h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

# Status Report

**Project Path Planning Team1**

**Report Period:**

**CW 44**

## Milestone:

| Milestone | Date | Status | |
|---|---|---|---|
| Run first OMPL based Motion Planning Algorithm in CoppeliaSim + Python | 08.11.2023 | 🟥 | |
| Adapt OMPL Functions for car model dimensions | | 🟥 | |
| Path Planning Runtime Optimization | | 🟥 | |
| | | ☺🟨🟥 | |

## Results in the report period

| Results (achieved, not achieved, planned) | |
|---|---|
| Results achieved | - Run CoppeliaSim + API on MacOS (use v4.5.1 or older) |
| Results not achieved | - |
| Planned results fort he next period | - **Documentation of Issues and Solvings**<br>- **Use Open Motion Planning Library (OMPL) Functions in Python**<br>- (Model sensors in CoppeliaSim → when Sensor Hardware decision was made)<br>- Model the car in CoppeliaSim → when Car Hardware decision was made<br>- Check existing Motion Planning Algorithms on YouTube → Roman, Lam<br>- **Implementation of Path Tracking (following) Algorithm:**<br>**- Motion Task „ROTATING“ → tbd**<br>**- Motion Task „MOTION_STOP“ → tbd**<br>**- Motion Task „MOTION_PATH_TO_GOAL“ → tbd** |
| | |

| **Problems, Risks, Measures in Report Period** |
|---|
| a) Which problems have been occured? |
| - **OMPL Library is C++, Python binding needed (Windows & MacOS)** |
| b) Which (new) risks can lead to problems? |
| - **Inefficiency of Motion Planning algorithms (runtime)**<br>- **Simulation Sensor Gap (virtual sensors “too perfect” compared to real sensors)** |
| c) So far undertaken countermeasures? Who? Until when? |
| - OMPL Python Binding MacOS → Lam |

- OMPL Python Binding Windows → Laurens

d) Necessary decisions to take? By whom? Until when?

●
●