SET Level

Simulation-based Development and Testing of Automated Driving

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

PROJECT of the PEGASUS FAMILY

Closed Loop Simulation for AD and how to close the loop?

Marc Heinrich (FZI), Barbara Schütt (FZI), Riccardo Bartolozzi (LBF)

29.04.2021

Closed Loop Simulation

Using the Architecture to close the Loop

SET Level

Part I

- Integrating multiple OSI models into a closed loop simulation outside the Simulation Use Cases
    - Third party tool

Part II

- How to close the loop: modular concepts to close the loop
    - Take use case requirements into consideration
    - Utilize motion control, actuators and vehicle dynamics with variable modelling depth

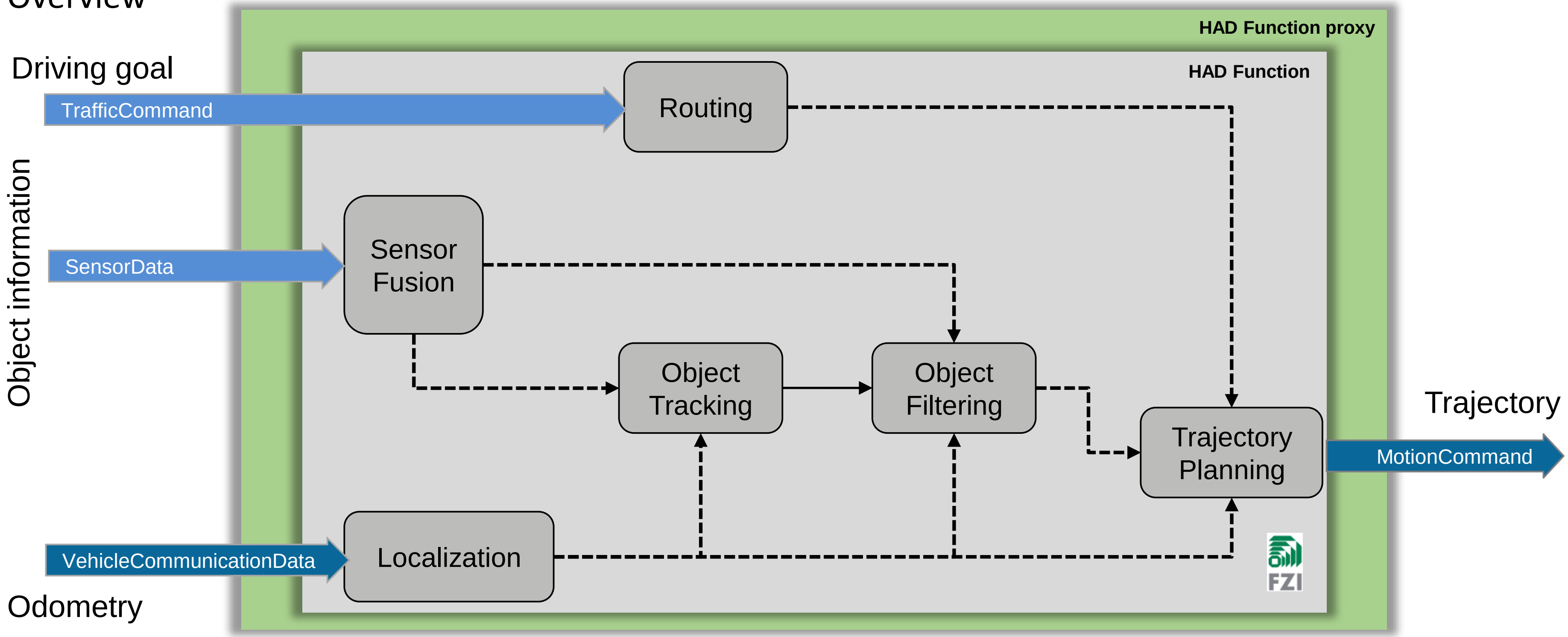
Part I: Closed Loop Simulation - FZI

Intentions

- Integrating multiple OSI models into a closed loop simulation outside the Simulation Use Cases
    - Third party tool
    - Proof of concept of the SET Level interface architecture
    - Not meant as a complete test with traceability and CSP
        - No test case defined
        - Integration is the goal.

Module 1: HAD Function

Overview



fmi Functional Mock-Up Interface | Model FMU | FMU XML C Code Libs

ASAM OSI®

SET Level specific (based on OSI formats)

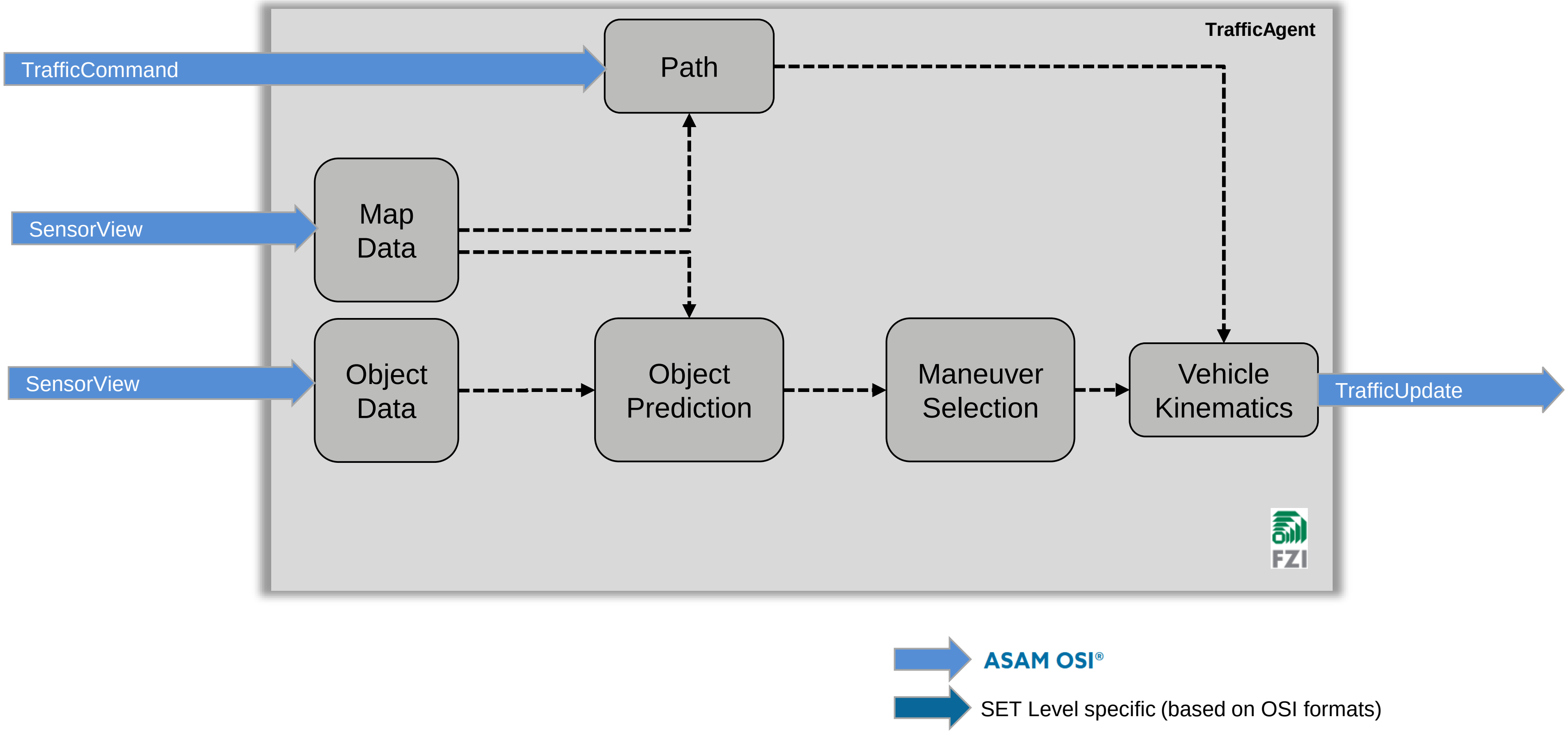
Module 1: HAD Function

Modelling Details

- Characteristics
  - Object list fusion
  - Object tracking with constant velocity & timeout
  - Reaction to road users
    - Differentiation according to classes
    - Relations to ego vehicle and lanes
- Modeling basics
  - Intelligent Driver Model (by Kesting, Driver, Helbing)
  - Extension for tight cornering
- Implementation
  - Modular, distributed system based on ROS
  - OSI messages are converted into equivalent ROS messages

Module 2: Traffic Agent

Overview



Module 2: Traffic Agent

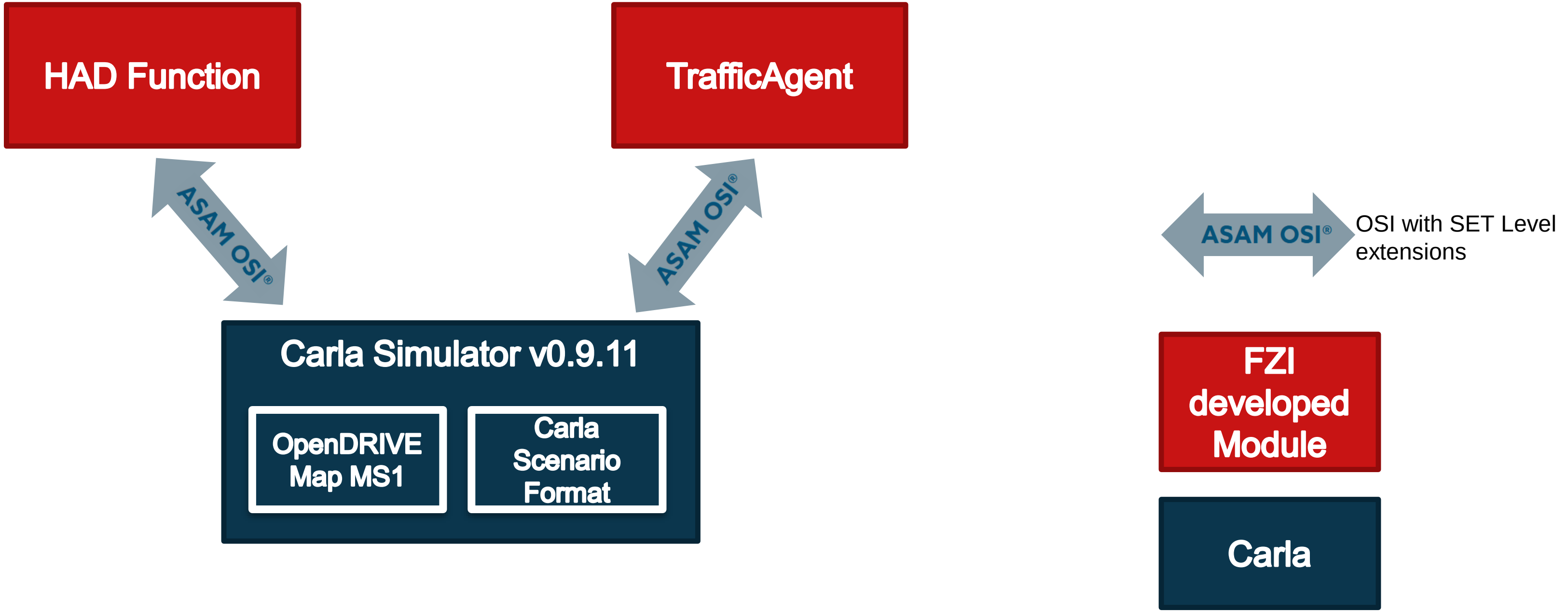
Modelling Details

SET Level

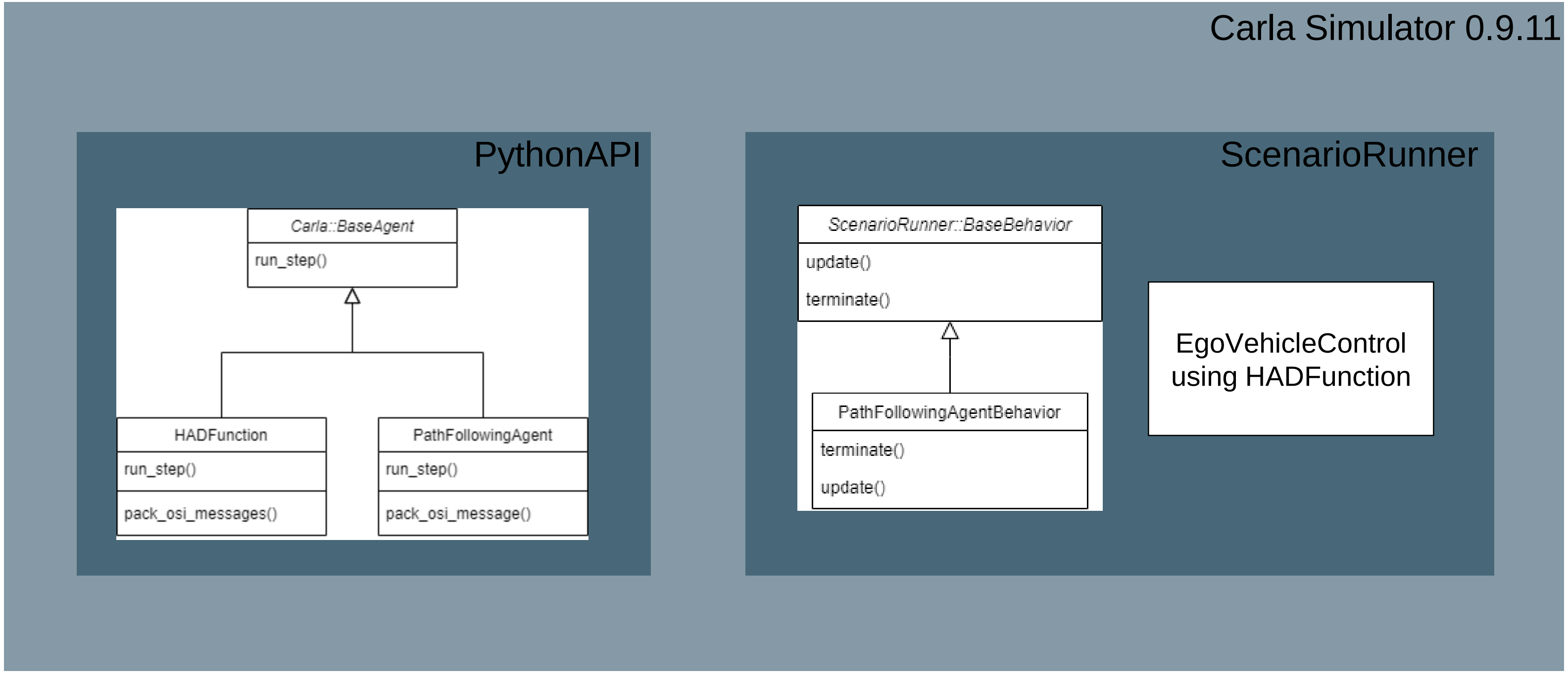
- Characteristics
  - Path based kinematic motion model
  - Rule based object prediction
  - Modeling of interaction behavior at intersections
- Modeling basics
  - Maneuver based game theoretic decision making
- Implementation
  - Interfaces use OSI standard messages only

Carla – TrafficAgent and HAD Function

Overview

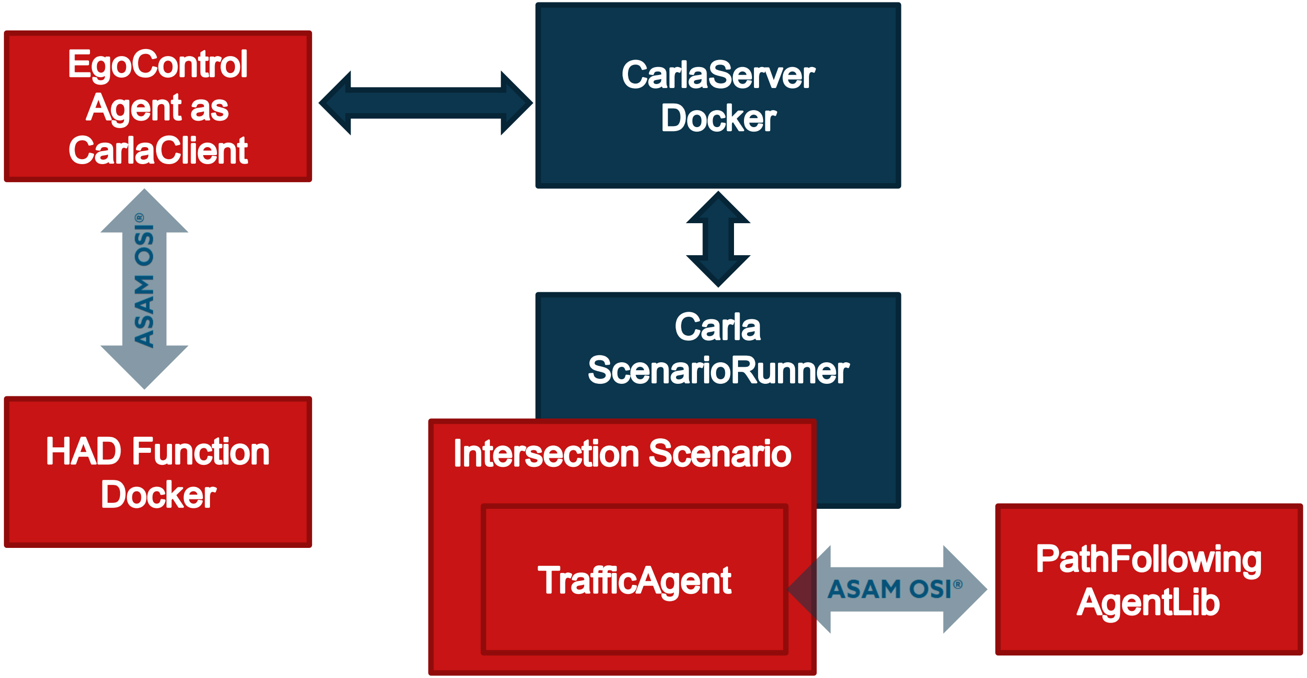


Carla
Integration



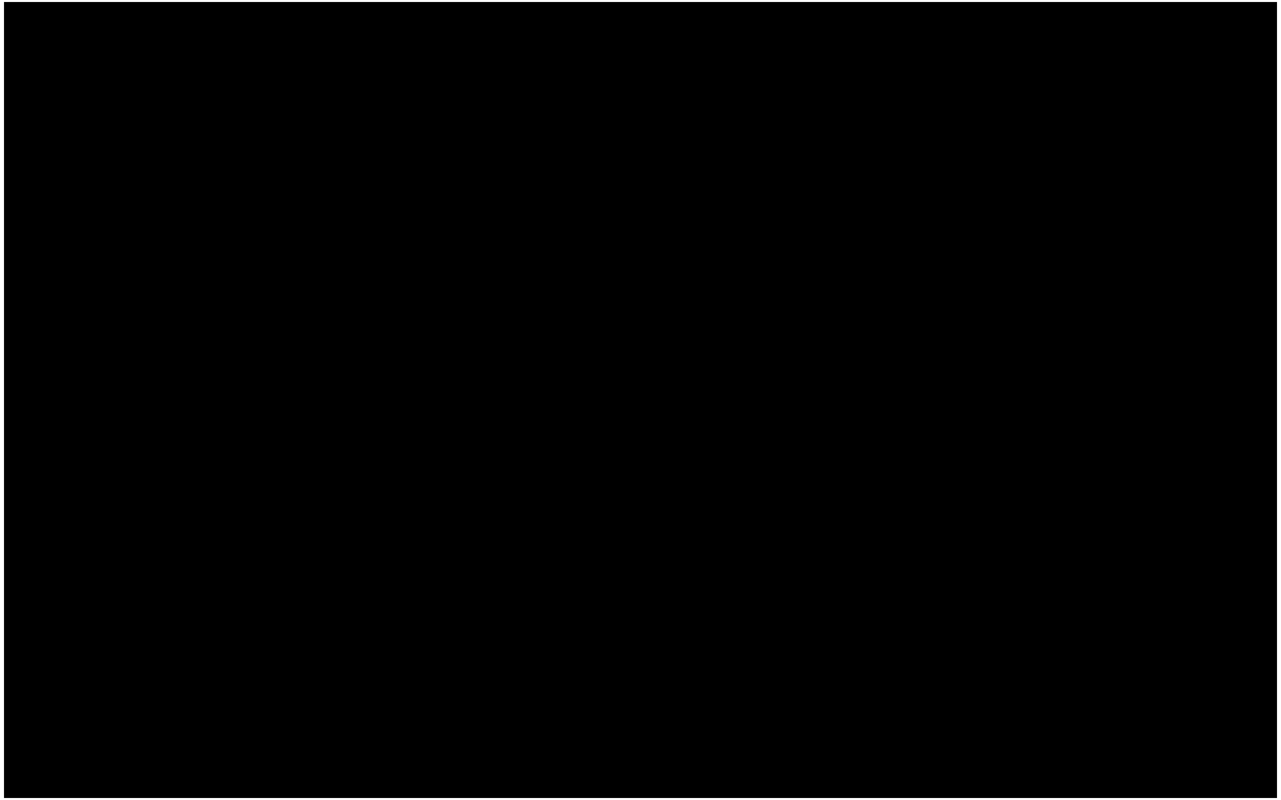
Carla

Complete Setup with Scenario



Video

Example Scenario



Problems and Challenges

- Synchronization between all parties (sync mode vs. asynch mode)
- Ego vehicle not really part of scenario
- Extraction of all needed information for OSI messages

Summary Part I

- Possible to use third party tool with SET Level interface architecture
- Possible to integrate multiple OSI models
- Just a proof of concept
- Simplified closed loop

Depending on the use case, more sophisticated methods to close the loop after the HAD function are needed

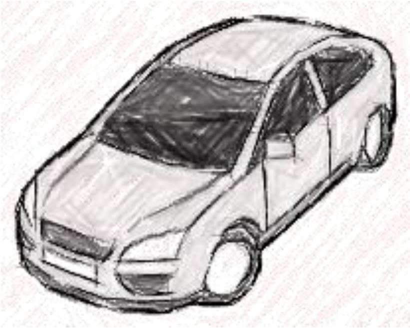
Part II

Motivation Part II

How to close the loop for AD simulation

SET Level

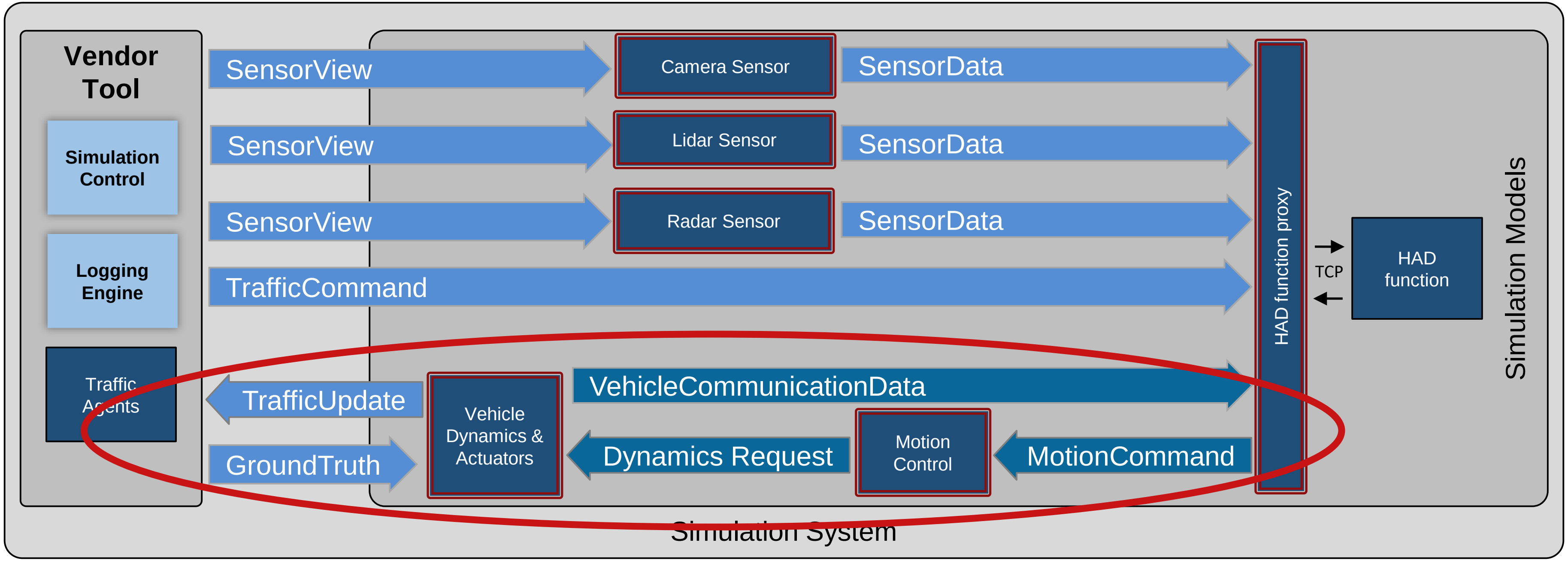
- Scalable solution for closing the loop with EGO vehicle dynamics models
  - Addressing different definitions of system under test (SUT)
  - Different model details required depending on
    - SUT
    - Test specifications



- Implementation of EGO vehicle dynamics models within SET Level project
  - Interface with other SET Level components/partners
  - Support of simulation standards (such as FMI, OSI)
  - Wide acceptance including conformity to best practices and current standards

Architecture

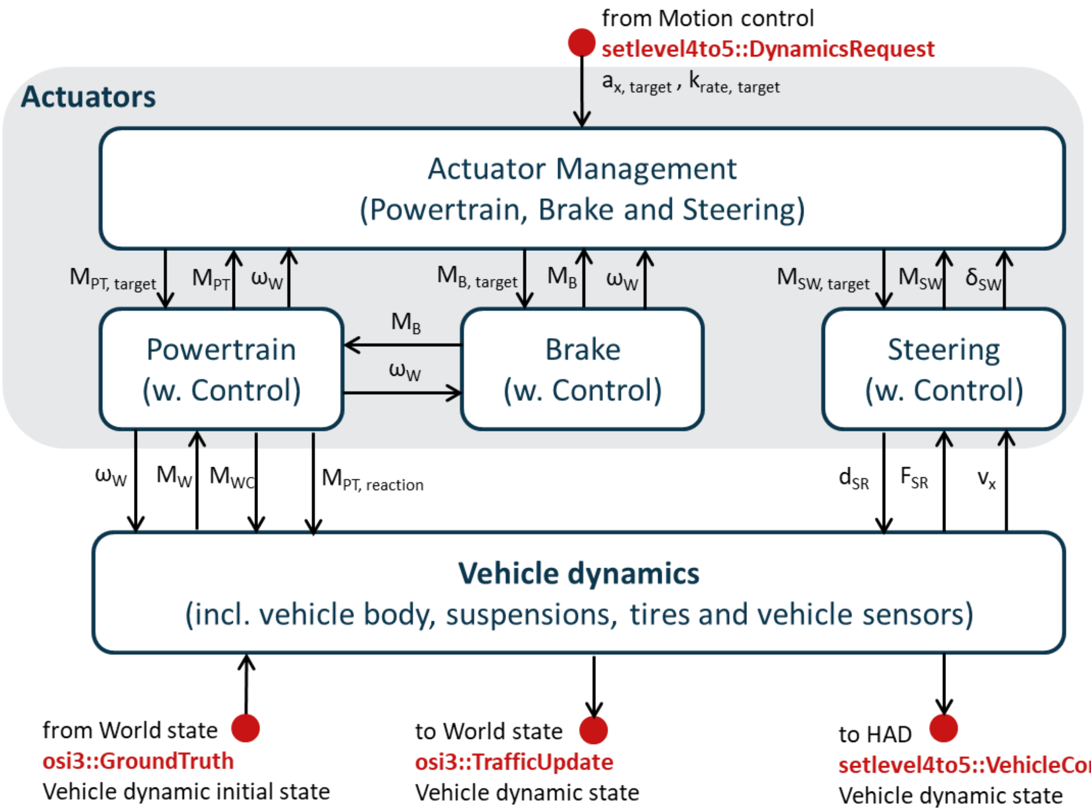
Whole simulation environment architecture from SUC2



Architecture

Internal architecture – Vehicle dynamics and actuators

SET Level

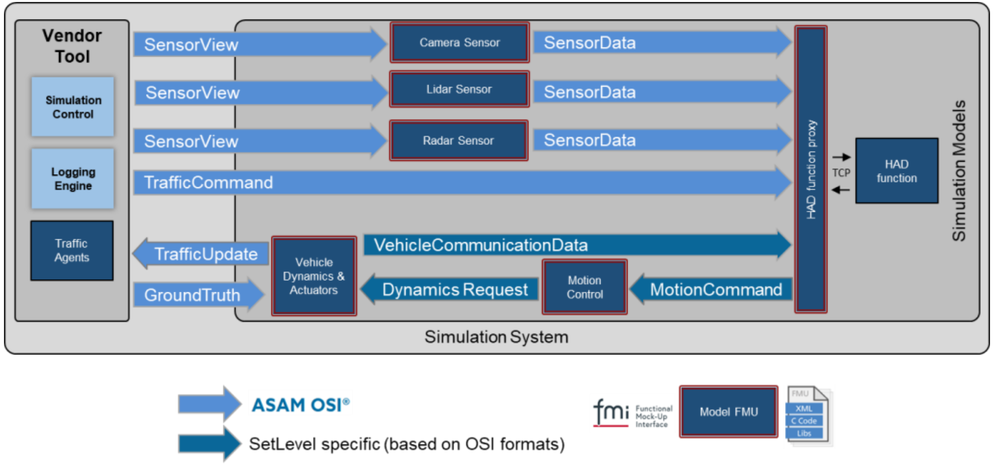


Legend
PT: Powertrain
W: Wheel
SW: Steering wheel
SR: Steering rack
WC: Wheel carrier
B: Braking
F: Force
M: Moment
ω: Rotational velocity
δ: Steering angle
d: Displacement
v: Velocity
a: Acceleration
k: Curvature

Note
Physical quantities (kinematic and force/moments) and main signals are displayed

Interfaces

Overview of implemented OSI-interfaces



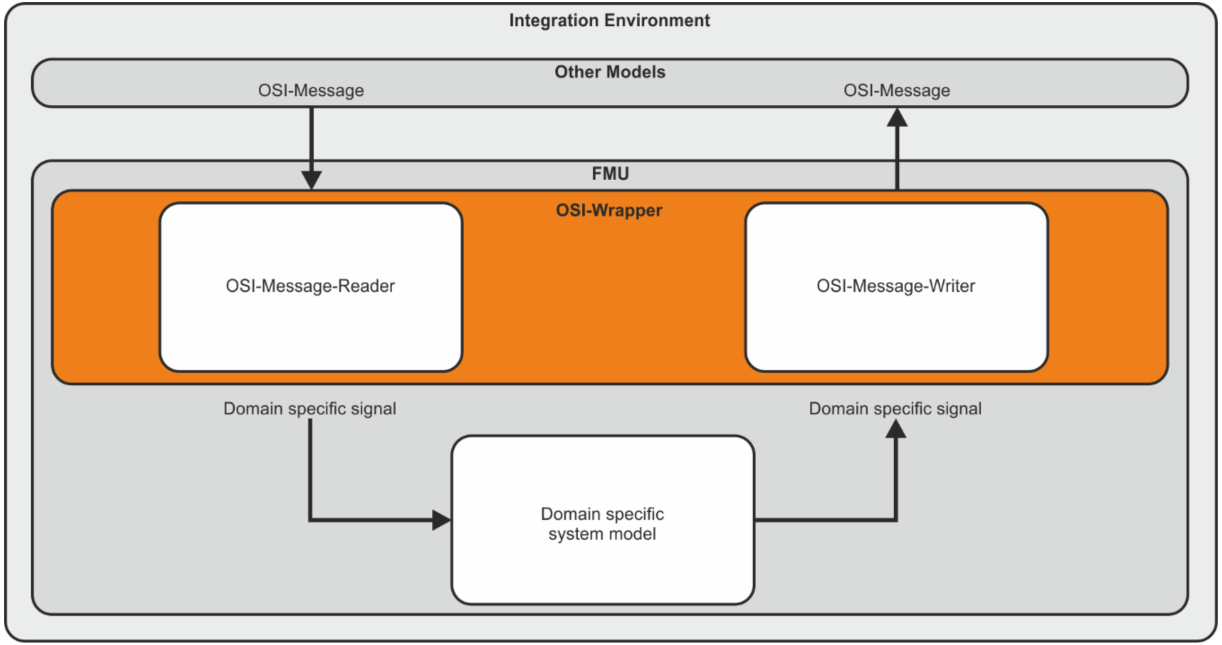
- **GroundTruth**
  - Initialisation message for initial dynamic state
- **DynamicsRequest**
  - Vehicle dynamics target signals (long. acceleration and curvature)
- **TrafficUpdate**
  - Update of vehicle dynamic states (including position, velocity and acceleration vectors of the centre of the bounding box in global coordinate system)
- **VehicleCommunicationData**
  - Similar content to TrafficUpdate → work in progress for possible extensions
- **MotionCommand**
  - Current vehicle dynamic state and target time based trajectory

Interfaces

Implementation of OSI standard in Matlab/Simulink

SET Level

- Need to support development environments of specific component models (such as Matlab/Simulink)
- OSI standard for the interfaces
- FMI standard for model deployment and integration

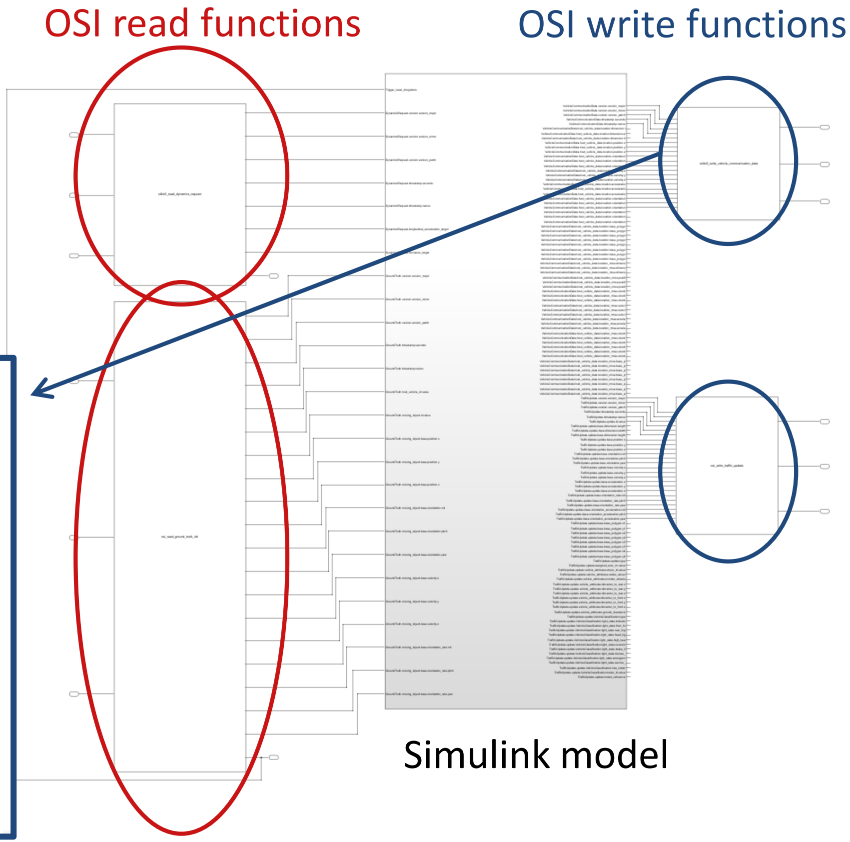


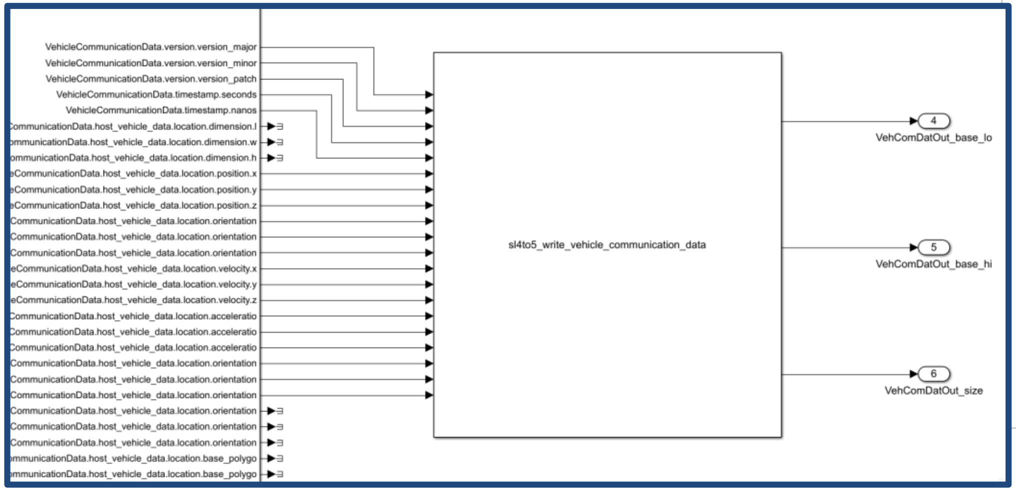
OSI wrapper: C-Code based function compiled as Matlab executable

Interfaces

Implementation of OSI standard in Matlab/Simulink

- Read and write functions of OSI messages
- Simulink interface blocks with required channels for connecting the model
- Model compiled as FMU including the OSI wrapper functions

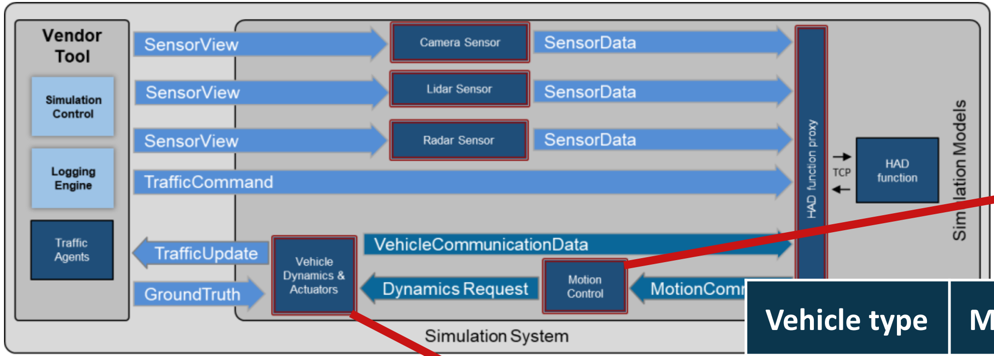




Models

Overview of developed and provided models within SETLevel

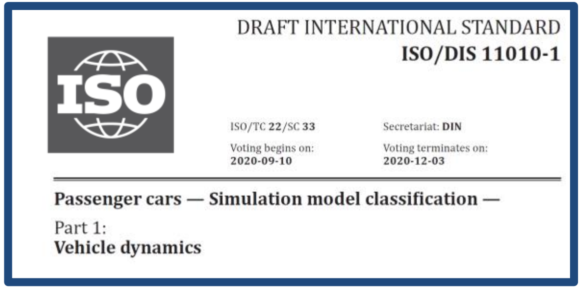
SETLevel



Motion control

Fraunhofer LBF

Conformity to ISO standard for model classification currently under review


DRAFT INTERNATIONAL STANDARD
ISO/DIS 11010-1

ISO

ISO/TC 22/SC 33 Secretariat: DIN

Voting begins on: 2020-09-10 Voting terminates on: 2020-12-03

Passenger cars — Simulation model classification —

Part 1:
Vehicle dynamics

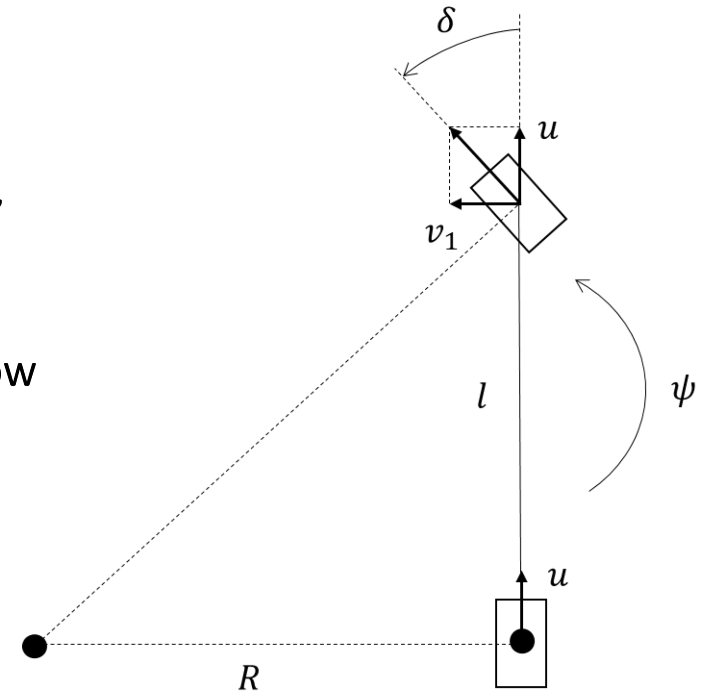
Vehicle dynamics & actuators

Vehicle type	Model detail 1	Model detail 2
Car	Simplified single model (Fraunhofer LBF)	Vehicle dynamics (Fraunhofer LBF)
		Steering (ZF, Fraunhofer LBF)
		Powertrain (ZF, Fraunhofer LBF)
		Brake (Fraunhofer LBF)
		Actuator manag. (Fraunhofer LBF)
Truck	Simplified single model (MAN)	Detailed single model (MAN)

Models

Vehicle dynamics & actuators (car), simplified single model (detail 1)

- Type: 2D kinematical single track model with idealised actuators
- Limits as max. values of: grip, motor torque and power, steering angle and speed
- Model use:
  - Simplified approximation of vehicle dynamics in low acceleration ranges
  - No 3D dynamics should be relevant (i.e. vertical, pitch and roll)

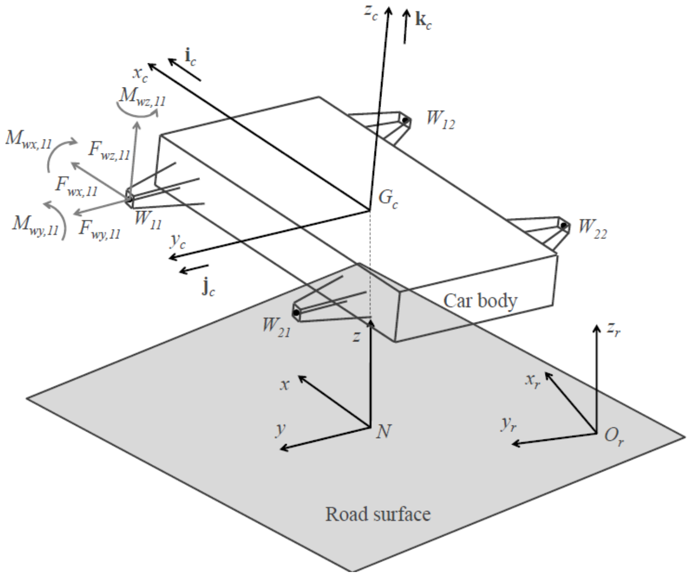


Models

Vehicle dynamics & actuators (car), vehicle dynamics model (detail 2)

SET Level

- Type: reduced multi-body model (14-34 DOFs)
  - Sprung mass (car body): 6 DOFs
  - Unsprung masses: 4-24 total DOFs
  - Wheels (rotation): 4 total DOFs
- (Elasto)-kinematic suspension modelling
- Pacejka handling tire model with vertical compliance
- Model use:
  - Vehicle dynamics low frequencies applications (handling)



Models

Motion Control

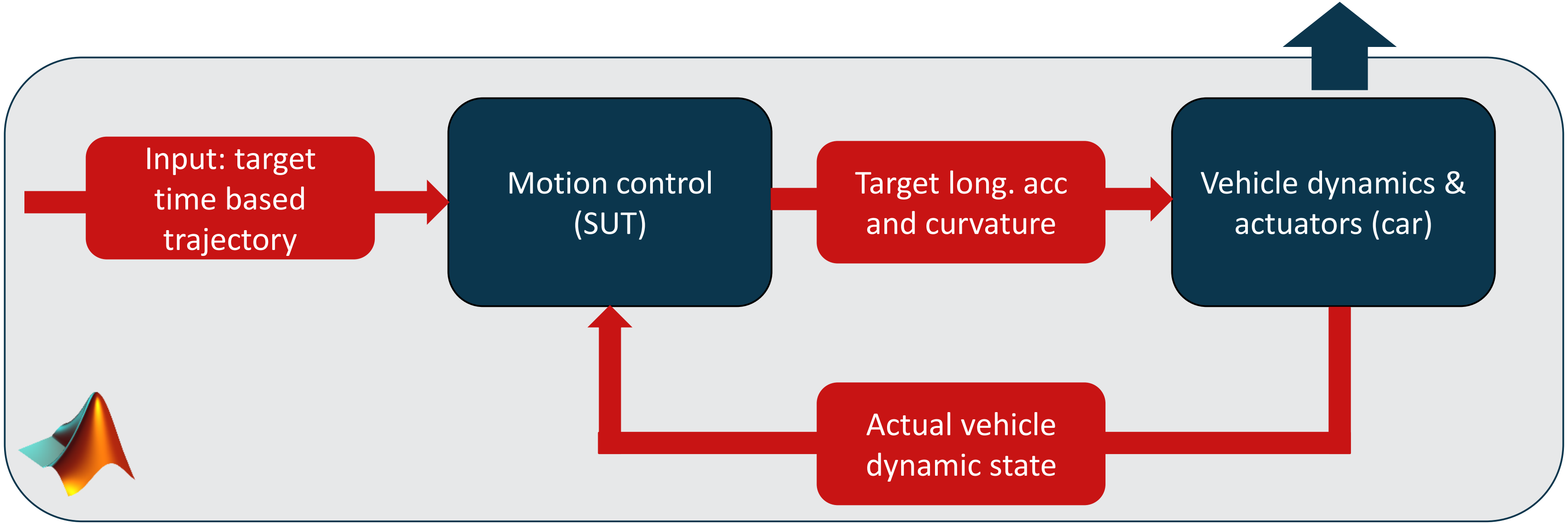
- Type: rear wheel feedback controller (kind of nonlinear state-space controller)
  - Based on kinematic single-track model
- 4 tunable controller parameters, one for each state under control:
  - Longitudinal position, lateral position, heading angle, velocity
- Two controller versions
  - for low computational cost
  - for reduced deviation

Demo simulation

Setup: closed loop vehicle dynamics – motion control

- SUT: motion control with defined parameters
- Scope: investigate influence of model detail of vehicle dynamics
- Demo simulation implemented in Matlab/Simulink @ LBF

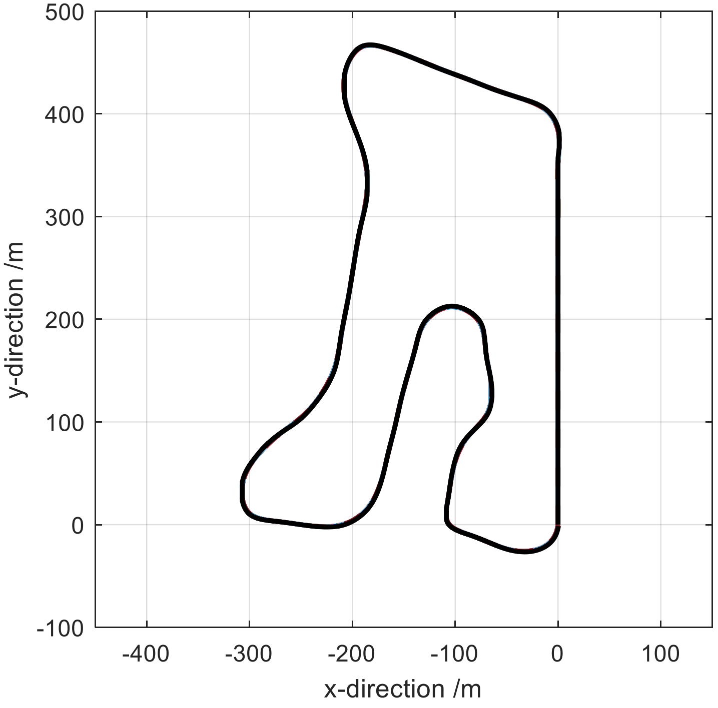
Detail 1: simplified single model
Detail 2: detailed vehicle dynamics model

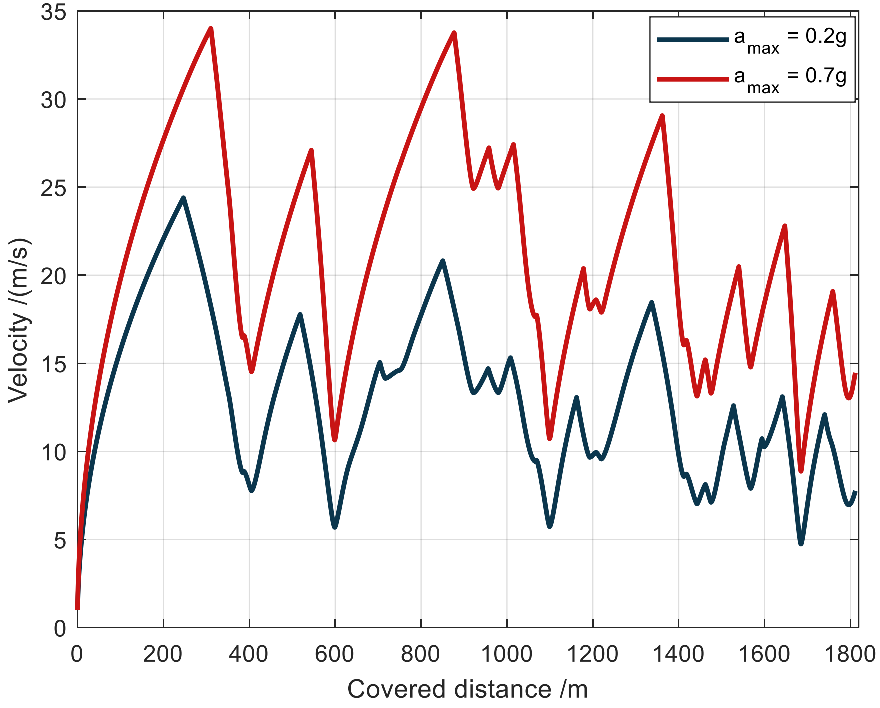


Demo simulation

Results example: handling course

- Two different velocity profiles (low and middle-high acceleration range)

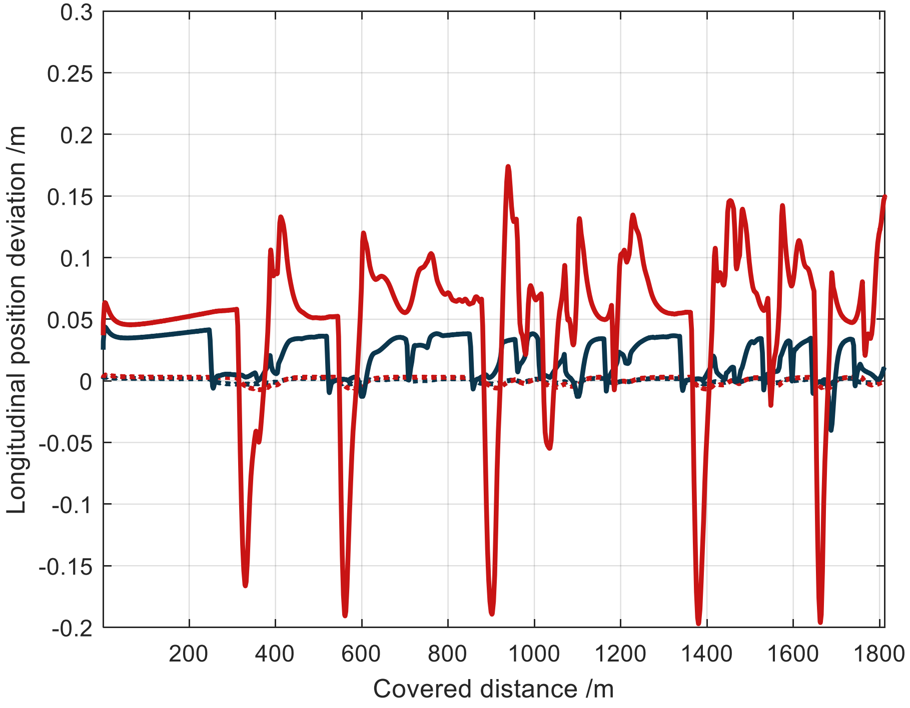


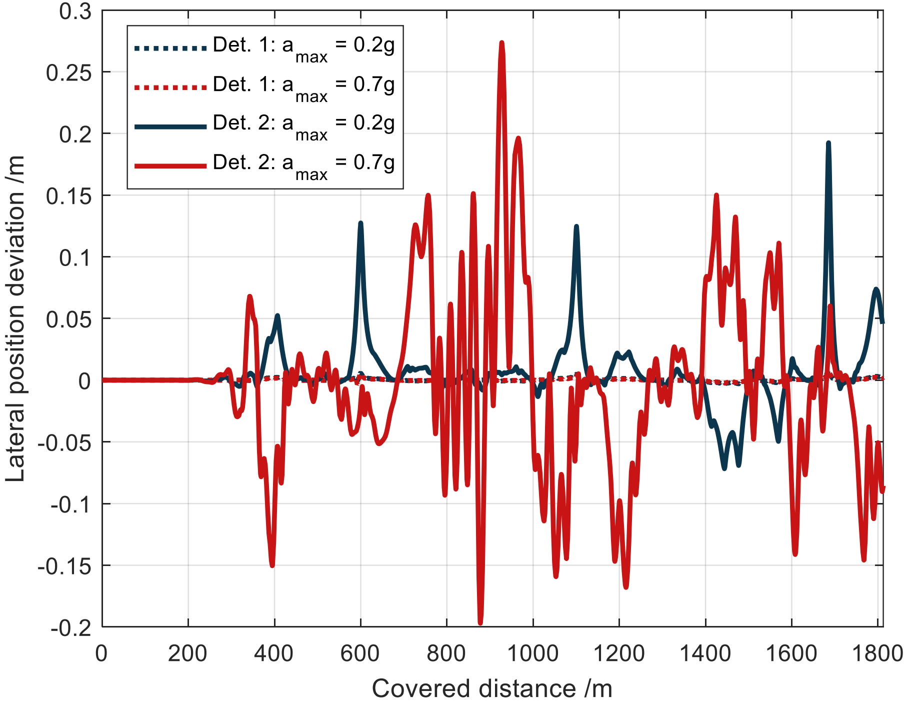


Demo simulation

Results example: handling course

- Two different velocity profiles (low and middle-high acceleration range)





Conclusions and next steps

SET Level

- Proof of concept showed the possibility to implement and use the SET Level architecture in other tools by integrating different models with no extra support required
- Implementation of motion control, vehicle dynamics and actuators models within SET Level focusing on modelling detail
- Possible extension of the shown proof of concept may address a different SUT in order to implement different models of vehicle dynamics and actuators