Eclipse ADORe (Automated Driving Open Research)

Daniel Heß (Daniel.Hess@DLR.de)

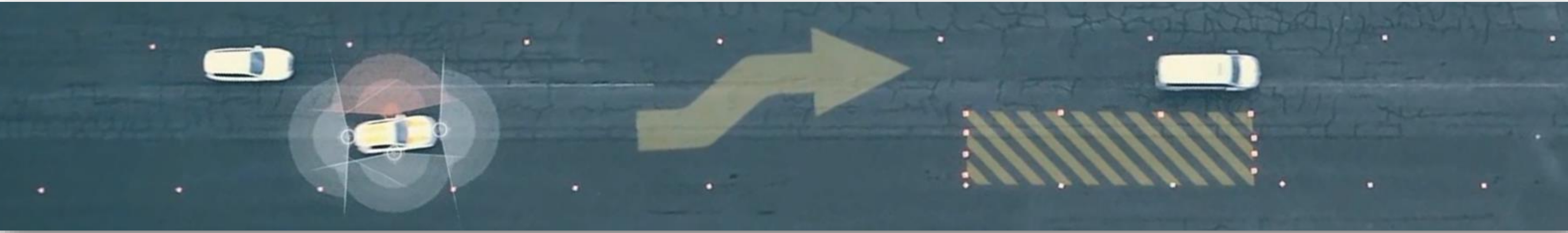
https://projects.eclipse.org/projects/technology.adore

https://github.com/eclipse/adore

DLR

Agenda

- Motivation and Goals
- Overview/Approach
- Version 0.1
- Future Work



Motivation and Goals

DLR-TS team: Designed several specific, automation based mobility solutions:

Cooperative maneuvering, platooning, emergency maneuvers, interactions at intersections, shared control, infrastructure interactions like „Green-light optimal speed advisory“ or „lane change adivce“, etc.

➔ Basis for **Eclipse ADORe: Planning & control for Cooperative Automated Vehicles (CAV)**

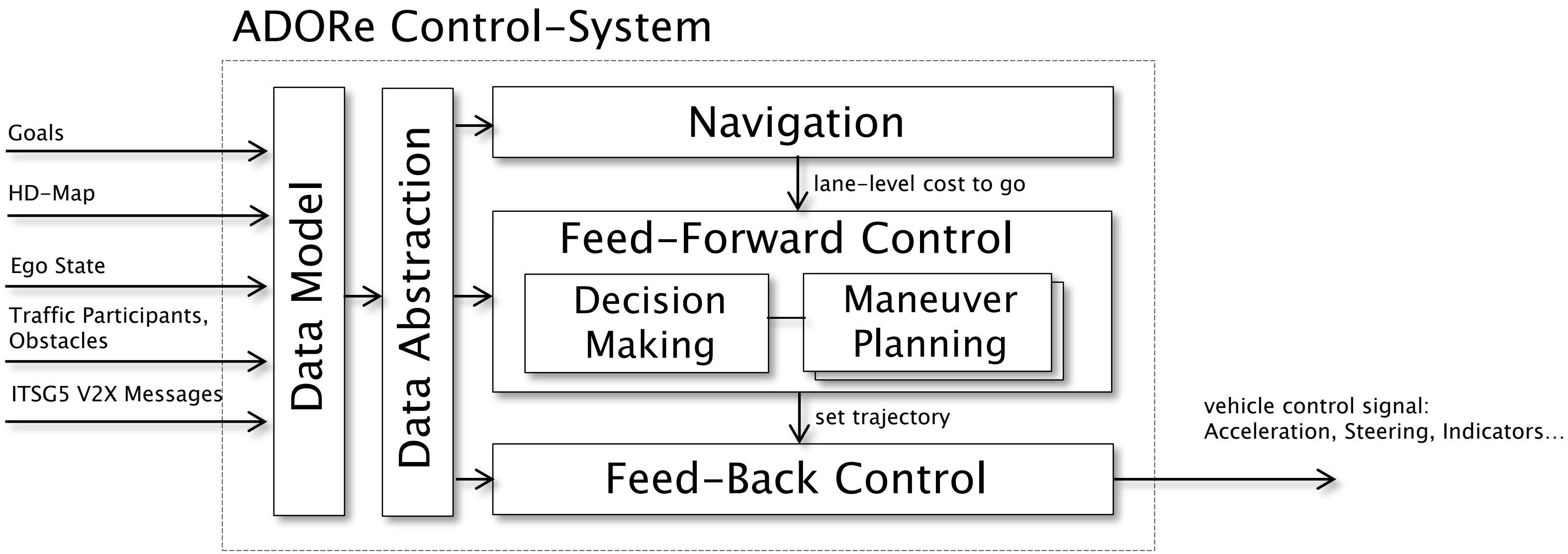
Why Open Source?

- Work with a community – DLR internal and worldwide
- Sustainable platform for research (reproduce results, compare methods)
- Integrate into open tool landscape (middle-ware, simulation, visualization, validation)

Goal of ADORe: Research impact of *specific* CAVs on traffic (SUMO: „Impact of *generalized/idealized* CAVs on traffic“)

- How to design a Cooperative Automated Vehicle?
- How to ensure safety of CAV in mixed traffic?
- How to interact with other CAVs and intelligent roadside infrastructure?
- How to interact with manual vehicles, VRUs?

Overview / Approach



- Modular, system-independent c++ library „libadore“
- Exemplary realization based on ROS 1 (melodic, noetic) „adore_if_ros“

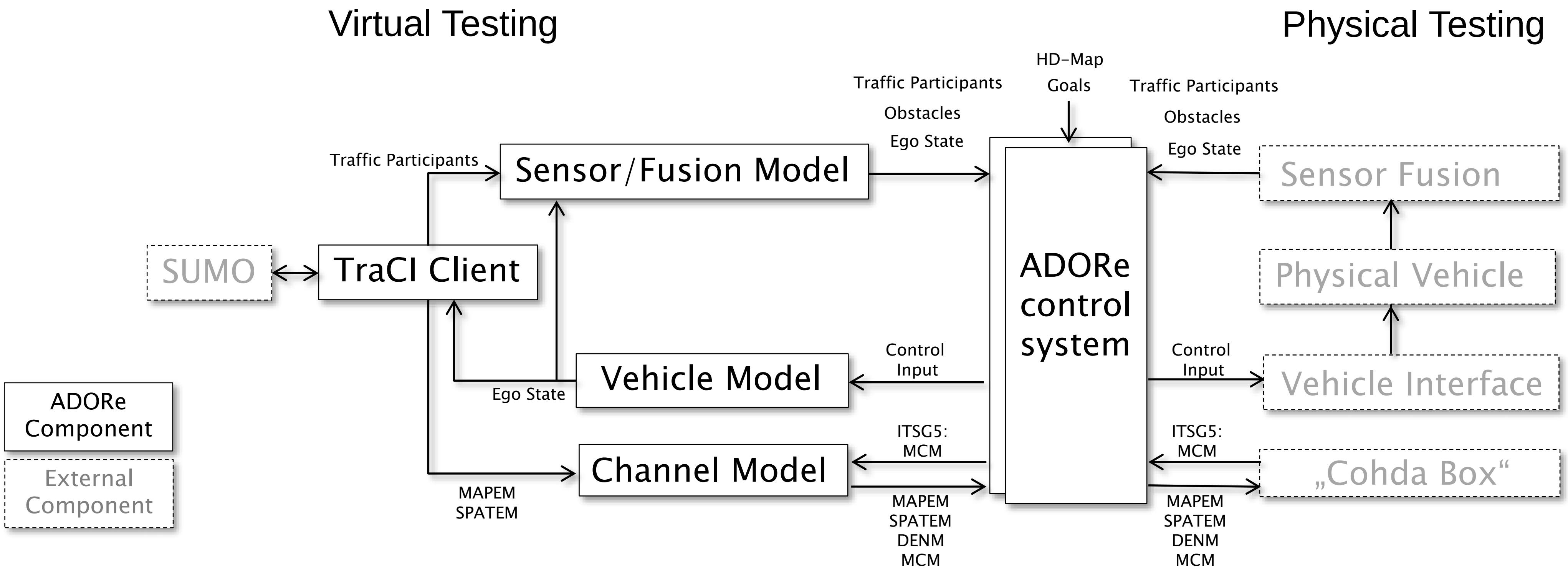
Overview / Approach

- Decision Making:
  - Parametrizing specialized maneuver planners and switching between optimization results
  - Lane-Following, multiple lane-change maneuvers, emergency maneuvers
  - Each maneuver planner solves a local optimization problem
- Maneuver Planning:
  - Non-linear road-coordinate system
  - Decoupled trajectory planning: 1. Longitudinal motion profile, 2. lateral motion profile
  - Linear-quadratic minimization of acceleration and jerk with qpOASES
  - Input-Ouput linearization reconstructs full state and input of nonlinear bicycle model
    - Precise feed-forward control signals
    - Evaluation of state- and control-space constraints
- Feed-back Control:
  - Trajectory tracking
  - PID + feed-forward control signals

→ Further details: Heß et al. "Fast maneuver planning for cooperative automated vehicles." *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018.

DLR

Overview/Approach



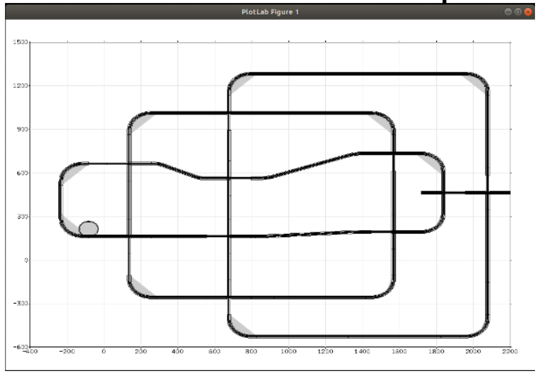
- „Real-time“ simulations
- Headless, slower (or faster) than „real-time“ simulations
- VIL / HIL / SIL testing with mixed virtual / physical traffic components
- Multiple ADORe vehicles

Version 0.1 (2020-06)

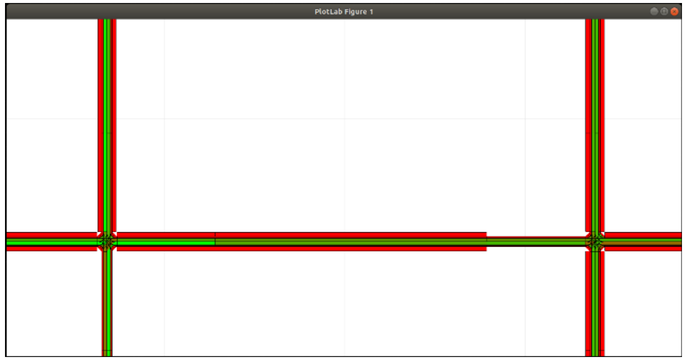
- ROS Kinetic, Ubuntu 18.04
- Vehicle model
- Vehicle control
- Simple sensor fusion model
- qpOASES-based planner
- Open-Drive loader
- Navigation
- Lane following
- SUMO co-simulation

https://github.com/eclipse/adore/tree/master/adore_if_ros_demos

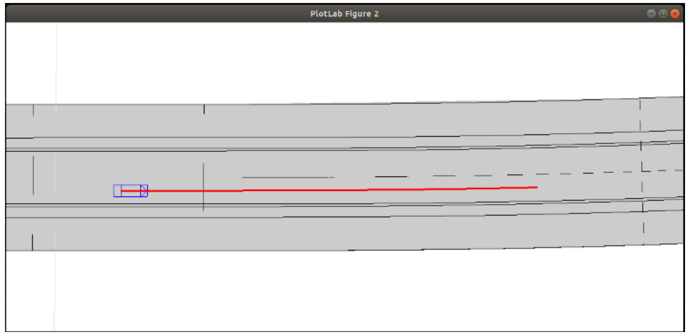
Demo 1: Load xodr map



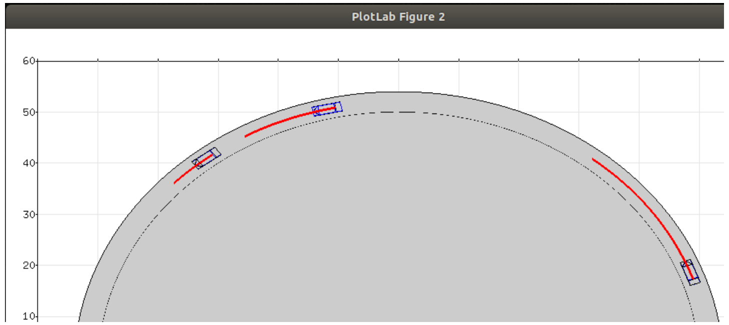
Demo 2: Navigation Function



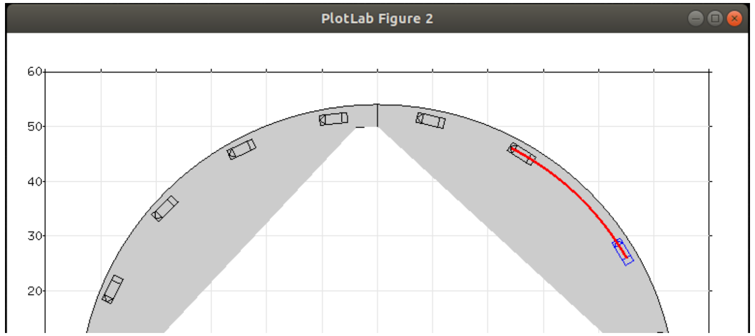
Demo 3: Lane Following



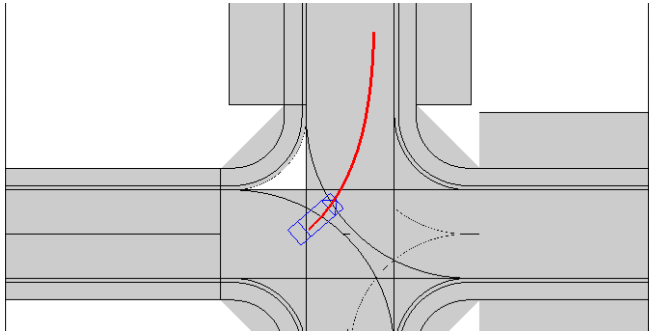
Demo 4: Multi-Vehicle Simulation



Demo 5: SUMO Co-Simulation



Demo 6: Lane Following & Navigation



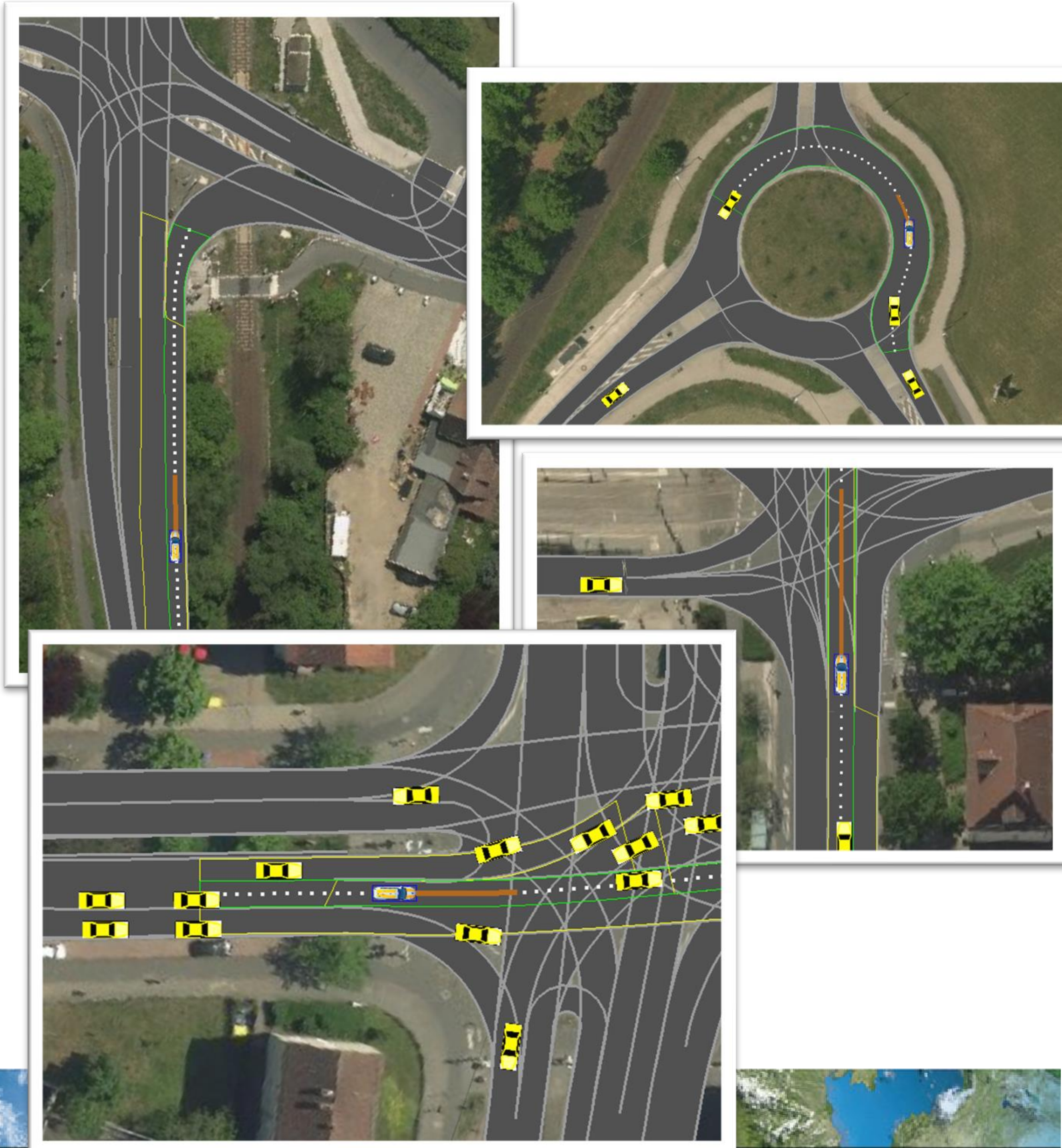
Future Work

V0.2: end of 2020

- ROS Noetic, Ubuntu 20.04
- ITSG5 messages in ROS (.asn1 → .msg)
- V2X Channel model
- Improvements to environment model: Behavior prediction, traffic lights
- Improvements to planning framework: Lane-change and emergency maneuvers, decision making
- Improvement to visualization (Sat-images from Geoserver)

V0.3: 2021

- Continuous development on github
- Integration of machine learning framework
- Coupling with CARLA via ROS
- Automatic simulation testing
- Realworld test scenarios: Braunschweig, Düsseldorf



DLR