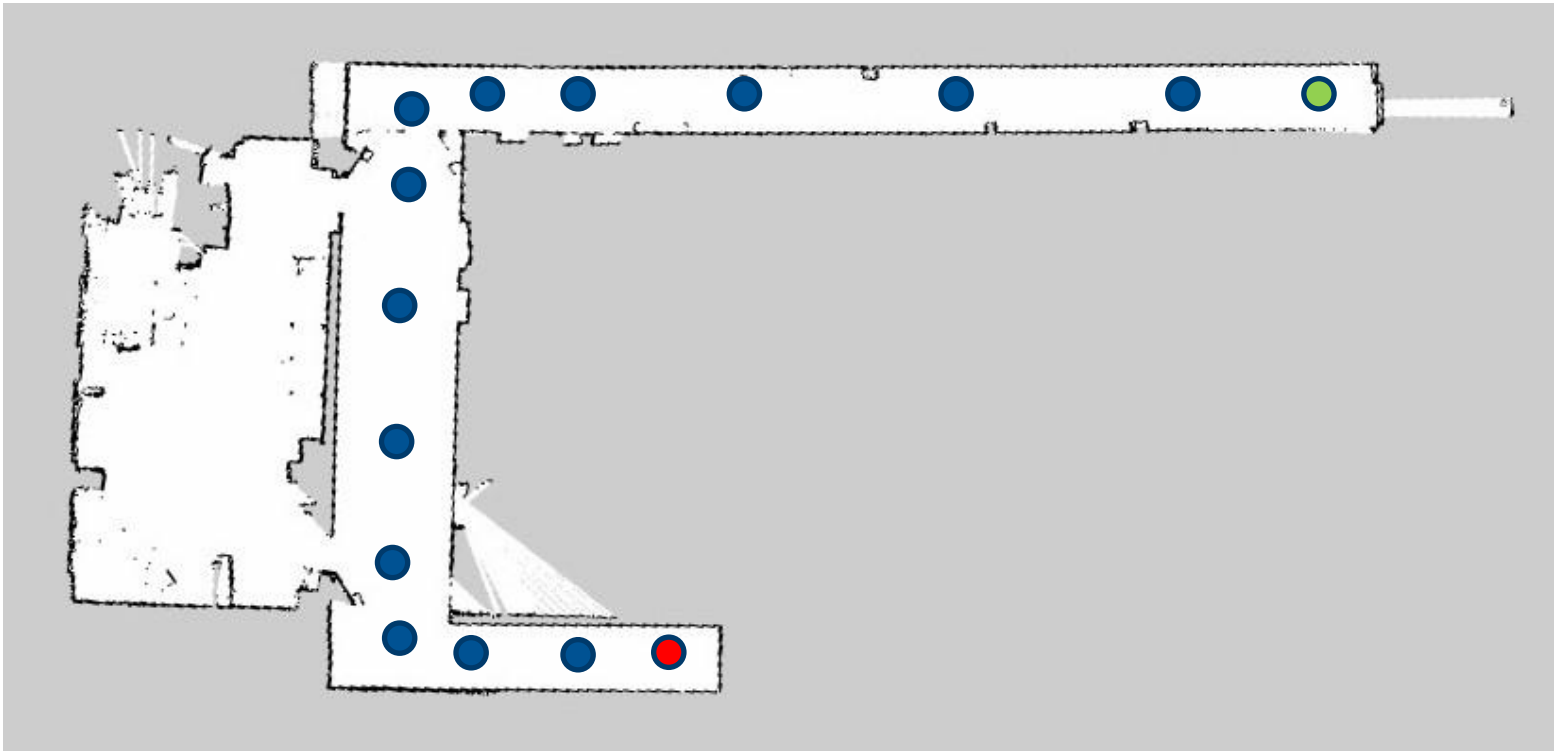
- Preset **new navigation goals** as waypoints
- **Calibrate parameters** in control files

Parameter calibration

- Move_base parameters:
  - obstacle_range decreased
  - inflation_radius decreased
  - heading_lookahead decreased
- Car dependent parameters
  - cmd_steeringAngle = 1500 ± 500/30*cmd_steeringAngle;
  - autonomous_control.control_servo.x = 1580;

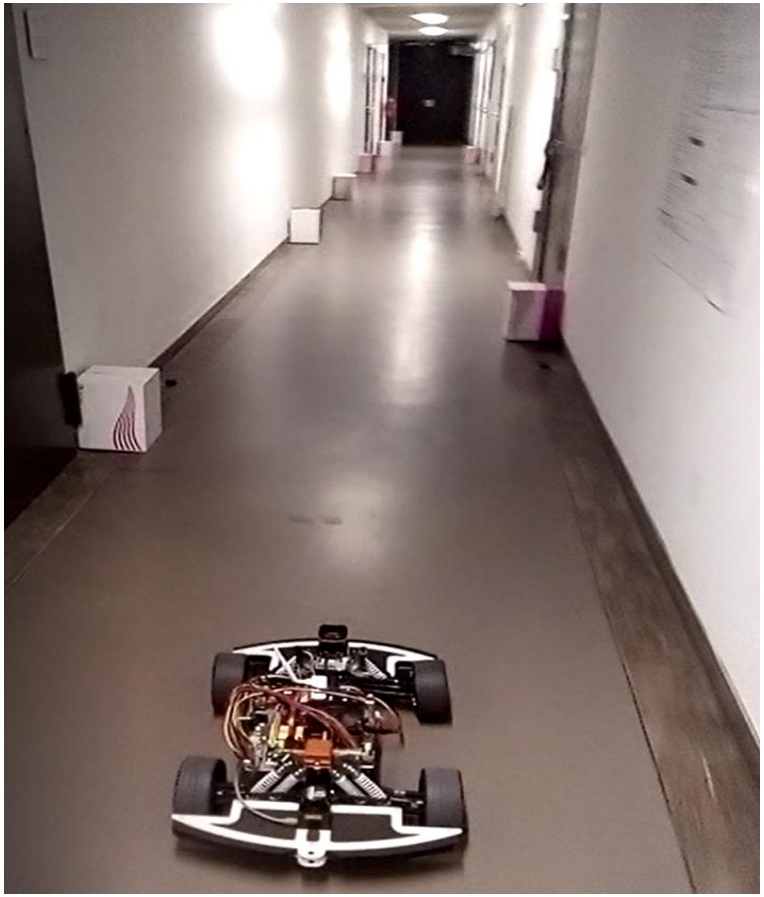
Setting waypoints

- Waypoints set on existing map



Final result

- Fullfill time: 61s
- 18 preset waypoints



Autonomous slalom with trailer

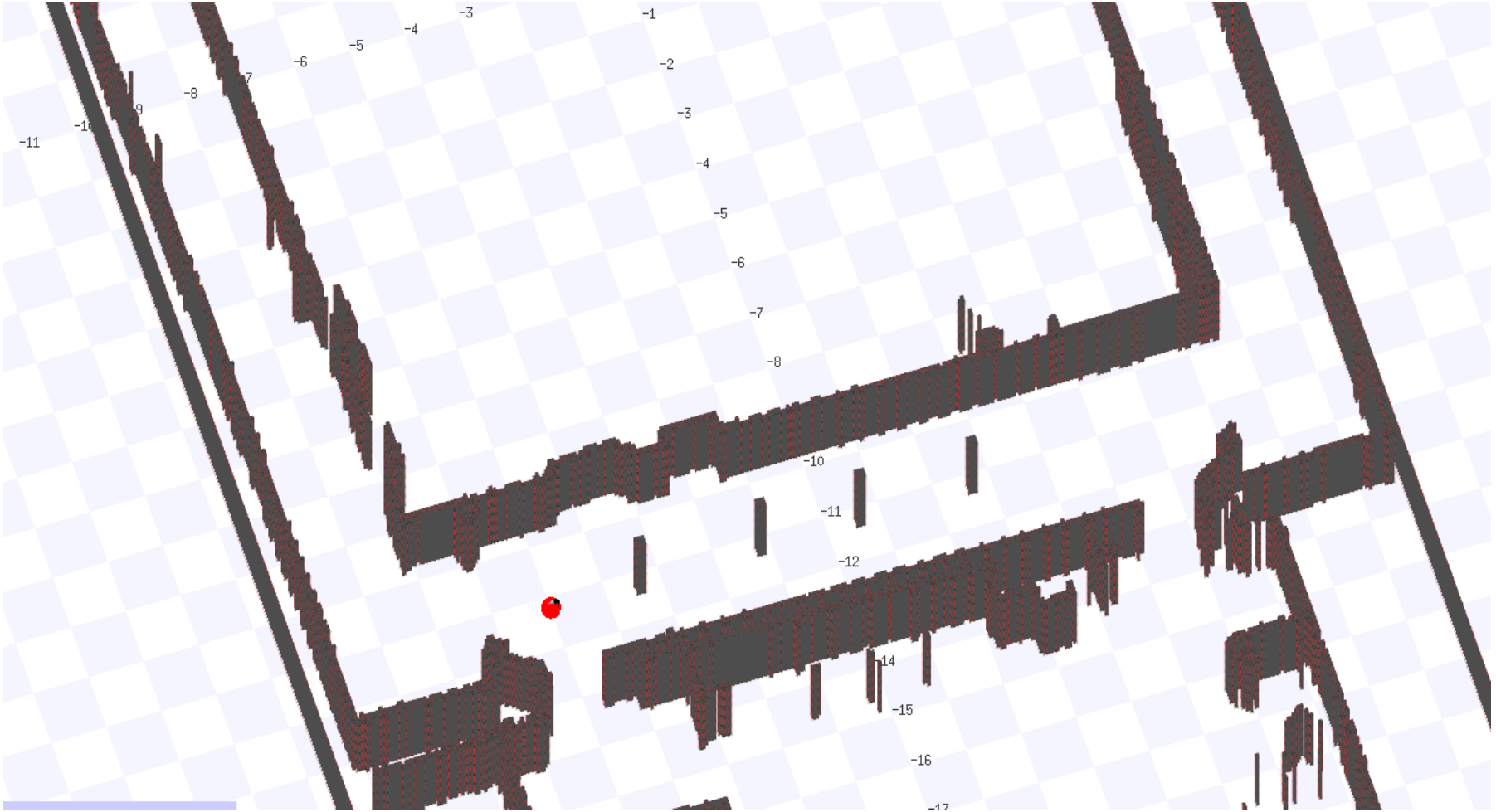
Contribution

- Build **simulation** environment
- Implement **two methods** for **slalom driving**
- Accomplish **slalom** driving **with trailer**

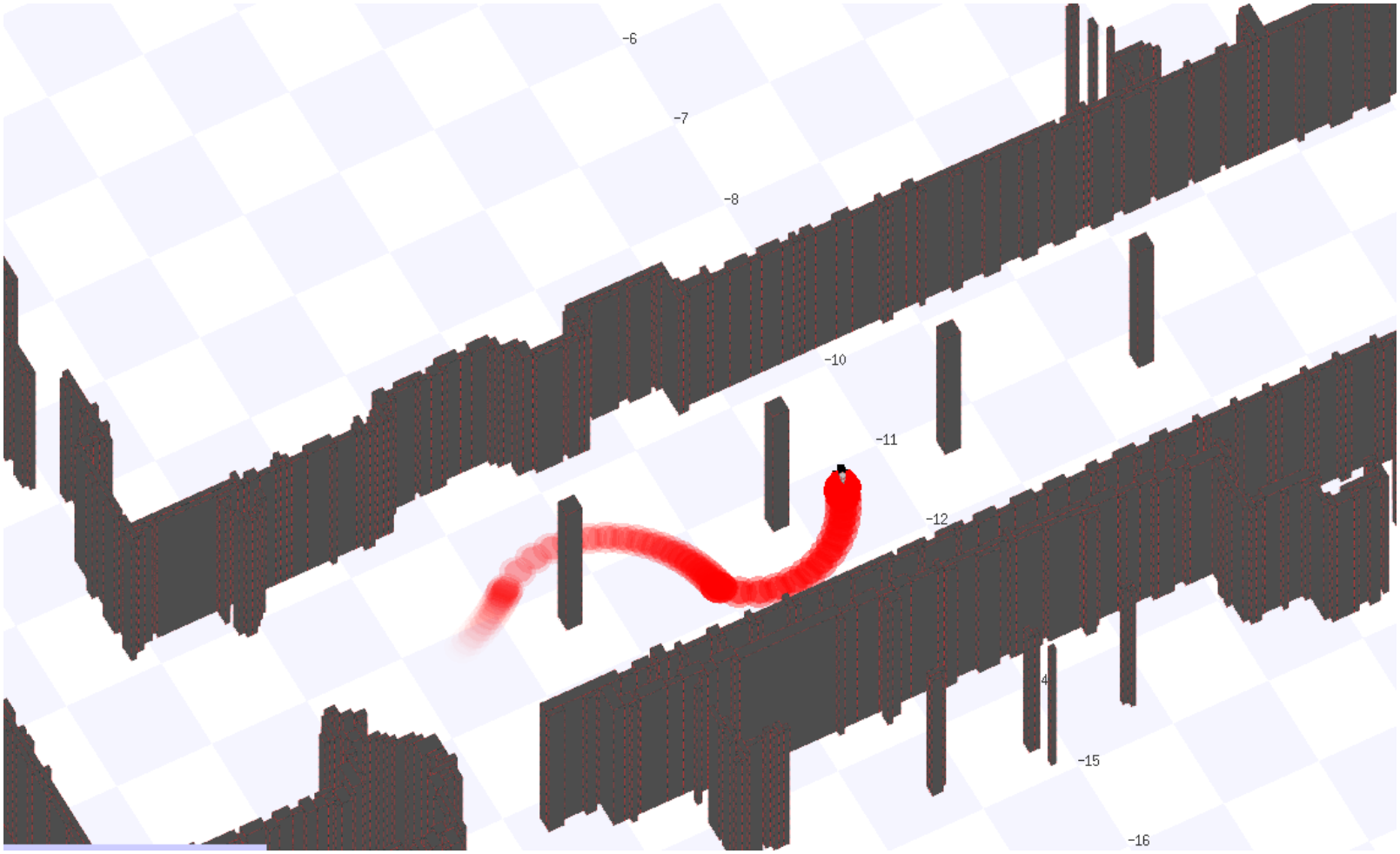
Build simulation environment

- Simulation in **Stage**
- **Custom map** based on .PNG image
- **LaserScan** parameters
- Works for **SLAM**

Build simulation environment



Build simulation environment

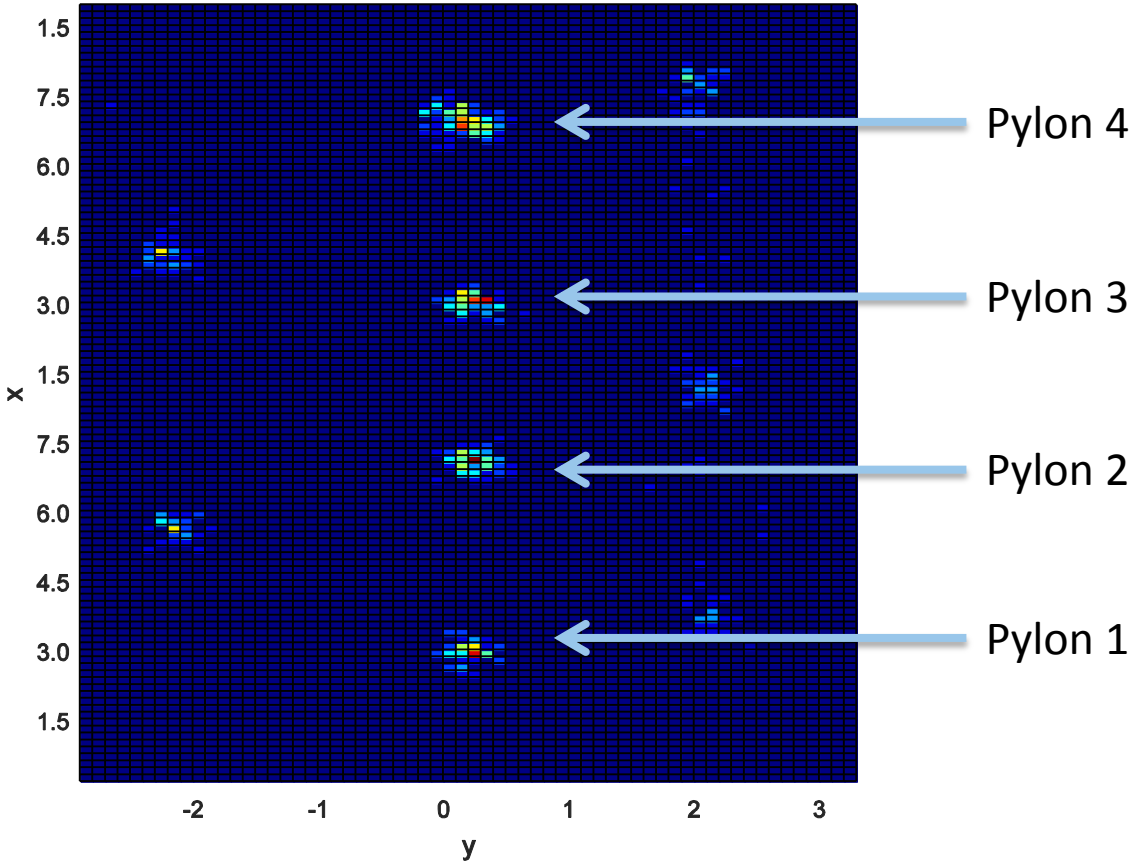


Slalom driving method 1

- **Look for** 1st pylon
- **Transform** point to frame /odom
- **Project** point **4 times** in x-direction
- **Broadcast tfs** for every pylon
- Define **relative goals**

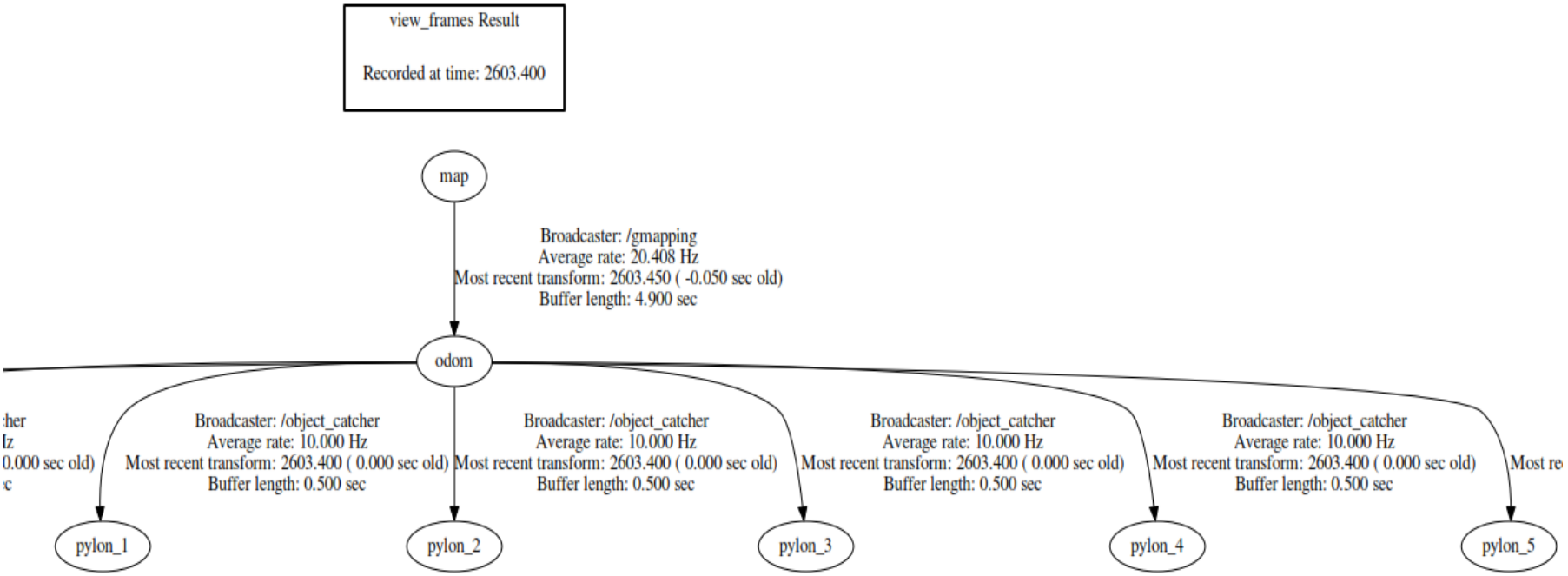
Slalom driving method 1

- Look for pylons and **build a heatmap**



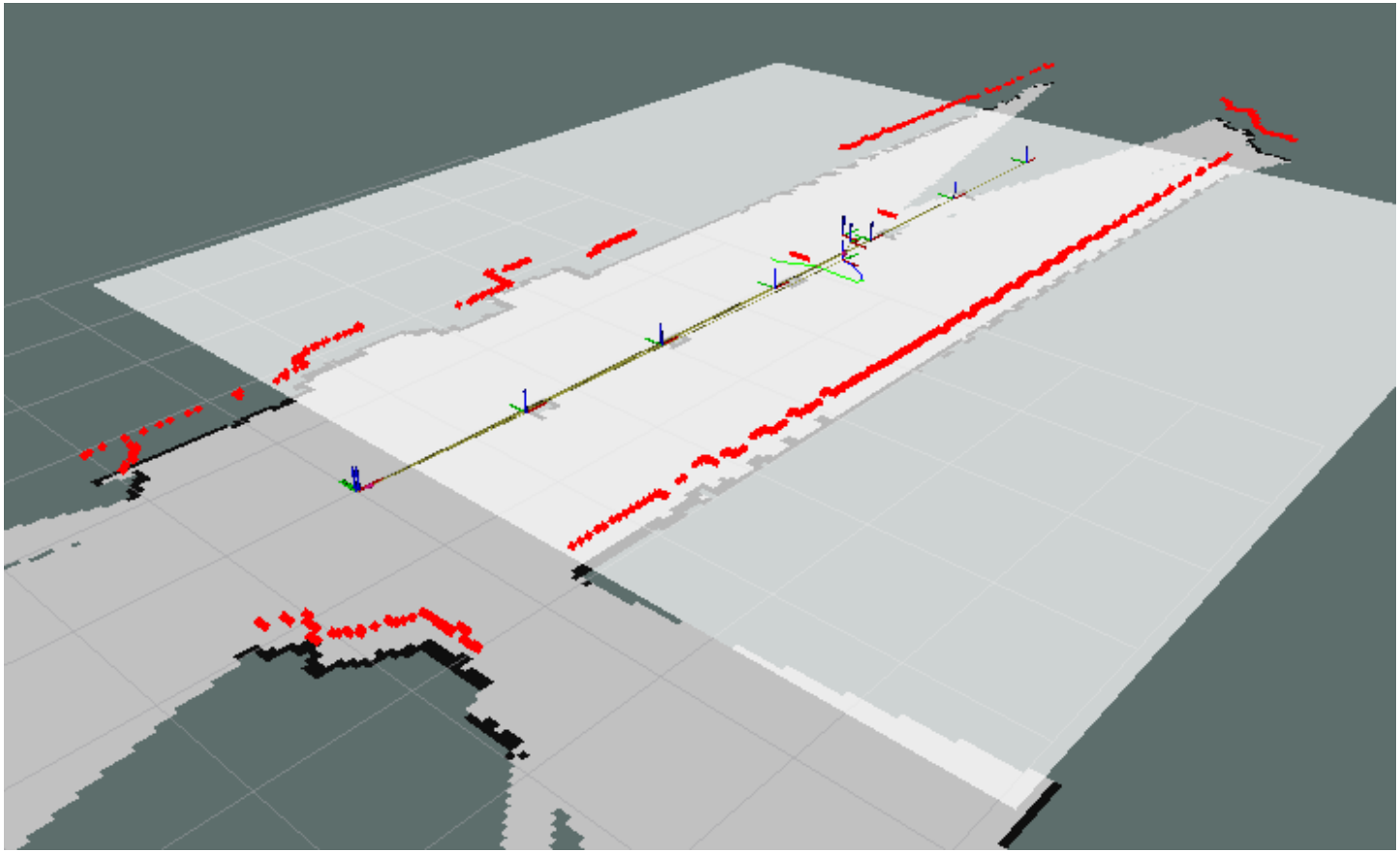
Slalom driving method 1

- **Broadcast transforms** for pylon_0 to pylon_5



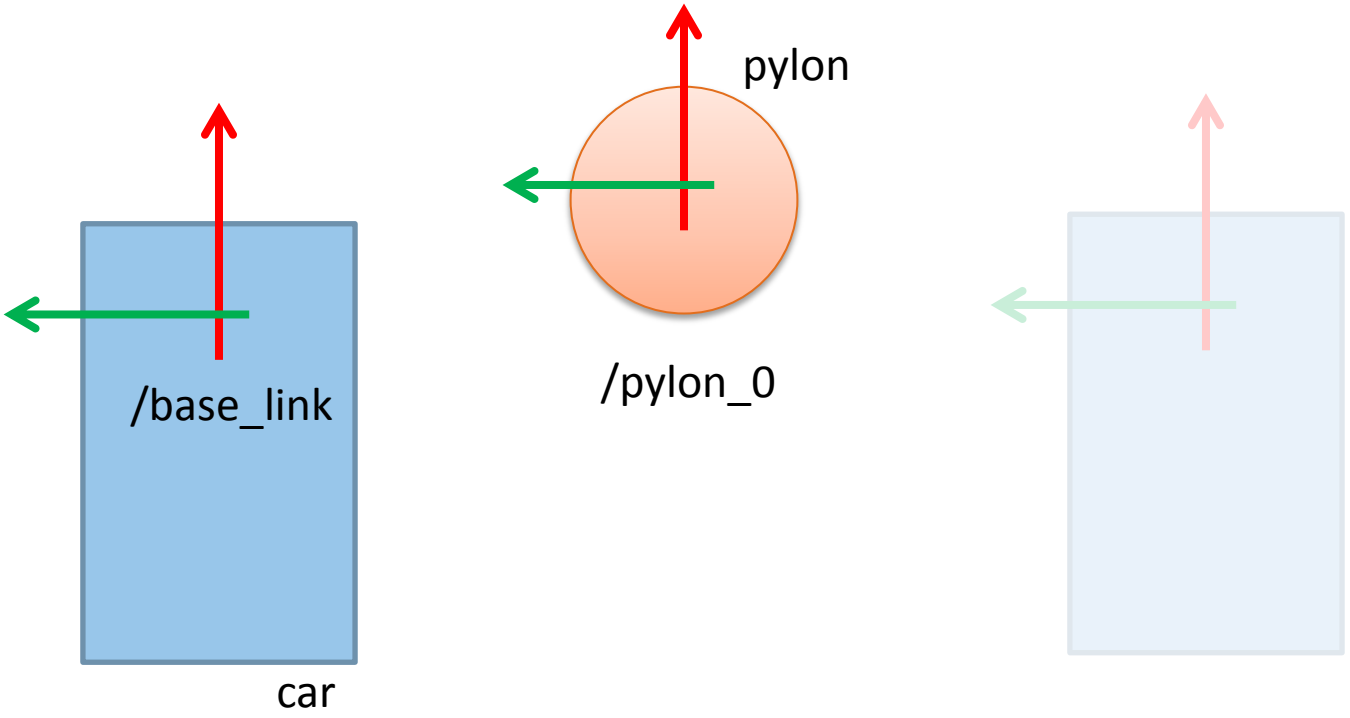
Slalom driving method 1

- **Project** point 4 times in **x-direction**



Slalom driving method 1

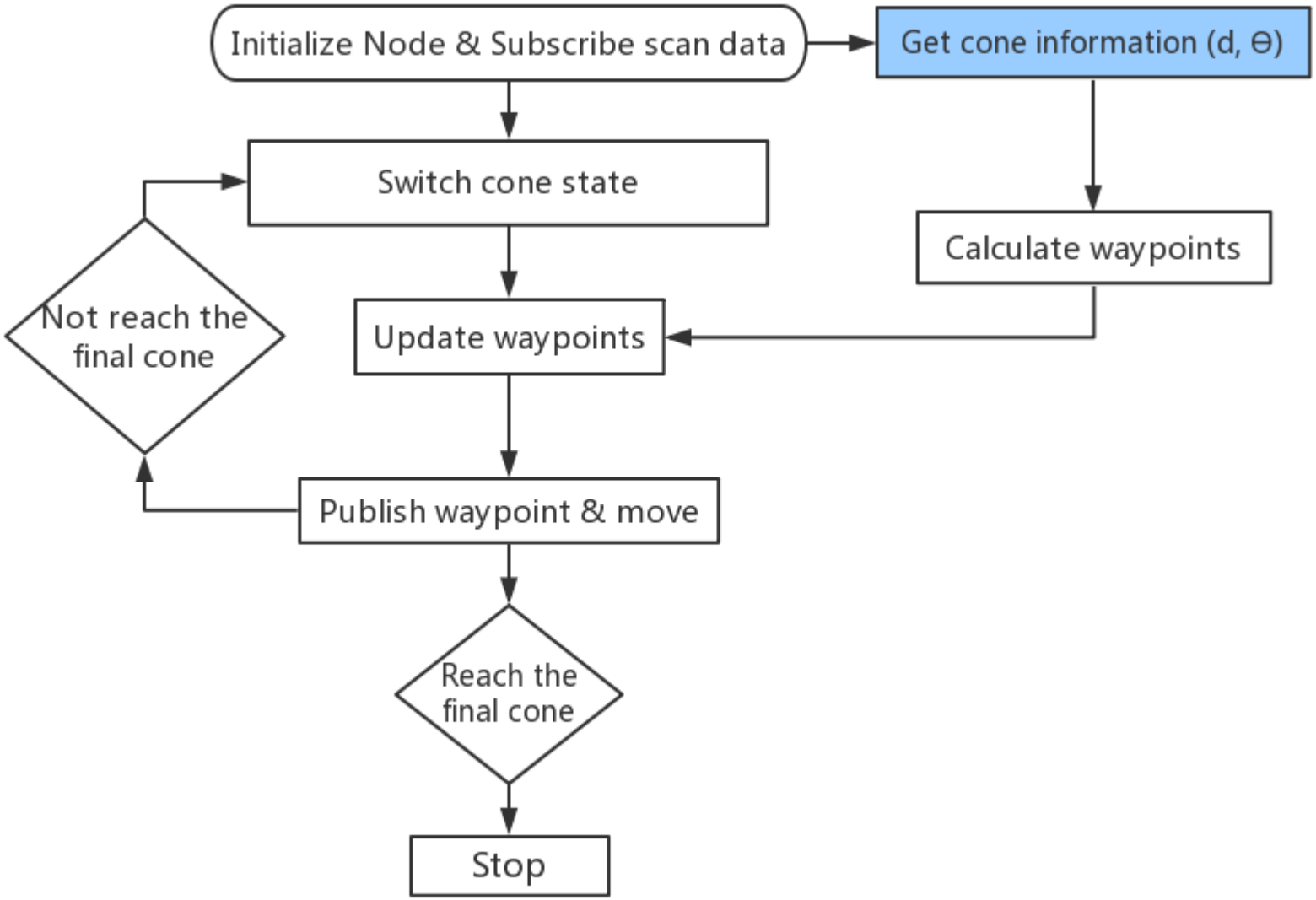
- Definition of **relative goals**
- X, Y, and Yaw



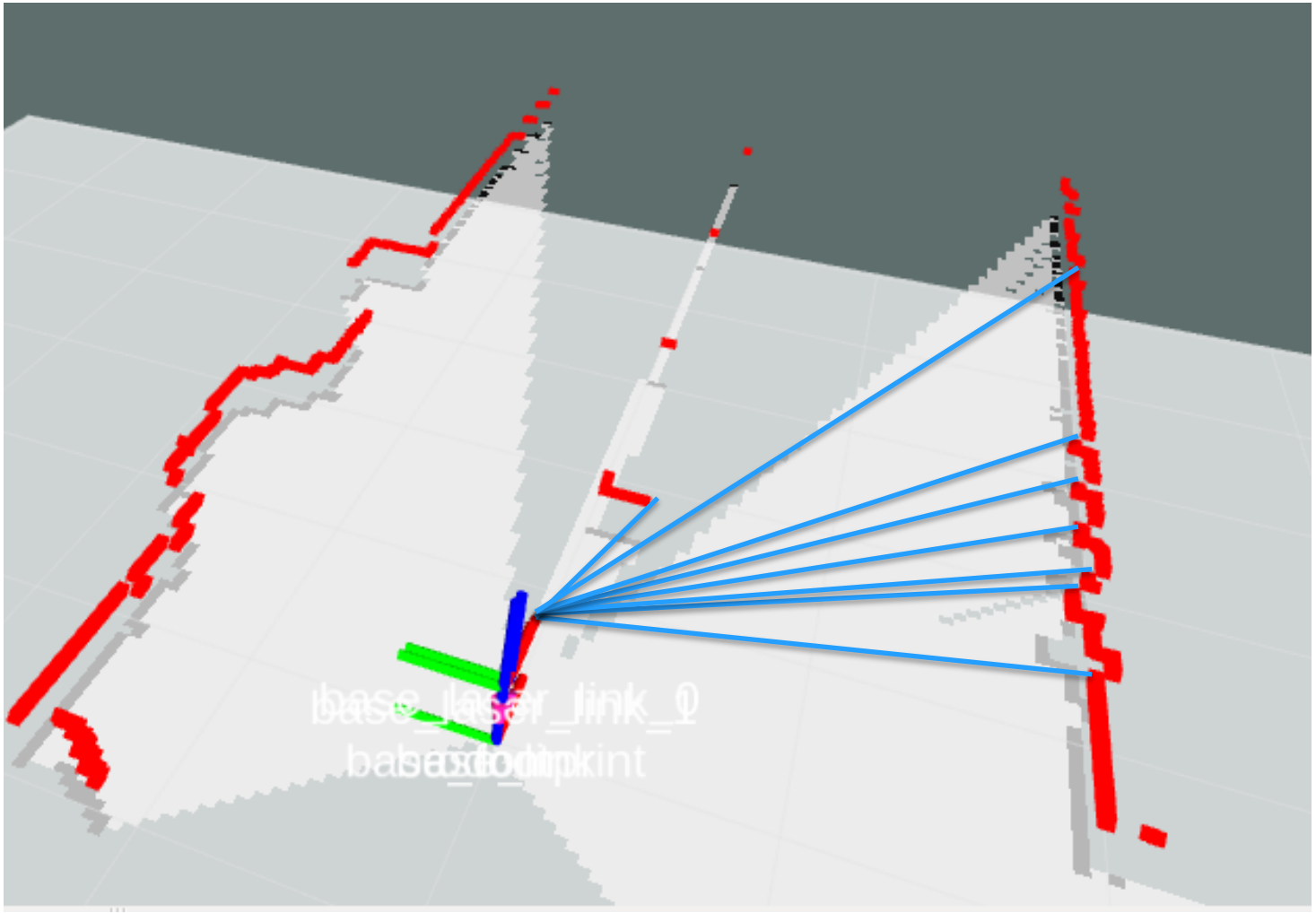
Slalom driving method 2

- Detect different cone according to distance
- Replace pre-defined cone model with the detected mode
- Calculate waypoints using distance and angle parameter

Slalom algorithm

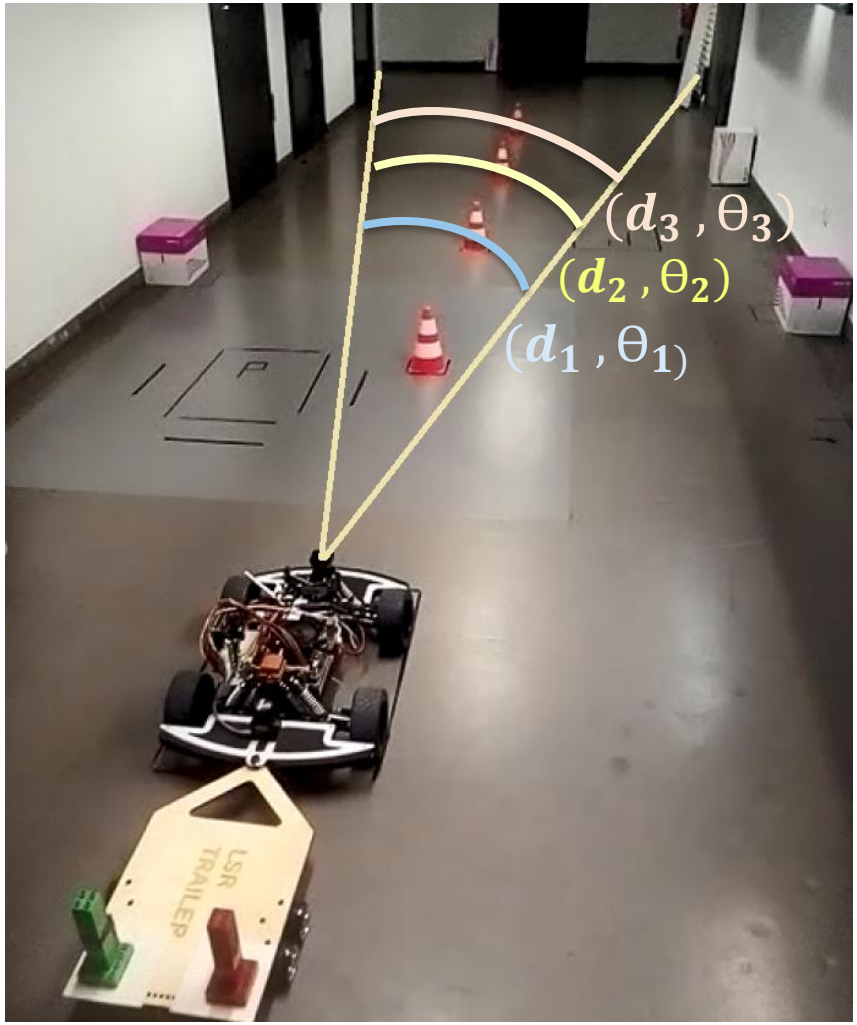


Cone detection using cone frame



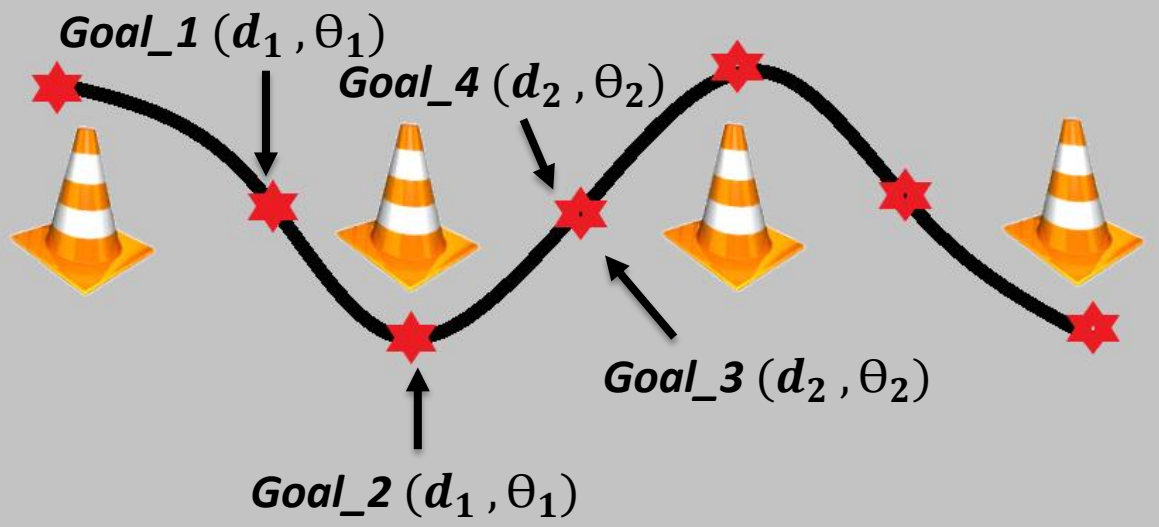
Cone detection using distance and view angle

- Scan once at initial state
- Set view angle range
- Get distance and angle information
- Calculate waypoints



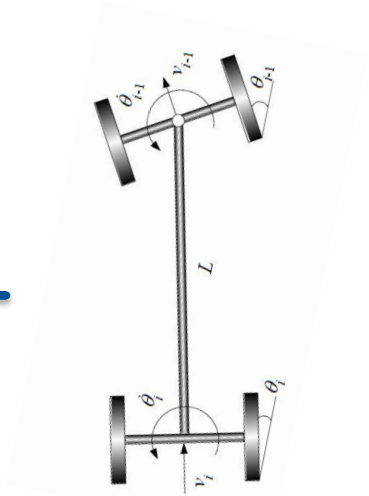
Adding more waypoints

For *Goal_1:*



- More stable
- Local planner provides driving angle.

Slalom with trailer



Direct-hooked trailer

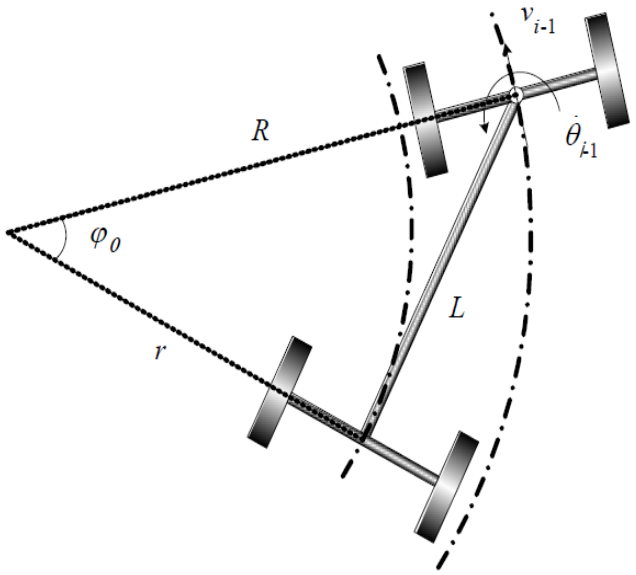
Off-hooked trailer

【Lee et al., IJCA04】

Tracking error in trailer system

$$\varepsilon = R - r = R - \sqrt{R^2 - L^2}$$

- R: trajectory radius of the front trailer
- r: trajectory radius of the back trailer
- L: lengths of the link



【Lee et al., IJCA04】

Tracking error in trailer system

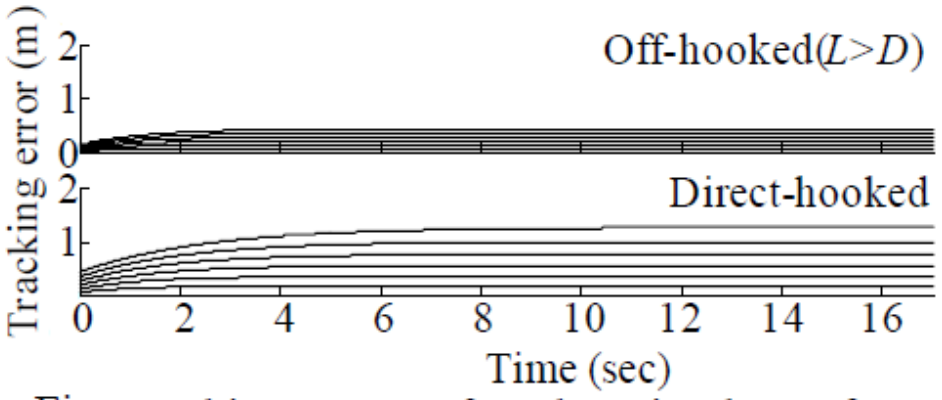
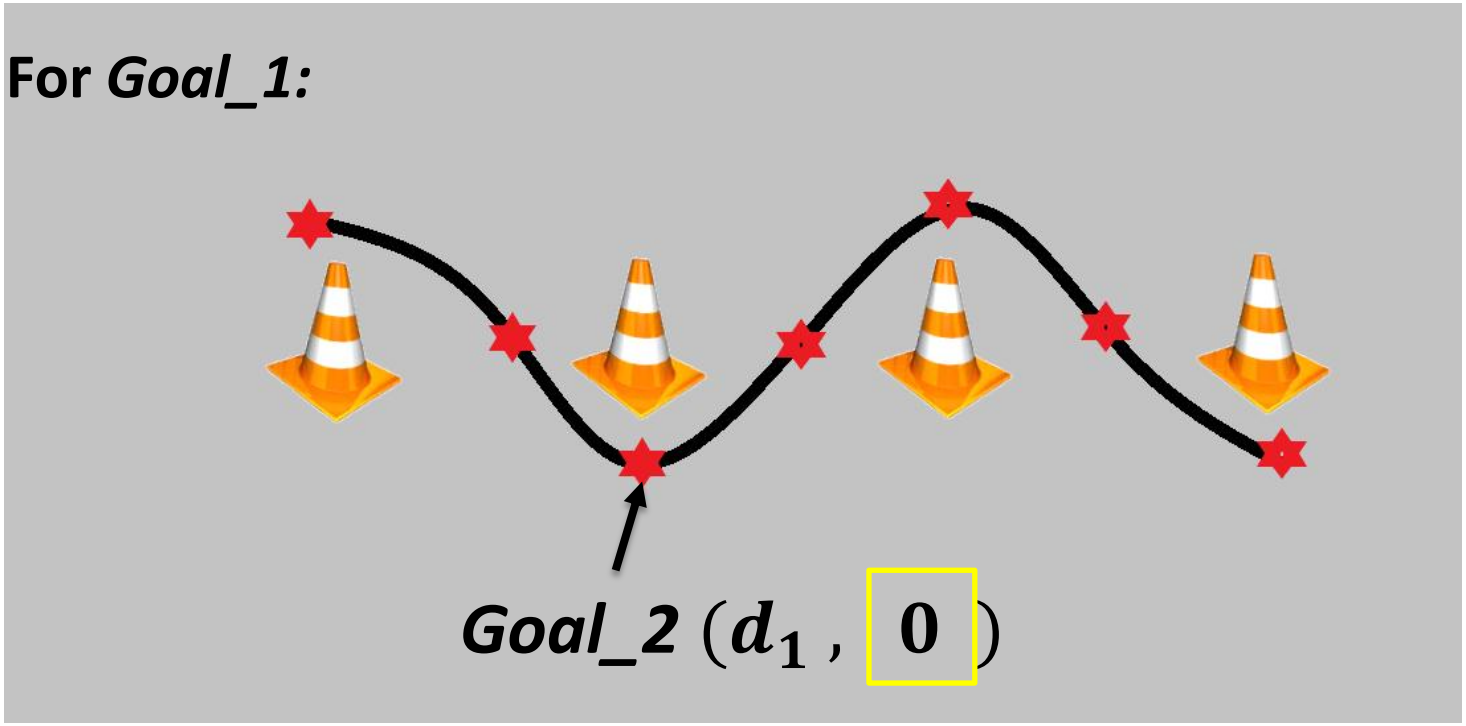


Fig. Tracking errors for the circular reference trajectory.

【Lee et al., IJCA04】

Off-hooked trailer has smaller tracking errors than the direct-hooked trailer

Tracking error in trailer system



Set angle of waypoints beside cone equals 0:

- approximate the radius of curvature of the reference trajectory to infinity.

Summary

- Autonomous driving on a pre-defined path.
- Build simulation environment in Stage.
- Implement slalom (with trailer) using two methods.

References

- LEE, Jae-Hyoung, et al. A passive multiple trailer system with off-axle hitching. *International Journal of Control, Automation, and Systems*, 2004, 2. Jg., Nr. 3, S. 289-297.

- ZHENG, Kaiyu. ROS Navigation Tuning Guide. *arXiv preprint arXiv:1706.09068*, 2017.