Autonomous Vehicles and Artificial Intelligence SoSe 2026

Assignment 1: Robotics Operation System (ROS)

Hand-In via Moodle: May 14, 12:00

Task 1: "Kill-Switch" Application

To manually stop the race car in dangerous situations, a separate "Kill-Switch" application must be developed to send an emergency stop signal to the race car. At the start of the race car, the "Kill-Switch" application must establish a connection to the race car. Only when the "Kill-Switch" application is connected to the race car, the race car is allowed to drive autonomously. If the connection between the "Kill-Switch" application and the race car is interrupted, the race car should threaten this as it would have received the emergency stop signal. The current status of the race car should be shown in the "Kill-Switch" application.

The race car already provides the required action ($/simple_driving$) and subscribes to the topic $/control_driving$. Implement the following as part of the "Kill-Switch" application.

1. The application connects to the action server $/simple_driving$, which will enable the driving functionality of the race car, and displays the feedback messages on the current status of the race car (i.e., stopped or driving).
2. A publisher that sends messages to the topic $/control_driving$ that start or stop the race car.

For a detailed documentation of the implemented action server, topic, and message formats please refer to https://github.com/isselab/avai-autonomous-race-car.

Deliverables

Upload a document with your results in Moodle before the deadline.

- Submit all created artifacts in a single ZIP file, that contains the following:
  - A folder for the sources of your project—the artifacts should be documented properly
  - A folder containing the executable binaries.
  - A README that

    contains the author names with student IDs ("Matrikelnummern")

    explains how to execute your implementation