VDCWorkbench: A Vehicle Dynamics Control Test & Evaluation Library for Model and AI-based Control Approaches

Jonathan Brembeck[1] Ricardo Pinto de Castro[2]
Johannes Ultsch[1] Jakub Tobolar[1]
Christoph Winter[1] Kenan Ahmic[1]

[1]Department of Vehicle System Dynamics and Control, DLR Institute of Vehicle Concepts, Germany, {first.last}@dlr.de

[2] Dept. of Mechanical Engineering, University of California Merced, USA , rpintodecastro@ucmerced.edu

Abstract

The ability to systematically compare and evaluate diverse control strategies is essential for the development of effective control algorithms in autonomous driving. This contribution presents the VDCWorkbench Modelica Library, a unified platform designed to support the development, testing, validation and verification of vehicle dynamics controllers and energy management strategies. The presented library is an extension of the IEEE VTS Motor Vehicle Challenge 2023 models and offers multi-physical component modeling, including a hybrid energy storage system (fuel cell & hydrogen tank and battery with aging model), as well as vehicle dynamics control for autonomous driving research projects. Two path-following approaches are featured: an open-loop lateral controller with a static inversion of a single-track model, and a closed-loop state-dependent geometric path-following controller with static control allocation. The library may also serve as the foundation for development of vehicle control methods, such as two-degree-of-freedom control approaches concepts. One example is given for this combination of a feedforward controller with residual reinforcement learning, where a learned agent improves the performance of the open loop controller. The entire library will be released as open source on GitHub in September 2025.

Keywords: Vehicle Dynamics Control, Lateral Control, Trajectory Following, Reinforcement Learning, Inverse Model, Energy Management, Vehicle Dynamics Simulation, Open Source, Powertrain Simulation, Electric Vehicle, Functional Mockup Interface

1 Introduction

To accelerate the development of control algorithms in the field of vehicle dynamics, efficient modeling tool chains are required to capture key physical effects with reduced computational complexity. In this work, we present the Vehicle Dynamics Control Workbench Library (*VDCWorkbench*), a Modelica-based library that can be used as a workbench for the design, testing and validation of vehicle dynamics controllers and energy management algorithms of electric vehicles. This open-source library gives students, researchers and engineers the opportunity to familiarize themselves with vehicle dynamics modeling and control. The toolbox can be used, for example, for autonomous vehicle projects and assessment of different vehicle configurations.

Other Modelica-based simulation libraries for electric and autonomous vehicles are typically commercial libraries such as Modelon's Vehicle Dynamics, DLRs Powertrain and Alternative Vehicle Library and Claytex's VeSyMA offerings. Outside Modelica, open-source platforms like Gazebo (robotics simulation, can be adapted for vehicle control), CARLA (open-source simulator focusing on automated driving scenarios), Bullet Physics (general rigid and soft body physics, can be used for vehicle dynamics) provide multibody physics, sensor modeling, and scenario generation for vehicle dynamics and control [1]. Most of these previous simulation packages lack support for comprehensive multi-physics modeling of the entire vehicle system and do not allow exporting Functional Mock-up Units (FMUs) for training AI-based controllers—capabilities that are uniquely provided by our *VDCWorkbench* Library. Moreover, tools such as CARLA emphasize high-fidelity scenario visualization for generating synthetic perception data, rather than integrated vehicle-level control and energy management modeling, which are supported by the *VDCWorkbench* library. Note that advanced visualization capabilities can also be provided by the *VDCWorkbench* via the DLR Visualization Library.

A preliminary version of the *VDCWorkbench* library was used as the simulation engine for the IEEE VTS Motor Vehicle Challenge 2023 [2], a research competition for energy management strategies of hybrid electric vehicles.

This library has not yet been published and was only made available to the participants of the challenge as a black box FMU for the development of their energy management algorithms. The entire *VDCWorkbench* library is released as open-source software in the GITHUB repository of the Vehicle System Dynamics and Control department: DLR-VSDC/VDCWorkbench. The library is tested with Dymola 2025x Refresh 1 (64-bit Windows) and also works in OpenModelica v1.25.0 (64-bit Windows) to increase the potential user community for academic projects.

This publication is structured as follows. First, we will discuss the structure of the library with its control modules and the multi-physical modeling of the components. We then describe two concepts for the integration of vehicle dynamics control following a precomputed time-independent parametric path.

2 VDC Workbench Structure

This chapter presents the structure of the *VDCWorkbench* library. Figure 1 shows the schematic structure of the library which was derived and extended from the IEEE VTS Motor Vehicle Challenge 2023 [2]. The library has four types of components: i) vehicle dynamic and actuator models; ii) energy storage models; iii) vehicle control algorithms and iv) energy management algorithms (EMA).

The reference vehicle for the library is an adapted version of the DLR ROboMObil ([3] [4] [5] [6]), which is configured with an simplified drivetrain and a hybrid energy storage system (fuel cell & hydrogen tank and battery). The vehicle has two in-wheel electric motors installed in the rear axle (τ_{RL}, τ_{RR}), one central front motor (τ_F), and a front steer-by-wire actuation (δ). Within the vehicle control block different vehicle dynamics control algorithms are implemented as descripted later in Chapter 3. Some of these vehicle control algorithms can enhance energy efficiency of the vehicle by enabling energy recuperation in both front and rear axle [7] and also decreasing tire slip losses [8]. The in-wheel motors can improve motion control of the vehicle by extending the maximum lateral acceleration [9] and decreases response of inner (yaw-rate) control loops [2]. Additionally, the energy management algorithm (EMA), as developed in the IEEE VTS Motor Challange 2023 [2], sets the control variables for optimizing the energy consumption of the powertrain.

These normalized control variables of the EMA have the following meaning

- $\alpha_{AD} \in [0,1]$ the axle torque distribution ratio between front ($\alpha_{AD} = 0$) and rear axle ($\alpha_{AD} = 1$)
- $\alpha_{TV} \in [0,1]$ the torque vectoring ratio between rear left ($\alpha_{TV} = 0$) and rear right motors ($\alpha_{TV} = 1$)
- $\alpha_v \in [0,1]$ velocity derating factor,
- $\alpha_{FC} \in [0,1]$ the normalized fuel cell current,

For further details on the EMA, please refer to [2]. Several EMA strategies have been developed for this vehicle configuration, ranging from rule-based algorithms, machine learning and optimal energy management (see Ranking of the competition).

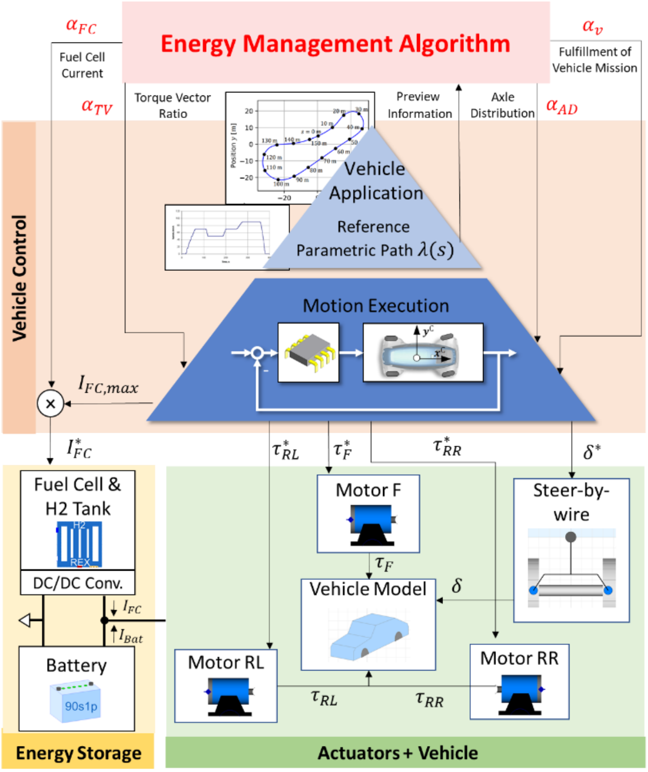

Figure 1. Schematic Structure of the VDCWorkbench Library

2.1 Overall Library Setup and Usage

Figure 2 provides an overview of the sub-packages available in the *VDCWorkbench*. In the `VehicleComponents` package, there is a sub-package for the mechanical structure of the `chassis` (Vehicle Model & Steer-by-Wire Model in Figure 1) based on the PlanarMechanics Library [10] [11].

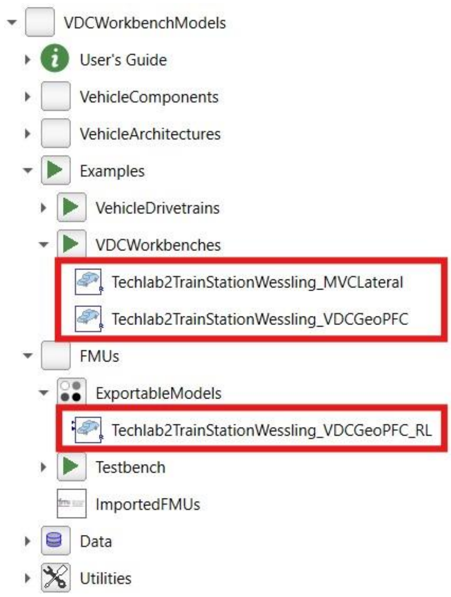

Figure 2. Architecture of the VDCWorkbench Library

The `Powertrain` sub-package contains firstly a model of a quasi-stationary Permanent Magnet Synchronous Machine (PMSM) with an approximated control in the d/q coordinate system. Secondly, a table-based first-order (RC) equivalent circuit battery model with a cell aging model that models the capacity loss over the cell lifetime as a function of the number of charging and discharging cycles, current load and temperature. The third component is a functional quasi-stationary model of a fuel cell with tank. In addition to the vehicle dynamic controllers, which are discussed in detail in Chapter 3, the last sub-package `Controllers` also contains a base class for energy management and the implementation of the baseline energy management strategy as described in Chapter IV.D in [2].

In the sub-package `Examples.VehicleDrivetrains` there are various test configurations to configure the component models with each other to create a multi-physical vehicle and thus build your own vehicle architecture. The examples contained here in `VariantsVehicleDrivetrains` range from a simple vehicle with only one centrally arranged PMSM with differential to a vehicle with four in-wheel hub motors. To test these vehicle configurations, various driving maneuvers are provided. The following two sub-packages `Examples.VDCWorkbenches` and `FMUs` contain complete vehicle models with the baseline energy management algorithm. In addition, there are driving dynamics controllers including test scenarios for experimental investigation or for export to AI training environments. The sub-package `Data` includes all the parameterizations used for the components and controllers of our simplified ROboMObil vehicle model [3]. This is intended to ensure that the models are consistently parameterized and represent a blueprint for your own vehicle configurations. In addition to the bus definitions, which are based on the VehicleInterfaces Library, the last sub-package `Utilities` contains various auxiliary models, such as a sensor model for determining the vehicle dynamics variables required for the controllers.

In order to use our library and simulate the examples, the user must clone the GIT repositories VehicleInterfaces (tested release V2.0.1) and PlanarMechanics (tested release V1.6.0) locally and load them in the Modelica simulator.

2.2 Vehicle Workbench Configuration

In this section, we provide more details about the main model of the *VDCWorkbench* library. The structure of the overall simulation model with the individual building blocks, as shown in Figure 1, is implemented in the model `VDCWorkbench2025_Model` of the sub-package `Examples.VDCWorkbenches` (see Figure 3). The color-coded boxes corresponding to the subcomponents in the schematic sketch. All data relevant for the simulation is stored in the record `data`. The controller block is implemented as replaceable, which we will instantiate, explain and examine in Chapter 3 with various vehicle controllers. The `vehicle`, `drivetrain`, `powerSource` and `baselineEMA` blocks are each taken from the parameterized models of the `VehicleComponent` sub-package.

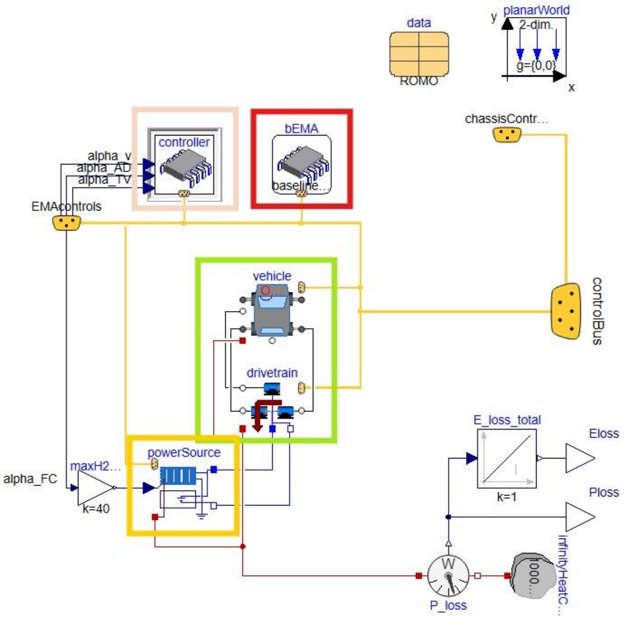

Figure 3. Structure of vehicle components and controllers

2.3 Test Scenario Generation

In order to be able to test our driving dynamics controllers in different scenarios, it is necessary to provide the `controller` block with a parametric path description of the test scenarios. This is defined as a time-independent vehicle movement request along a flat geometric path. For this purpose, we have developed an interactive path planning tool with which parametric paths, subject to constraints, can be generated. This is not planned to be part of the *VDCWorkbench* repository but we are open for inquiries, both for commercial and academic use. Based on an interface to Google Maps (expendable to other map systems such as Here MAPS or Open Street Maps), the reference vehicle path can be created via point-and-click and numerically optimized within the lane boundaries. An example of this is shown here for a section of a path around a roundabout on the route from the DLR site in Oberpfaffenhofen to the next railroad station in Wessling.

Figure 4. Interactive path planning GUI with boarders (green), unoptimized path (blue) and optimized path (red)

As can be seen in Figure 4, the geometric path is described as a two-dimensional polynomial (blue) with its interpolation points (small red dots), that are calculated under boundary conditions for the start and end points. Additionally, an optimization of the path (red) based on energy-optimal criteria with constraints, such as compliance with the road boundary (green), is given [12]. All variables are stored in the path description function $\boldsymbol{\lambda}(s) \in \mathbb{R}^5$, which depends on the path arc length s.

3 Vehicle Dynamics Control

The library also includes two baseline vehicle dynamics controllers designed to regulate both the lateral path and longitudinal velocity of a vehicle. These controllers serve as reference implementations that can be used out of the box or extended with more advanced control strategies, including machine learning methods.

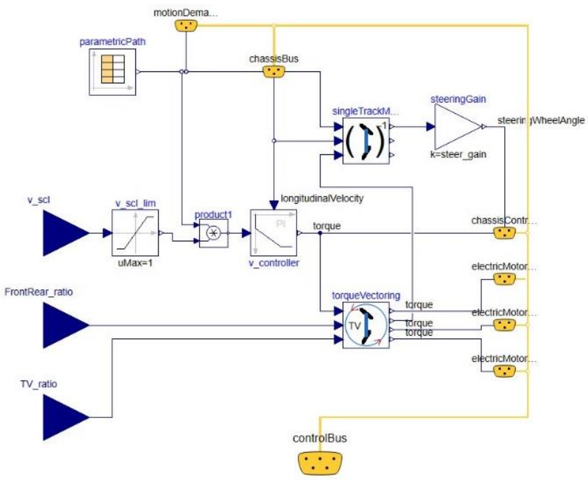

Figure 5. Baseline lateral and speed controller

The first controller is an open-loop lateral controller based on the static inversion of a single-track vehicle model (cf. Figure 5). It calculates the required steering angle to follow a predefined path using the kinematic properties of the vehicle. This is combined with a PI (Proportional-Integral) speed controller, referred to as `v_controller`, which regulates the longitudinal velocity of the vehicle. This modular structure allows the user to separately tune lateral tracking and velocity control loops. Further details can be found in [2].

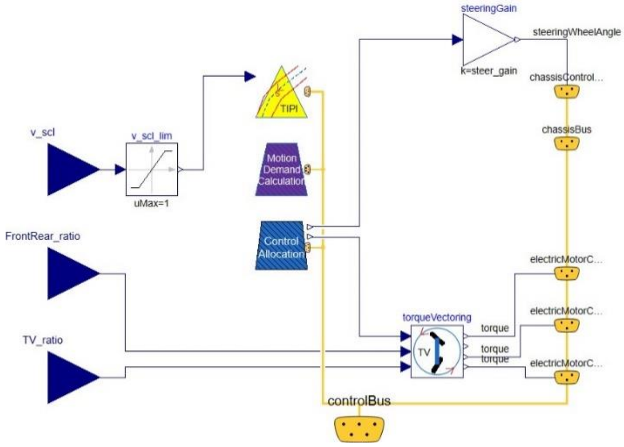

Figure 6. Geometric path following control

The second controller implements a closed-loop geometric path-following strategy (GeoPFC, cf. Figure 6). It relies on a state-dependent computation of the control input, which is then mapped to physical actuator commands through an optimization-based control allocation procedure. This approach allows the controller to adjust the vehicle's motion, depending on its current state with respect to a reference path (e.g., lateral deviation, heading error). For more information, refer to [5] [13].

Example: In Figure 7 we show the lateral and longitudinal deviation of the close-loop controller for our test scenario from Chapter 2.3. While the longitudinal error is minimized very well, particularly in dynamic maneuvers such as the roundabout in the section $s = [900 - 1200]$m there are larger deviations in the lateral error. These can be explained by the very simple control allocation approach, which does not take vehicle and actuator dynamics into account, and the pure front axle steering.

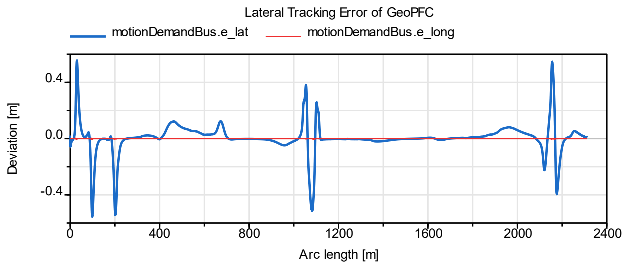

Figure 7. Deviation of controller tracking error

Figure 8 shows the results of the `Control Allocation` and `TorqueVectoring`, including the torque allocated to the traction motors and the steering angle.

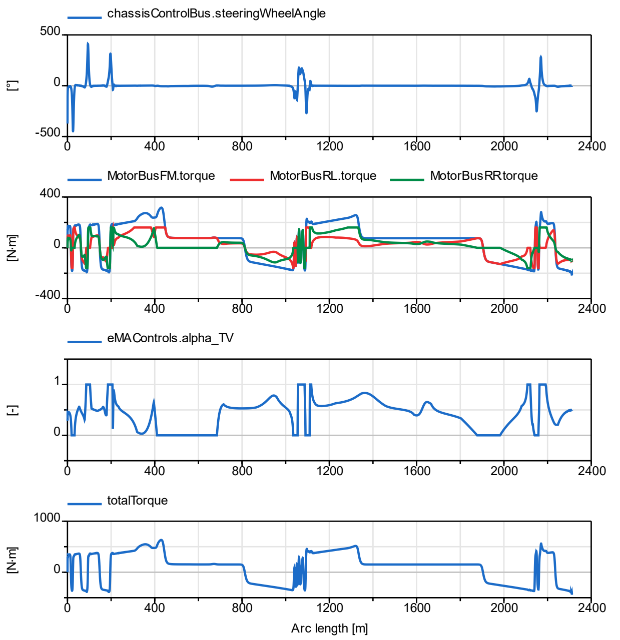

Figure 8. GeoPFC controller actuator demand values

These baseline controllers are designed to be compatible with residual reinforcement learning (rRL) frameworks. In this paradigm, the baseline controller is augmented with a learned agent that contributes an additional control input (red boxes in Figure 9). This residual term allows for performance improvement while preserving the interpretability and domain knowledge of the model-based controller [14] [15].

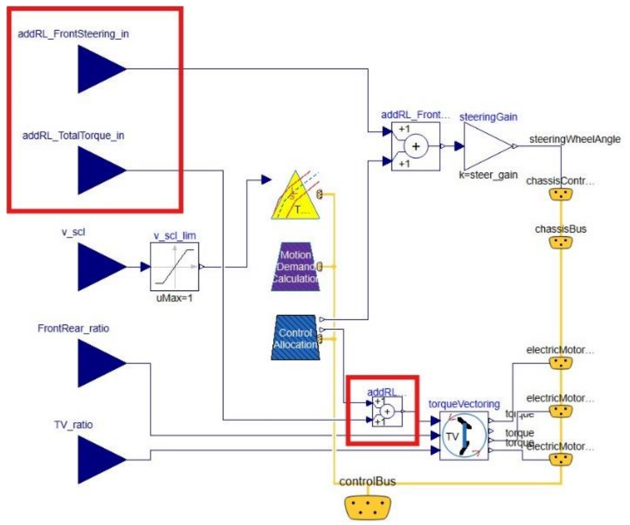

Figure 9. GeoPFC with additional RL agent inputs

4 Summary & Outlook

This work introduces a new Modelica library (*VDCWorkbench*) for simulation of vehicle dynamics and control. It supports developing, testing, and validating control algorithms in hybrid electric vehicles based on the DLR ROboMObil prototype vehicle. The library offers both open-loop and closed-loop vehicle dynamics controllers along a parametric path,. Energy management functions that make use of the multi-physical modeling approach can enhance the control strategy and optimize vehicle energy efficiency. A baseline policy EMA is included, and more sophisticated strategies can be added. *VDCWorkbench* library will be released open source in the DLR VSDC GitHub repository by September 2025, making it broadly accessible for all kind of research applications. It is planned to add further controllers, both for vehicle dynamics and energy management, as well as vehicle models to the library and thus provide the community with further application examples.

Acknowledgements

This research was supported by the DLR project Validation and Verification for the Next Generation Vehicle (V&V4NGC).

References

[1] Y. Li, W. Yuan, S. Zhang, W. Yan, Q. Shen, C. Wang and M. Yang, "Choose Your Simulator Wisely: A Review on Open-Source Simulators for Autonomous Driving," *IEEE Transactions on Intelligent Vehicles*, vol. 9, pp. 4861-4876, May 2024.

[2] J. Brembeck, R. De Castro, J. Tobolář and I. Ebrahimi, "IEEE VTS Motor Vehicles Challenge 2023: A Multi-physical Benchmark Problem for Next Generation Energy Management Algorithms," in *2022 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2022.

[3] J. Brembeck, L. M. Ho, A. Schaub, C. Satzger, J. Tobolar, J. Bals and G. Hirzinger, "ROMO - The Robotic Electric Vehicle," in *22nd IAVSD International Symposium on Dynamics of Vehicle on Roads and Tracks*, Manchester Metropolitan University, 2011.

[4] J. Brembeck and P. Ritzer, "Energy optimal control of an over actuated Robotic Electric Vehicle using enhanced control allocation approaches," in *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium*, Alcala de Henares, Spain, 2012.

[5] J. Brembeck, "Model Based Energy Management and State Estimation for the Robotic Electric Vehicle ROboMObil," Ph.D. dissertation, Technical University of Munich, Munich, Germany, 2018.

[6] DLR Vehicle System Dynamics Blog, "10 years ROboMObil," 08 2022. [Online]. Available: https://vsdc.de/en/robomobil-timeline/.

[7] R. de Castro, M. Tanelli, R. E. Araújo and S. M. Savaresi, "Design of safety-oriented control allocation strategies for overactuated electric vehicles," *Vehicle System Dynamics*, vol. 52, pp. 1017-1046, 2014.

[8] J. Torinsson, M. Jonasson, D. Yang and B. Jacobson, "Energy reduction by power loss minimisation through wheel torque allocation in electric vehicles: a simulation-based approach," *Vehicle System Dynamics,* vol. 60, pp. 1488-1511, 2022.

[9] B. Lenzo, "Torque Vectoring Control for Enhancing Vehicle Safety and Energy Efficiency," in *Vehicle Dynamics: Fundamentals and Ultimate Trends*, B. Lenzo, Ed., Cham, Springer International Publishing, 2022, p. 193–233.

[10] D. Zimmer, "A Planar Mechanical Library for Teaching Modelica," in *Proceedings of the 9th International Modelica Conference*, Munich, Germany, 2012.

[11] D. Zimmer and M. Otter, "Real-time models for wheels and tyres in an object-oriented modelling," *Vehicle System Dynamics,* 2009.

[12] C. Winter, R. de Castro and T. Bünte, "Experimental Validation of Online Motion Planning for Semi-Autonomous Vehicles," in *2022 IEEE Vehicle Power and Propulsion Conference, VPPC 2022*, 2022.

[13] P. Ritzer, C. Winter and J. Brembeck, "Experimental Validation of Geometric Path Following Control with Demand Supervision on an Over-Actuated Robotic Vehicle," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, 2016.

[14] J. Ultsch, K. Ahmic and J. Brembeck, "train_fmu_gym: A Functional Mock-up Unit-based Framework to Train Reinforcement Learning Agents for Multi-Physical Systems," *Manuscript submitted for publication,* 2025.

[15] J. Ultsch, J. Brembeck and R. de Castro, "Learning-Based Path Following Control for an Over-Actuated Robotic Vehicle," in *VDI-AUTOREG*, Düsseldorf, 2019.