applied sciences

MDPI

Article

Reinforcement Learning for Semi-Active Vertical Dynamics Control with Real-World Tests

Johannes Ultsch *, Andreas Pfeiffer, Julian Ruggaber, Tobias Kamp, Jonathan Brembeck and Jakub Tobolář

Institute of System Dynamics and Control, German Aerospace Center (DLR), 82234 Weßling, Germany; andreas.pfeiffer@dlr.de (A.P.); julian.ruggaber@dlr.de (J.R.); tobias.kamp@dlr.de (T.K.); jonathan.brembeck@dlr.de (J.B.); jakub.tobolar@dlr.de (J.T.)

* Correspondence: johannes.ultsch@dlr.de

Abstract: In vertical vehicle dynamics control, semi-active dampers are used to enhance ride comfort and road-holding with only minor additional energy expenses. However, a complex control problem arises from the combined effects of (1) the constrained semi-active damper characteristic, (2) the opposing control objectives of improving ride comfort and road-holding, and (3) the additionally coupled vertical dynamic system. This work presents the application of Reinforcement Learning to the vertical dynamics control problem of a real street vehicle to address these issues. We discuss the entire Reinforcement Learning-based controller design process, which started with deriving a sufficiently accurate training model representing the vehicle behavior. The obtained model was then used to train a Reinforcement Learning agent, which offered improved vehicle ride qualities. After that, we verified the trained agent in a full-vehicle simulation setup before the agent was deployed in the real vehicle. Quantitative and qualitative real-world tests highlight the increased performance of the trained agent in comparison to a benchmark controller. Tests on a real-world four-post test rig showed that the trained RL-based controller was able to outperform an offline-optimized benchmark controller on road-like excitations, improving the comfort criterion by about 2.5% and the road-holding criterion by about 2.0% on average.

Keywords: reinforcement learning; vertical dynamics control; semi-active damping; FMI; Modelica

Citation: Ultsch, J.; Pfeiffer, A.; Ruggaber, J.; Kamp, T.; Brembeck, J.; Tobolář, J. Reinforcement Learning for Semi-Active Vertical Dynamics Control with Real-World Tests. *Appl. Sci.* **2024**, *14*, 7066. https://doi.org/10.3390/app14167066

Received: 28 June 2024
Revised: 26 July 2024
Accepted: 29 July 2024
Published: 12 August 2024

1. Introduction

Vehicle vertical dynamics control aims at improving ride comfort as well as maximizing ride safety by improving road-holding. Given the contrasting nature of these two objectives, at a certain point enhancing one metric yields a corresponding deterioration in the other. In contrast to a pure passive suspension system setup, semi-active and active suspension systems are able to achieve a better tradeoff between ride comfort and road-holding [1]. While the performance of fully active systems is superior to that of semi-active systems, the energy consumption of semi-active systems is significantly lower. Since semi-active suspension systems provide a good tradeoff between performance and energy consumption and are widely used in production vehicles, this work focused on the control of a road vehicle equipped with a semi-active suspension setup.

Several aspects of semi-active suspension systems pose a major challenge to the development of suitable control algorithms for such setups. On the one hand, the limited actuation range, the nonlinear damper characteristics as well as a complex input to force dynamics result in a nonlinear system. On top of that, the usage of rubber bearings in the suspension system and the elastic bearing of the engine mass affects the dynamic behavior of the overall dynamic behavior. On the other hand, the simultaneous application of the damper force to both the chassis mass and the wheel mass, the additional coupling via the suspension spring, and the unknown road excitation further complicate the control problem.

Despite the complex system dynamics, many vertical dynamics controllers are based on linear two-mass oscillators and incorporate other simplifications. The widely used skyhook controller (SH) introduced in [1] assumes a virtual control force acting on the vehicle body and neglects the induced force acting on the wheel. The groundhook control law (GH) introduced in [2] extends this concept by a virtual force acting on the wheel to control wheel oscillations. In ref. [3], the authors compare several semi-active suspension control methods on a linear two-mass oscillator vehicle model. Even though the incorporation of a first-order input dynamics results in a nonlinear quarter-vehicle model (QVM), a linear damper characteristic is assumed. In ref. [4], a simplified damper model is used during the controller design process and a more sophisticated damper model is used during simulative assessment.

In contrast to analytically derived control laws, data-driven methods such as Deep Reinforcement Learning (DRL) have been increasingly used for solving complex real-world control tasks in recent years, e.g., for quadruped robots [5]. The increasing popularity of applying Reinforcement Learning (RL) methods to control problems is based on multiple advantages: Multi-physical simulation models can be directly used within the training process. Many analytically derived control laws rely on specific simplified model structures, e.g., linear models or state space models. Moreover, deriving analytical control laws is often a tedious and error-prone task that requires expert knowledge. RL methods are rather generic and can solve a wide range of control problems based on a repeated interaction with the system or a simulation model of the system. In contrast to first-generation RL algorithms, the increased popularity of RL is based on the use of artificial neural networks (ANN) as function approximators within DRL algorithms. The use of ANNs leveraged the performance of RL and enabled the applicability of RL algorithms to a wide range of sequential decision-making problems across different domains and research fields [6].

In this work, we applied RL to semi-active suspension systems for vertical vehicle dynamics control in the German Aerospace Center's (DLR's) research vehicle AI For Mobility (AFM) [7]. Since the training in simulation is fast, scalable, and safe, our approach relies on a simulation-based training strategy. We addressed the entire Reinforcement Learning design process for the vertical dynamics control of semi-active damper systems. This included the modeling of the vehicle as an assembly of QVMs. To ensure sufficiently accurate training models, the model structure and the model parameters were optimized using real-world measurement data. The training process covered different road excitations as well as the comparison of different reward function designs. After the deployment on an embedded rapid control prototyping system, the RL-based controller was benchmarked against a combined SH/GH controller. The parameters of the SH/GH controller were obtained in an offline nonlinear optimization setup to assure a fair comparison. Additionally, evaluation on a real-world four-post test rig ensured reproduceable real-world test results. The real-world tests showed that the RL agent was able to outperform an offline optimized benchmark controller in the comfort criterion in seven out of nine road-like excitations and improved the road-holding criterion in all tested road-like excitations.

1.1. Related Work

An early application of RL to the vehicle suspension control is documented in [8]. In this contribution, the authors used a stochastic continuous action RL automata algorithm to obtain the parameters of a simple linear control law. The training was performed in a real-world setting on a four-post test rig with the aim of minimizing the root mean square (RMS) of the chassis acceleration. In the approach presented in the paper, the RMS of the chassis acceleration could be minimized in comparison with a fixed damper characteristic. In contrast to our approach, the policy utilized in the abovementioned paper was a pure linear control law and the RL algorithm could only adjust three parameters per vehicle corner. We utilized an ANN as a policy that enabled the RL algorithm to obtain a much more sophisticated control law.

In ref. [9], the authors applied a Batch RL algorithm to the semi-active damping of a vehicle. In their contribution, the RL algorithm was trained in a simulation on a QVM. The applied RL algorithm used tree-based regression methods as function approximators. Additionally, the authors selected a discrete action space, i.e., the RL algorithm could only choose between minimum and maximum damping. The goal of the approach was to maximize ride comfort. To achieve this, the authors compared three different reward functions: one was based on the vertical chassis acceleration, one was based on the vertical chassis velocity, and one was based on the vertical chassis displacement. The authors found that the velocity-based reward function performed best. Simulative assessments showed that the trained controller was able to outperform their benchmark controller, a combined SH- and acceleration-driven damping controller (SH-ADD controller, c.f. [4]), in low frequencies, but performed slightly worse on higher frequencies. In contrast to [9], we used a deep RL algorithm and utilized a more sophisticated training model. Additionally, we validated our trained controller in real-world tests, which was not performed in [9].

An explicit model predictive control (MPC) approach was presented in [10]. In this contribution, the authors designed an MPC for a nonlinear QVM incorporating nonlinear axle kinematics, damper friction, and nonlinear damper characteristics. The MPC was then solved offline for various sampled points from the state space and the result was stored. After that, an ANN was trained such that a damper input could be obtained by evaluating the ANN with measurement inputs. Experiments on a quarter vehicle test rig and in a full vehicle simulation showed the benefit of the proposed approach compared to a combined SH/GH controller. In general, the explicit MPC approach was tractable as long as the state space was small. The computational effort grew exponentially with the number of states and, thus, became computationally intractable very fast. Additionally, it was not straightforward to generate the sample pattern for the state space. On the one hand, the samples should cover all regions that might be encountered during operation. On the other hand, oversampling increases the computational cost and, thus, should be kept at a minimum. In the RL approach presented in the contribution at hand, no sample grid had to be selected manually.

The authors of [11] proposed a sequential learning algorithm to iteratively optimize the parameters of a predefined policy. In contrast to standard DRL, the proposed algorithm used a policy that is parametrized as a quadratic function of all available measurements. The authors showed that their approach was superior to a linearized variant of the skyhook control algorithm in real-world tests.

In addition to the non-standard DRL approaches to tackle the vertical dynamics control problem via the learning-based approaches listed above, many applications of standard DRL algorithms can be found in the literature. In refs. [12,13], the deep deterministic policy gradient (DDPG) RL algorithm was applied to the vertical dynamics control of a QVM. Both used a linear QVM to train the controller and assess the trained agent in simulation.

The contributions of [14,15] applied the proximal policy optimization (PPO) DRL algorithm to vehicle vertical dynamics control. Both contributions trained the agent on a linear QVM that was extended by a friction term resulting in a nonlinear model. In ref. [15], the authors trained the agent on a single road bump excitation, while it seems in [14] the agent was trained on artificial road excitations. The work of [14] compared the trained agent in a simulation against a passive- and a fuzzy-based control strategy and was able to show some improvements for different road types and vehicle loading conditions. A simulative evaluation in [15] was conducted on a bump excitation and showed that the PPO agent was able to minimize the momentum of the unsprung wheel mass but concurrently increased the momentum of the body mass.

In ref. [16], the ability of RL to handle an uncertain delay of the input was investigated. The authors applied the twin delayed DDPG (TD3) algorithm to a linear QVM extended by an input delay. Even though an input delay violates the Markov assumption that is assumed in the theoretical analysis of RL algorithms, the authors were able to show an advantage of their agent compared to a pure passive suspension setup in a simulative

assessment. Compared to a DDPG agent, the TD3 agent proposed in the contribution outperformed the DDPG agent in some simulative scenarios.

The comparative study in [17] benchmarked different DRL algorithms with different state-of-the-art vertical dynamics controllers. To train the agents, the authors used a nonlinear two-mass oscillator with the damping coefficient as input and a first-order actuator dynamics. The authors found that the use of trust region policy optimization (TRPO) with generalized advantage estimation (GAE) yielded a close-to-optimal policy and was advantageous in comparison with other RL algorithms. The simulative assessment on the QVM was supported by further investigation of the selected agent on a nonlinear full-vehicle model.

In ref. [18], different DRL algorithms were trained on a nonlinear full-vehicle model excited by synthetical road profiles. An evaluation of different DRL algorithms showed that the soft actor–critic (SAC) algorithm performed best in their setting. The authors proposed a dual approach in which one agent is trained to cope with the road excitation according to [19] and another agent is trained for impulse excitations. Additionally, an impulse detector algorithm was developed, based on which one of the two agents was selected for application. The proposed approach was evaluated in simulation and showed an improvement in the comfort criterion compared to an SH controller, two different MPC versions, and a passive setup. Additionally, the RL-based agent that was trained for handling the road excitations was tested on a real test circuit. In these real-world tests, the SAC agent was able to outperform the SH controller and a passive setup on the RMS of the vertical acceleration. In our work, we describe the derivation of the training model in great detail. Additionally, instead of choosing to test the agent in a real-road setting, we evaluated our trained agent on a four-post test rig. This way, we were able to not only measure the chassis vertical acceleration as a basis for the comfort criterion but were also able to measure the wheel loads and could derive the road-holding criterion from these measurements.

1.2. Contribution and Overview of This Work

In this work, we propose one way of applying DRL to the semi-active suspension control of a real street vehicle. We describe the entire process of the vertical dynamics RL controller design process, which is presented in more detail in Section 2. This process started with taking measurements of the whole vehicle on a four-post test rig and the measurements of individual components. Based on these measurements, we derived component models and optimized a QVM model that was later used for training. The modeling, identification, and evaluation are presented in Section 3. Thereafter, Section 4 describes the whole controller training process. The verification of the controller trained on QVMs in a full-vehicle model is discussed in Section 5. Finally, Section 6 presents the results of conducting real-world tests.

As this work is partially based on the results of a funded project, parts of the contribution were published as a short summary in the German final report [20]. On top of that, the contributions of the paper at hand can be summarized as follows:

1. We address the complete process of applying DRL to the semi-active suspension control problem in great detail. This process includes taking measurements, deriving a training model, training the controller, verifying the controller in simulation, and conducting real-world tests.
2. In our approach, we propose to optimize the QVM model structure as well as the QVM parameters in order to obtain an accurate training model. Additionally, we show that the optimized model structure is able to approximate the real measurement data better than a standard two-mass QVM.
3. We propose to train the controller on different QVMs, which represent the different corners of the vehicle to avoid overfitting. Additionally, we train on different excitation types to make the resulting controllers more robust. The whole training process,

including the design of the reward function and the selection of the trained agent, is presented in great detail.

4. We evaluate the resulting controller in real-world tests on a four-post test rig. The selected RL agent was able to outperform an offline-optimized benchmark controller on road-like excitations, improving the comfort criterion by about 2.5% and the road-holding criterion by about 2.0% on average.

2. The Vertical Dynamics RL Controller Design Process

There are two main options for training an RL controller: training in simulation or training directly on the real plant, i.e., the test vehicle. The benefit of training in the real world is that no simulation model is needed and that no model–reality mismatch (sim-to-real gap, see [21]) can occur. Nevertheless, ensuring safety can pose a challenge. Additionally, the training is restricted to real time. In contrast to that, training in simulation is fast, scalable, and safe. Simulation models often can be simulated faster than real time and the training can be parallelized on high-performance computing systems, which provides an additional speed-up. Because of these advantages, training in simulation is often preferred. In some applications, the agent trained in simulation is afterwards trained on the real system to overcome the sim-to-real gap.

In order to train a vertical dynamics controller directly on the vehicle, it is necessary to excite the vehicle with defined vertical excitations. Since this is not possible in normal road traffic, a so-called four-post test rig, as depicted in Figure 1, is required for real-world training. However, the experiments on the four-post test rig and equipping the vehicle on the test rig present a considerable effort. Starting directly with real-world training is unfavorable for the vertical dynamics control problem, since the training process may have to be repeated several times to find a robust training setup. Due to these limitations, we chose to train the controller in simulation. To narrow the sim-to-real gap, the training model structure as well as the parametrization were optimized with respect to extensive measurement data.

Figure 1. The DLR's test vehicle AFM on a four-post test rig (adopted from [20]).

The controllers were developed for and tested on the AI For Mobility (AFM) research platform [7], a test vehicle at the German Aerospace Center that enables the investigation of AI-based control methods. A hybrid production vehicle served as a test platform. To provide the data required for AI methods, the vehicle was equipped with an extensive range of sensors that could record numerous quantities. A complete by-wire control system enabled automated driving and the reproducible testing of controllers. Additionally, a rapid control

prototyping (RCP) system allowed control algorithms to be tested directly on the AFM. Furthermore, the AFM was equipped with a custom-made semi-active damper system, which is described in more detail in Section 3.2. Despite the modifications, the vehicle was approved for use on public roads and fulfilled the driving dynamics characteristics of a normal production vehicle.

In Figure 2, the RL-based controller design process for vertical dynamics control applied in this work is depicted. The process applied in this work started by generating measurement data for the whole vehicle. Additional measurement data were collected for the semi-active hydraulic damper and additional components. In a second step, component models and several different QVM structures were generated. The obtained measurement data were then used to parametrize the models through optimization. Since the damper was the central component that converted the control signal into forces, special attention was paid to the measurement and modeling of the damper. Deriving an accurate training model is a crucial task for the RL training process. Since the controller adapts its behavior from interaction with the simulation model, the performance of the resulting controller in the real-world application relies on the accuracy of the model. For the modeling, we used the multi-domain modeling language Modelica [22] together with the integrated development environment Dymola.

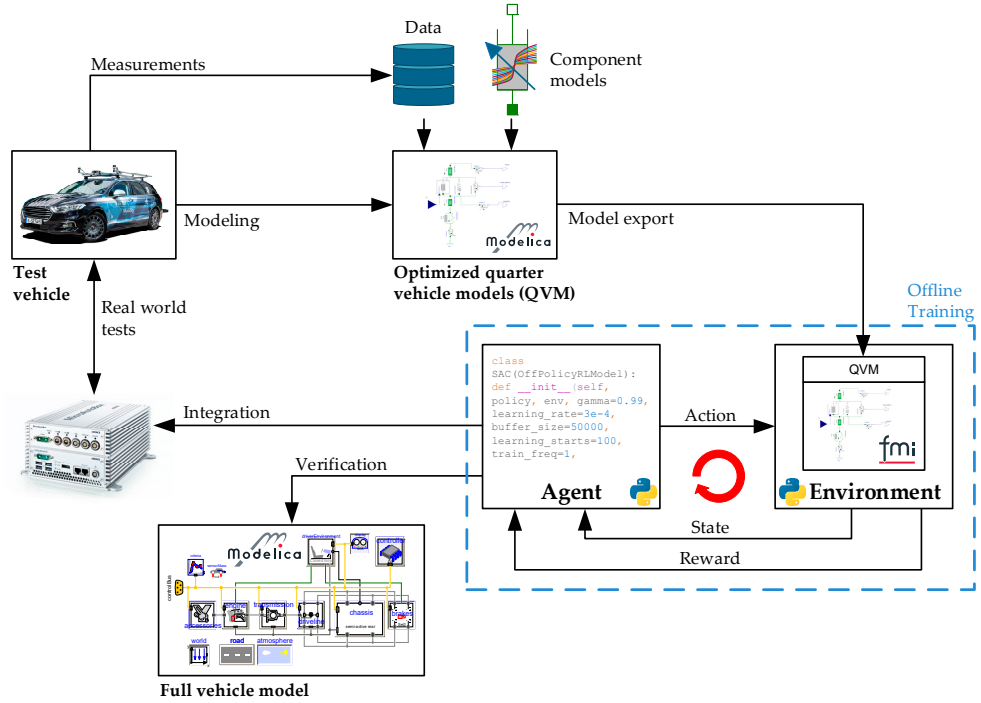


Figure 2. Overview of the whole reinforcement learning toolchain utilized in this work (adapted from [20]).

The most accurate QVM structures were then selected and exported as functional mock-up units (FMU) [23]. The functional mock-up interface (FMI) [23] is a standardized interface that allows the exchange of simulation models between different simulation frameworks as FMU. Since most RL algorithm libraries support a Python interface, the exported model FMU was integrated into a Python-based RL training framework. The whole training process consisting of reward function design, training, and evaluation was then executed inside this Python-based framework. After training a set of RL agents, the best agents, regarding specific metrics, were selected for further evaluation and deployment.

These agents were then exported as C-Code and integrated into both the AFM's RCP and a full-vehicle model for verification. In a last step, the obtained RL agents were tested quantitatively on a four-post test rig and qualitatively in a driving test in the real world. To

ensure a fair comparison in the quantitative tests, we implemented an optimized combined SH/GH reference controller as a benchmark.

In contrast to the model-based controller design process discussed in [24], the RL-based design process allowed the direct integration of the simulation model into the RL training process. Aside from ensuring a reasonably quick simulation process and adhering to the Markov property, no specific requirements were imposed on the training model.

3. Modeling and Parameter Optimization of the Training Model

Training an RL controller in simulation requires an accurate simulation model of the system. Depending on the development stage, it is advantageous to utilize vehicle models of various levels of detail for the training of the controller and its validation. The training of the controller agent starts from scratch as the agent has no a priori knowledge of the system. At this stage, using a QVM plant with fewer states helps to make fast progress and to obtain an agent in an acceptable time.

During the training, the RL algorithm interacted with the simulation model and adapted the policy based on the observed behavior of the simulation model used for the training. A controller trained in simulation will only exhibit the same performance in reality if the training model approximates the real-world system sufficiently accurately. We, therefore, paid particular attention to the modeling and parameterization of the training model.

In Section 3.1, the selection of the training model structure is discussed. Since the semi-active damper was the central element for the controller input, Section 3.2 deals with the modeling of the damper in great detail. The models used for the training of the controller are then presented in Section 3.3.

3.1. Selection of the Training Model Structure

The model used for the training of the RL-based controller should fulfill two objectives: First, it should represent the real-world dynamics of the plant reasonably well. Second, a fast execution time of the model is desirable. This can reduce the time for the controller training significantly. Usually, the training model realizes a tradeoff between these two objectives.

For the vertical dynamics control, several types of vehicle models can be considered for training. The most accurate model types are full-vehicle models. These types of models can usually represent the horizontal vehicle dynamics, the induced pitch, and roll dynamics, as well as the vertical dynamics induced by road excitations. In contrast to that, half-vehicle models cover either pitch or roll dynamics and also represent vertical dynamics induced by road excitation. QVMs are only able to represent road-induced dynamics.

In this work, the aim of the RL controller was to compensate for the vertical road-induced disturbances. Since we assumed that the pitch and roll dynamics were handled by feed-forward control, we selected the QVM as the training model. The QVM model structure was able to represent the desired dynamics and had favorable simulation complexity. In order to obtain a sufficiently accurate simulation model, we used an extended model structure instead of a linear QVM. We considered complex valve behavior, nonlinear kinematics, and elastic bearings in the extended model structure.

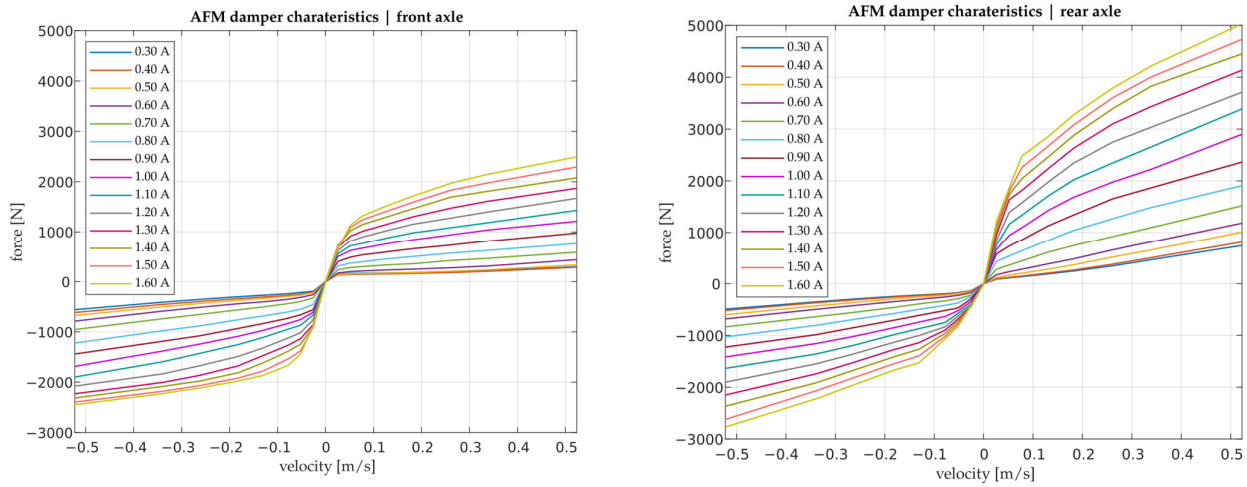
3.2. Damper Identification and Modeling

The test vehicle was equipped with hydraulic triple-tube dampers with an external valve. The damper characteristics could be changed by the positioning of the valve, which was controlled by the magnetic field induced by a current flow through the valve coil. A low-level current controller ensured the tracking of the current setpoint. In this work, we selected the current setpoint of the low-level current controller as the input variable.

Since the front axle of the vehicle was constructed as a McPherson strut and the rear axle was constructed as an integral link suspension, different damper configurations were

used for the front axle and the rear axle. This implied different characteristics at the front and rear axles and different model parametrizations were needed.

In a dedicated damper test rig, several measurements were conducted on one damper from the front axle and one from the rear axle. The damper test rig could apply defined velocity profiles to the damper and measure the induced force by the damper. The following measurements were conducted to obtain the whole damper characteristics and parameters: First of all, a standardized identification process with different constant currents was conducted to obtain the force–velocity map, which is depicted in Figure 3.



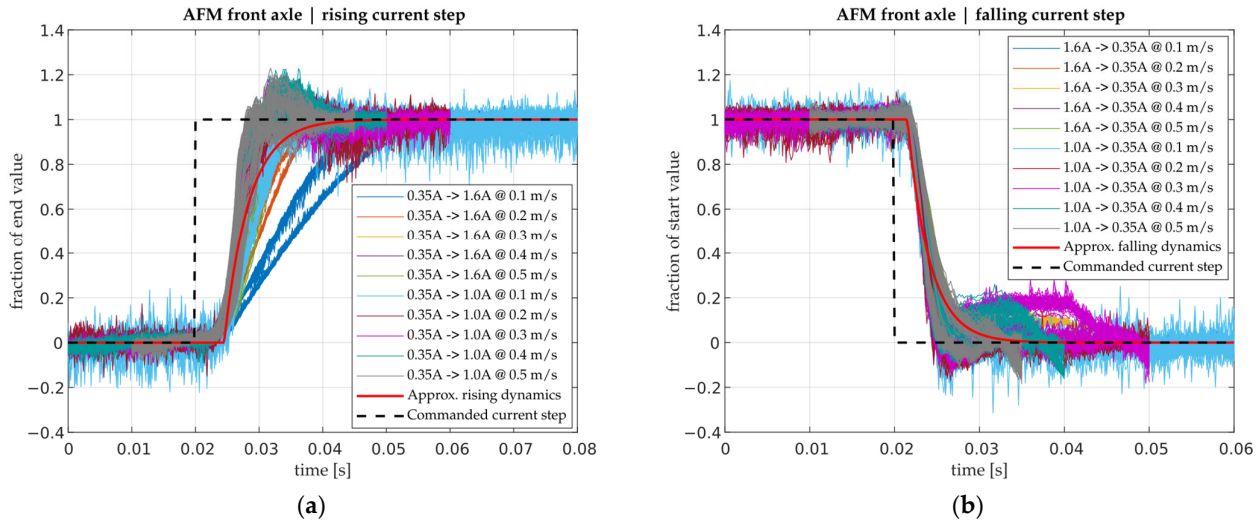
(**a**) (**b**)

Figure 3. Damper force–velocity characteristics for different damper currents for AFM's (**a**) front axle and (**b**) rear axle (compare [20]).

Figure 3a,b depict the velocity–force maps of the front and rear axle dampers, respectively. The damper characteristics of the front and rear dampers differed significantly. The atypical characteristics of the front axle damper resulted from the conversion of the uncontrolled standard damper to a semi-active one. A quasi-static profile with different breakpoints was applied to measure the friction force as well as the air spring stiffness k_i with $i \in \{\mathrm{f}, \mathrm{r}\}$ for the front axle and the rear axle, respectively. This identification resulted in $k_\mathrm{f} = 895\ \mathrm{N/m}$ and $k_\mathrm{r} = 680\ \mathrm{N/m}$.

To identify the input-to-force dynamics of the dampers, different steps in the current setpoints with constant damper velocity were applied and the induced damper forces were recorded. With this test, the input-to-force dynamics could be identified, which included the dynamics of the low-level current controller, the valve dynamics, and additional dynamics within the damper. The results of these tests for the front axle damper are depicted in Figure 4a for a rising current step and in Figure 4b for a falling current step. In this measurement setup, we were only interested in the dynamical behavior of the damper. Thus, the depicted measurement data as well as the input signal were scaled between 0 and 1.

As can be seen in a comparison of Figures 4a and 4b, the dynamical behavior differed between a rising current step and a falling current step. The dynamics were more responsive for a falling current step. However, only small differences in the dynamical behavior between rebound and compression could be determined. Therefore, our model distinguished between positive and negative current steps but did not represent rebound and compression in the input-to-force dynamics.



(**a**) (**b**)

Figure 4. Input-to-force dynamics for different current steps and different damper velocities with (**a**) a rising current step and (**b**) a falling current step. The variables are depicted as fraction of their start or end value over time. In addition to the measurement data, the input signal and the result of the fit are plotted.

Based on this measurement data, the input-to-force dynamics were approximated by a first-order transfer function with an additional delay:

$$G(s) = \frac{1}{1 + T_{i,j}\, s} \cdot e^{-s\, d_{i,j}} \tag{1}$$

with $i \in \{\mathrm{f}, \mathrm{r}\}$ for front axle and rear axle and $j \in \{\mathrm{falling}, \mathrm{rising}\}$.

The transfer function parameters were obtained through optimization using MATLAB's *tfest* function in MATLAB 2022b. We performed a grid search over the delay time $d_{i,j}$ with a spacing of 0.5 ms. Thus, we obtained a time constant $T_{i,j}$ for each grid point $d_{i,j}$ by running the optimization. The parameters $d_{i,j}$ and $T_{i,j}$ that resulted in the lowest cost function values and were considered plausible were selected. The results of the parameter optimization are listed in Table 1.

Table 1. Optimized damper input-to-force transfer function parameters.

Front Axle		**Rear Axle**	
$T_{\mathrm{f,rising}}$	3.915 ms	$T_{\mathrm{r,rising}}$	9.654 ms
$d_{\mathrm{f,rising}}$	4.5 ms	$d_{\mathrm{r,rising}}$	4.0 ms
$T_{\mathrm{f,falling}}$ *	2.615 ms	$T_{\mathrm{r,falling}}$	3.459 ms
$d_{\mathrm{f,falling}}$ *	1.5 ms	$d_{\mathrm{r,falling}}$	1.5 ms

* Second best parameter option selected after visual inspection.

The measured force–velocity map depicted in Figure 3 and the estimated parameters from Table 1 were used to parametrize a slightly modified version of the force map-based damper model presented in [25]. In contrast to [25], we modified the input-to-force dynamics model to match the falling and rising dynamics identified above, whereas [25] differentiated between damper compression and damper rebound movement. Additionally, we included a simple friction model in the damper model, whose friction parameter was identified by the optimization described in the next section.

3.3. Quarter-Vehicle Modeling

The damper model developed in Section 3.2 as well as additional measurements of the full vehicle on the four-post test rig (see Figure 1) built the basis for deriving the training

model. This section describes the derivation of the QVM structures, the optimization-based parametrization, and the evaluation of the training models.

The four-post test rig was used to generate an extensive vehicle motion dataset subject to vertical excitations. In contrast to measuring while driving on a real road, the vehicle can be excited with predefined vertical excitations. Using the four-post test rig also yielded the advantage that additional measurements, such as the dynamic wheel loads, were available. Moreover, the measurements were reproducible and were not affected by external environmental factors, such as the weather.

The vehicle was excited with three different types of sine sweeps to cover a wide range of frequencies: sine sweeps with linear increasing frequency from 0.5 Hz to 5 Hz, from 1 Hz to 20 Hz, and an exponentially increasing frequency sweep from 1 Hz to 30 Hz. Each frequency range was applied with different zero-crossing velocities of 50 mm/s, 100 mm/s, 150 mm/s, 200 mm/s, and 250 mm/s. Additionally, each sweep excitation was performed with different constant damper currents of 0.4 A, 0.6 A, 0.8 A, 1.0A, and 1.6 A. Since the main focus of the training model was to approximate the vehicle's vertical dynamics and not the pitch or roll motion of the vehicle, only synchronous post excitations were selected for the modeling process. We used the following sensor signals for the optimization-based parametrization of the vehicle: position of each post, dynamic wheel load of each wheel, acceleration of each wheel carrier, and acceleration of the chassis at the front right, front left, and rear left sides of the vehicle. The chassis acceleration on the rear right side was not included in the standard sensor setup on the four-post test rig and, thus, not recorded.

Different quarter-vehicle model structures were considered during the modeling process to synthesize an accurate model of the vehicle (see also [20] and Table 2). All QVM structures shared the following elements: a linear spring/damper element as approximation of the tire vertical dynamics, the damper model from Section 3.2, and two mass elements for both the wheel carrier and the body, respectively. Apart from these shared elements, the QVM structures differed in the elements listed in Table 2.

Table 2. Properties of the different QVM model structures considered for comparison.

QVM Structure Name	Nonlinear Spring/Damper Transmission	Topmount Bushing as Linear Spring/Damper Element	Engine Mass with Linear Spring/Damper Bearing
Simple QVM	✗	✗	✗
Transmission QVM	✓	✗	✗
Topmount QVM	✓	✓	✗
Engine QVM	✓	✗	✓
Topmount engine QVM	✓	✓	✓

Both the vehicle spring and damper exhibited slightly nonlinear kinematics with respect to the vertical deflection between the body and wheel. This implies that a vertical deflection between the body and wheel l_{bw} resulted in a nonlinear deflection at the spring l_{s} or damper l_{d}. In the model variants, which included a nonlinear spring or damper transmission, the transmission ratio was defined as

$$i_j = \frac{\partial l_j}{\partial l_{\mathrm{bw}}} \tag{2}$$

with $j \in \{\mathrm{s}, \mathrm{d}\}$. We assumed with respect to l_{bw} linear varying transmission ratio

$$i_j = i_{\mathrm{a},j} + i_{\mathrm{b},j}\, l_{\mathrm{bw}}, \tag{3}$$

which yielded

$$l_j = i_{\mathrm{a},j}\, l_{\mathrm{bw}} + \frac{1}{2} i_{\mathrm{b},j} l_{\mathrm{bw}}^2 + l_{\mathrm{bw},0}. \tag{4}$$

All model variants from Table 2, even the more complex ones, are simplifications of the real-world system. Therefore, selecting the model's parameters as free parameters for the optimization-based parameter estimation offered the possibility to compensate for model mismatches. The optimization algorithm can achieve this by altering the physical parameters of the QVMs to match the measured real-world dynamics. Limiting the parameters within specific appropriate ranges by means of constraints prevented the optimization of non-physical solutions. Since it was not clear which excitations were best suited for the parameter optimization process, different subsets were used.

In the following, we formalized the approach to identify the parameters of different model structures. We assumed to have $n_c > 0$ measured excitation scenarios with input vectors u^i (in our case, post position and damper current) and the corresponding output vectors $y^{\text{ref},i} \in \mathbb{R}^{n_y}$ $(i = 1, \ldots, n_c)$: in our case, the dynamic wheel load, the vertical wheel acceleration, and the vertical vehicle body acceleration. The time duration of each scenario is represented by T_i. For each model type $m = 1, \ldots, n_M$ $(n_M > 0)$, we could simulate model outputs $y^m \in \mathbb{R}^{n_y}$ depending on a vector of model parameters p^m, the input signals u^i, and the time t. We split the components of the parameter vector p^m into two groups, namely, $p^{m,k}$ and $\hat{p}^{m,k}$ $(k = 1, \ldots, n_{S^m})$. Here, n_{S^m} is the positive number of different parameter splits for the model m. The first part of the split represents the free parameters to be identified by optimization; the second part collects the parameters that have directly assigned values from other sources.

For a model m with a parameter split k, we select a non-empty index subset $I \subset I_c := \{1, \ldots, n_c\}$ to define which set of scenarios is used for the identification of parameters $p^{m,k}$. The following optimization problem focuses on the minimization of the time domain error between measured signals and simulated ones:

$$\min_{p^{m,k} \in B^{m,k}} \sum_{i \in I} \sum_{j=1}^{n_y} w_{i,j} \int_0^{T_i} \left(y_j^m \left(p^{m,k}, \hat{p}^{m,k}, u^i(t), t \right) - y_j^{\text{ref},i}(t) \right)^2 dt. \tag{5}$$

The vector of free parameters $p^{m,k}$ is constrained by a box $B^{m,k} := \left\{ p^{m,k} \middle| \underline{p}^{m,k} \leq p^{m,k} \leq \overline{p}^{m,k} \right\}$ for fixed lower limits $\underline{p}^{m,k}$ and upper limits $\overline{p}^{m,k}$. The parameter subset $\hat{p}^{m,k}$ is kept constant. The deviations between model outputs and measured outputs are summed up using positive weights $w_{i,j}$ $(i = 1, \ldots, n_c;\ j = 1, \ldots, n_y)$ to reflect the different scaling of physical variables. The numerical solutions of each optimization problem are named $p_I^{m,k}$.

In our approach, we selected a few parameter splits and some scenario subsets for the five models in Table 2. We selected all the parameters that were not measurable as free parameters within the optimization. In contrast to that, we kept all the parameters constant that were either measurable with a high certainty or resulted from previous investigations, e.g., the damper parameters obtained in Section 3.2. Measured parameters with a low certainty were kept constant in one of the first splits and selected as free parameters for the optimization in other splits.

Since a large scenario subset means simulating the model many times with different input excitations in each evaluation of the optimization cost function, it was important to select only the most relevant excitations for the parameter identification. In our application, the sine sweeps showed to have the most information included. Thus, these excitations were preferred to define the optimization problems. We used the Optimization Library [26] and additionally extensive scripting in Dymola, an integrated development environment for Modelica, to set up and solve the problems automatically. The computations were executed on an in-house computing cluster using the parallelization features of the Optimization Library.

After solving the optimization problems, we finally obtained a list of identified parameter sets $(p_I^{m,k}, \hat{p}^{m,k})$ with a split k for the models m and excitation scenario subsets I. To evaluate each of the solutions, we computed separate error metrics for each of the components y_j^m for the whole scenario set I_c. This meant simulating the models parametrized by

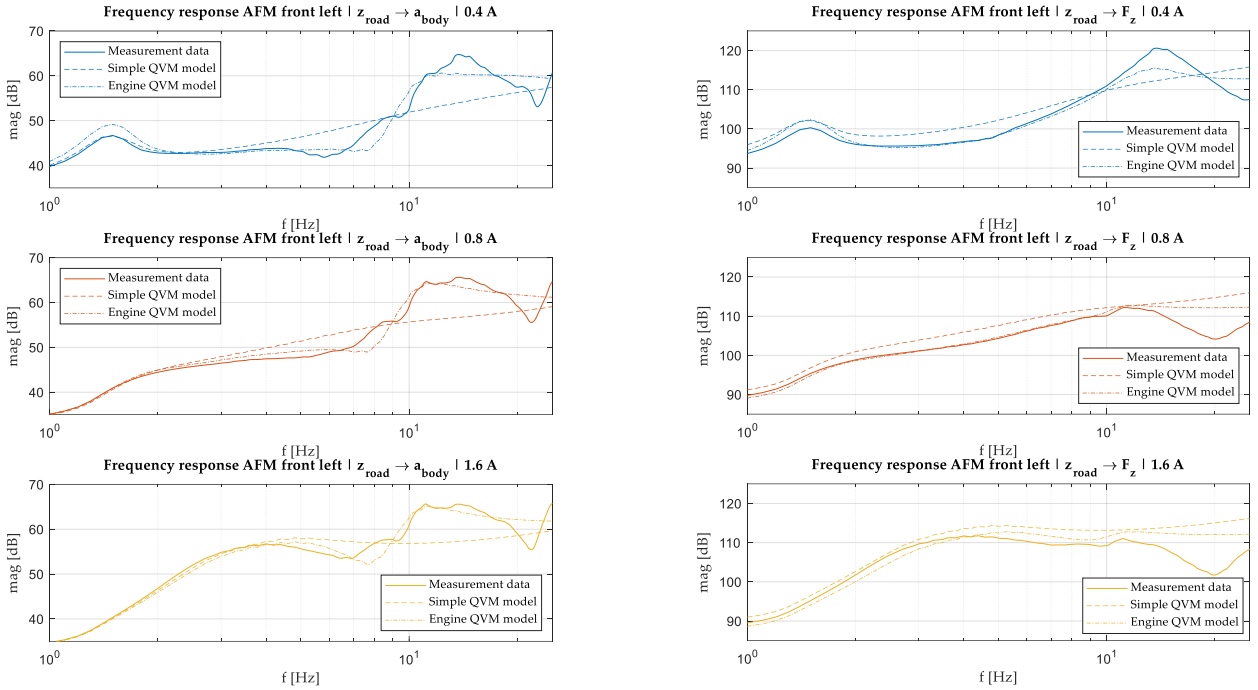
the identified parameters with all the test scenarios. Based on qualitative and quantitative comparisons of these results, we selected the best model structure with its identified parameters (cf. Table A1 in the Appendix A). Since the body acceleration was only measured at the front left (FL), front right (FR), and the rear left (RL) but not at the rear right (RR) corner of the vehicle, we performed the optimizations only for the three vehicle corners where the full measurement data were available.

The optimal model structures revealed by the optimization were the "Engine QVM" variant for FL and FR and the "Topmount QVM" for the RL side of the vehicle (cf. Table 2). This aligned very well with the structure of the vehicle: The AFM was equipped with uniball topmounts, which acted as an almost rigid link between the damper and the body. Additionally, the vehicle's engine was located in the front of the vehicle. This made the model variant "Engine QVM", which included an engine mass but no elastic topmount, a plausible choice for the front axle. In contrast to that, a standard elastic topmount was installed on the rear axle. Additionally, no big oscillating mass, like the engine in the front, was present in the rear. Therefore, the model variant "Topmount QVM" seemed an adequate choice for the rear left side of the vehicle.

To validate the obtained parametrized QVMs, we excited the optimized models with one selected road profile (exponential sweep, 1–30 Hz, 100 mm/s zero-crossing velocity). This excitation was also applied during the measurements of the vehicle performed on the four-post test rig. In addition, a standard linear two-mass quarter-vehicle model equipped with the nonlinear damper model described in Section 3.2 with an optimized parameter set was also simulated (compare simple QVM from Table 2). These simulations were repeated for a minimal damper current of 0.4 A, a medium damper current of 0.8 A, and a maximum damper current of 1.6 A. MATLAB's *tfestimate* function was used to calculate the frequency response from road displacement to the acceleration of the vehicle body depicted in Figure 5 and from the road displacement to the dynamic wheel load shown in Figure 6. Even though transfer functions were only well defined for pure linear systems, we assumed only limited errors because of the close-to-linear system behavior for constant damper currents. For constant damper currents, the input-to-force dynamics described in Section 3.2 did not come into effect. Nevertheless, the conducted investigation was only qualitative and, in contrast to pure linear systems, may have resulted in slightly different results for other excitations. It has to be noted that Figures 5 and 6 only show one excitation, namely, the exponential sweep, 1–30 Hz with a 100 mm/s zero-crossing velocity.

Figure 5a shows that the body acceleration on the front left vehicle side had one peak at around 1.5 Hz for the 0.4 A damper current case, which represents the body mass eigenfrequency. Increasing the damper current suppressed this first peak. A second broad peak appeared approximately between 10 and 16 Hz. This peak was only slightly influenced by the different damper currents. Inspecting the second peak in Figure 5b at around 14 Hz in the transfer function from the road profile to wheel load suggests that one part of the broad peak in Figure 5a resulted from the wheel eigenfrequency.

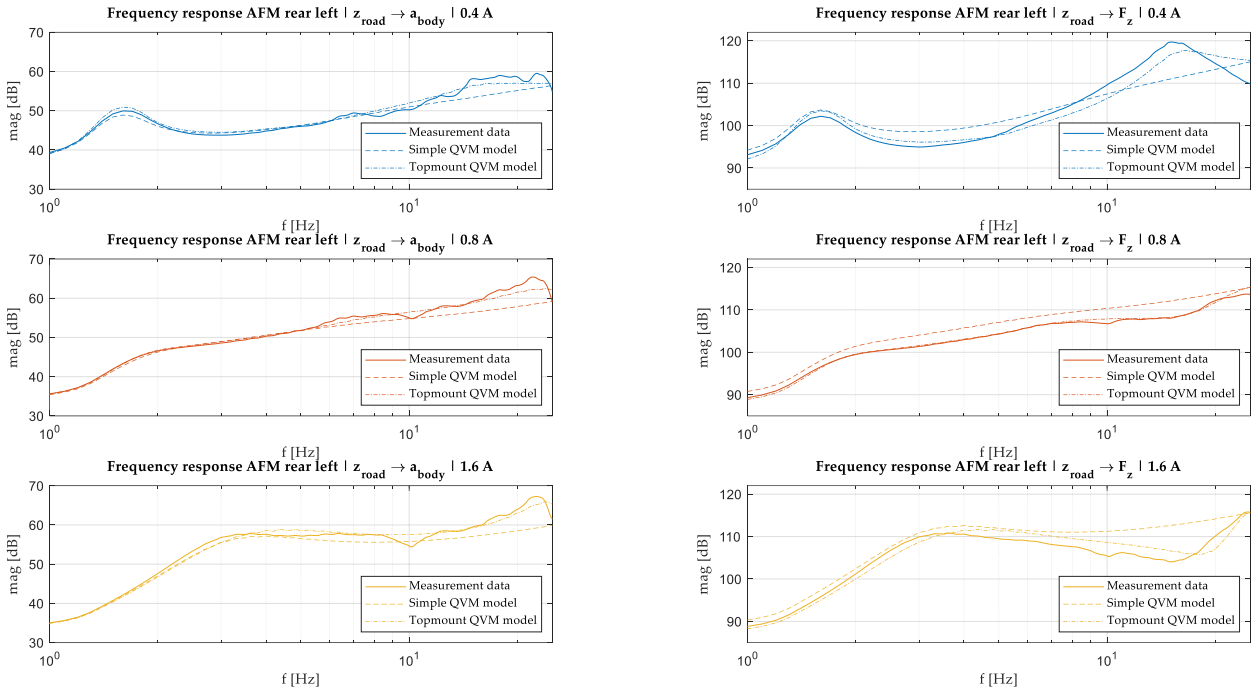
Additionally, the inspection of the simulation data showed that, at around 10 Hz, the body and engine mass oscillated in an anti-phased way. Hence, introducing the engine mass in the QVM enhanced the capability of the model to approximate the magnitude of the body acceleration, especially at higher frequencies. Even though the simple QVM approximated the height of the first peak for the 0.4 A damper current better, the engine QVM also met the eigenfrequency of the body mass. Looking at the 0.8 A and 1.6 A cases from Figure 5a exhibits a similar approximation capability for both the simple and the engine QVM model structure in the low-frequency range. For frequencies above 5 Hz, the engine QVM approximated the magnitude much better. Despite minor exceptions, Figure 5b shows that the engine QVM approximated the wheel load more accurately than the simple QVM.



(**a**) (**b**)

Figure 5. Comparison of AFM's frequency response from (**a**) road displacement z_{road} to the acceleration of the vehicle body a_{body} for different constant damper currents and (**b**) from road displacement z_{road} to the dynamic wheel load F_z for different constant damper currents for the front left side of the vehicle. Each subplot visualizes the measurement data, the data obtained from an optimized simple QVM model, as well as the resulting data obtained from the optimized best QVM model structure.

Figure 6a reveals that the broad peak between 10 and 16 Hz, which can be observed in Figure 5a, was less pronounced on the rear left side of the vehicle. Since the vehicle's engine was in the front, one cause for the broad peak was not present at the rear of the vehicle. Similar to the front left side, the eigenfrequency of the body mass could be observed as a peak in the magnitude of the body acceleration for the 0.4 A damper current in Figure 6a at around 1.6 Hz. The wheel eigenfrequency of the 0.4 A damper current in Figure 6b can be observed at around 15 Hz. Since the simple QVM model and the topmount QVM only differed in the additional topmount elasticity, the transfer functions of both fitted models were pretty close. Nevertheless, the more complex model structure can approximate the magnitude of the body mass acceleration and the wheel load better than the simple QVM. The advantage is evident for higher frequencies and the magnitude of wheel load depicted in Figure 6b.



(**a**) (**b**)

Figure 6. Comparison of AFM's frequency response from (**a**) road displacement z_{road} to the acceleration of the vehicle body a_{body} for different constant damper currents and (**b**) from road displacement z_{road} to the dynamic wheel load F_z for different constant damper currents for the rear left side of the vehicle. Each subplot visualizes the measurement data, the data obtained from an optimized simple QVM model as well as the resulting data obtained from the optimized best QVM model structure.

4. Training the Controller

Two different kinds of excitations stimulate the vertical dynamics of a vehicle: The vertical excitation induced by the road profile and the coupled pitch and roll excitation caused by the horizontal vehicle motion. The latter is mainly induced by steering, acceleration, and braking inputs, which all can be measured. This allows to predict the horizontal vehicle motion reasonably well. Thus, this impact on the vertical dynamics can also be estimated by utilizing a simple pitch and roll model of the vehicle together with some auxiliary measurements. Consequently, a prediction-based feed-forward controller can be applied to compensate for undesired pitch and roll motion.

In contrast, it is less reasonable to apply a model-based feed-forward control for road-induced disturbances. Since it is inconvenient to measure the uncertain road excitation early and accurately enough, it is challenging to obtain an accurate prediction. Considering that the pitch and roll dynamics of the vehicle can be handled by feed-forward control as described above, we applied the RL-based control solely to the vertical dynamics part excited by the road profile.

As described in Section 1, model-based or model-derived vertical dynamics controllers are often designed based on different assumptions or modeling simplifications. Additionally, it is non-trivial to tune the controller parameters to account for different road excitations and vehicle speeds. RL algorithms optimize a control law based on iterative interaction with the to-be-controlled system or a simulation model of the system. By utilizing sophisticated training models and training on a wide range of road excitations, the RL-based training process could adopt a more performant control law operating on different road excitations. Because of the generic characteristics of RL methods, the toolchain can be automatized for different vehicle types with little additional effort.

Training in simulation is fast, scalable, and safe. The training of the controller in the real world would be performed on the four-post test rig, which is necessary to excite the real vehicle with defined excitations and measure the dynamic wheel load. The application of RL to a new problem usually includes a trial-and-error process and can be time consuming. Additionally, equipping the vehicle on the test rig poses a significant overhead. Therefore, our approach relies on a simulation-based training strategy utilizing the advanced QVMs developed in Section 3.

4.1. The Reinforcement Learning Setting

The basic setting of RL is depicted in Figure 7. An RL agent learns a desired behavior out of interaction with an environment. The agent can manipulate the environment by applying an action a and, in return, obtains an observation vector o and a scalar reward r. The goal of RL algorithms is to adapt a policy π such that the expected discounted return is maximized.

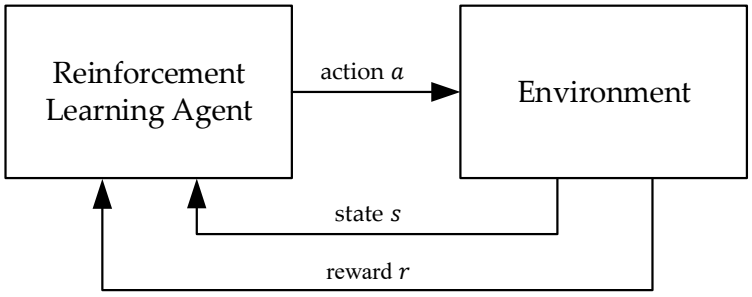


Figure 7. Basic RL agent environment setting (adapted from [27]).

As opposed to the standard setting in control theory, in the basic RL setting, the agent does not need any prior knowledge about the environment and a deterministic behavior of the environment is not required. The non-deterministic approach of RL makes these algorithms especially appealing for vertical dynamics control, where the road excitation is assumed to be stochastic.

Since the RL algorithm adapts the policy solely through interaction and no additional information about the environment is necessary, such algorithms are very generic and can solve a wide range of sequential decision-making problems. However, the theory on RL assumes that the environment behaves as a Markov Decision Process (MDP) [27]. Even though the time delay in the input-to-force dynamics described in Section 3.2 violates this assumption, the algorithms applied in this work are robust enough to train a performant policy.

In deep RL, artificial neural networks are used to approximate several functions, e.g., the so-called *action-value function*, *state-value function*, and the policy. In the last decade, several powerful deep RL algorithms were proposed, such as the deep Q-network (DQN) [28], proximal policy optimization (PPO) [29], deep deterministic policy gradient (DDPG) [30], and soft actor–critic (SAC) [31]. Over the years, different well-maintained open source implementations of these algorithms have been created and benchmarked. The most widely used libraries include the RLLib [32] and stable-baselines3 [33].

4.2. Application to the Vertical Dynamics Problem

When applying RL to a specific problem, various design choices can be made. The major degrees of freedom are the selection of the RL algorithm, its corresponding hyperparameters, the assembly of the observation vector o, and the design of the reward function r. Additionally, the implementation details of the environment itself can affect the training. In this section, we describe the application of RL to the vertical dynamics problem.

4.2.1. The Training Setup

The QVM models developed and analyzed in Section 3 serve as a basis for the simulation-based training environment. We utilized Modelica/Dymola to develop and optimize the models, but most RL libraries interact with the environment via the OpenAI gym [34] interface, which is implemented in Python. It has to be mentioned that the *gym* interface has now further evolved to *gymnasium* [35], but the main features remain unchanged or are only altered slightly. Since the training can most easily be conducted in a Python environment, the Modelica models were integrated as an FMU into Python. In this work, we used an improved version of the FMU-Python toolchain previously applied in [36,37] to accomplish this integration. In addition, we utilized the RL baselines3 Zoo [38] to facilitate the saving of the agent and to track the hyperparameters used during each training.

The aim of the training was to train *one* agent to handle road-induced disturbances, which would then be applied to *each* vehicle corner. Previous investigations showed that training separate agents specifically for each vehicle corner induced undesired pitch and roll oscillations for symmetric pure vertical excitations on the four-post test rig. Even though all posts performed the exact same motion in these tests, minor differences in the policy resulted in different damper forces at each wheel and, therefore, undesired pitch and roll motions occurred. In order to enable the agent to perform well on the different vehicle sides, we trained the controller sequentially on the three QVMs described in Section 3. In addition to changing the vehicle model during training, we exposed the agent to different road excitations. Based on previous experience and to cover a wide range of excitation profiles, we trained on sweep, bump, and wave excitations. The sweep excitations were chosen for the training to ensure that the eigenfrequencies of the wheel and the chassis were adequately excited during training. The bump and wave excitations posed challenging events, which frequently occur in the real world and also excite different modes of the vehicle.

In this work, we used a SAC implementation from the stable-baselines3 library [33]. The SAC is an actor–critic off-policy RL algorithm that augments the standard expected sum of rewards optimization objective with an information theoretical entropy term of the policy [31]. This entropy term favors stochastic policy behavior and is weighted against the expected sum of rewards objective by a so-called temperature parameter. Thus, the exploration vs. exploitation dilemma of RL is directly addressed in the objective function. The SAC is designed for continuous action spaces and has successfully been applied to a wide range of control problems, e.g., [37,39]. Additionally, a comparison of different DRL algorithms on the vertical dynamics control problem conducted in [18] showed that the SAC algorithm performed best in their setting. Therefore, we chose this algorithm for the vertical dynamics application.

The environment was implemented with a sample time of 1 ms and trained with various hyperparameters. The training with different hyperparameter sets and reward function designs was performed parallelly on a DLR in-house computing cluster, where one training with 70 million timesteps took around 24 h. Since only small neural network sizes were used within the RL algorithm, the training was solely performed on the CPU, as the training on the GPU did not increase training speed. During each training, the agent was saved in regular time intervals. This way, we were able to restore performant agents, even in cases where catastrophic forgetting [40] occurred. As described in Section 4.2.4, each stored agent was benchmarked after the training and the most performant agents were selected for evaluation in the real-world experiment. The hyperparameters used to train the most performant agent are summarized in Appendix A.3 in Table A3.

4.2.2. Environment Interface

In RL, the agent adapts its policy based on observations and rewards returned from the environment. The theory behind RL often assumes that the whole state of the environment can be observed by the agent. However, in real-world application, it is not always feasible to measure or estimate the whole state of the system and, therefore, it is not always possible

to provide the full state measurement to the agent, resulting in a Partially Observable Markov Decision Process (POMDP). Although many of the RL algorithms are developed based on the assumption that the whole state is available, in many applications RL delivers satisfying results even though not the whole state is available to the agent. In addition to the availability of measurement signals, the assembly of the observation vector can guide the training. Since it is not always obvious which signals provide valuable information to the agent, we performed various trainings with different observation sets.

The model structure "Engine QVM" described in Section 3.3 and utilized to model the front left and front right sides of the vehicle was essentially a three-mass oscillator which yielded six states, two for each mass. An additional state resulted from the approximation of the input-to-force dynamics by a first-order system, as described in Section 3.2. In contrast, the model structure "Topmount QVM" used to approximate the rear left side of the vehicle did not include the engine mass and, in consequence, also the two states used to describe the engine motion. However, introducing the spring/damper element to model the topmount again resulted in additional states. These two states, together with the additional state for the first-order input-to-force approximation, added up to seven states for all the model variants.

Nevertheless, in real-world applications, it is infeasible to measure all seven states on each corner of the vehicle. Even though the AFM test vehicle was equipped with a multitude of sensors [7], this is usually not the case for production vehicles. Therefore, we only utilized the following sensor signals for this work:

- acceleration sensors at all four-wheel carriers;
- acceleration sensors at the front left, front right, and rear left chassis;
- displacement sensors between the chassis and wheel carrier at all four wheels;
- current sensor for each damper.

These sensor signals are used to compute the vertical chassis velocity $v_{\mathrm{c},j}$, the vertical wheel velocity $v_{\mathrm{w},j}$, and the damper velocity $v_{\mathrm{d},j}$ for each side of the vehicle, i.e., $j \in \{\mathrm{fl},\ \mathrm{fr},\ \mathrm{rl},\ \mathrm{rr}\}$.

In the end, the agent that performed best in the real-world tests included the following quantities in the observation vector

$$o_j = \left[v_{\mathrm{c},j}, v_{\mathrm{w},j}, v_{\mathrm{d},j}, i_j\right], \tag{6}$$

with i_j representing the measured actual damper current and $j \in \{\mathrm{fl}, \mathrm{fr}, \mathrm{rl}, \mathrm{rr}\}$. This selection seems reasonable, given that the ride comfort is calculated based on the vertical chassis acceleration $a_{\mathrm{c},j}$ and the road-holding is closely linked to the vertical wheel acceleration $a_{\mathrm{w},j}$. The damper velocity $v_{\mathrm{d},j}$ is an important value because, in combination with i_j, it determines the damper force, as depicted in Figure 3. It should be noted that the quantities $v_{\mathrm{c},j}$, $v_{\mathrm{w},j}$, and $v_{\mathrm{d},j}$ were used as inputs for an SH/GH controller implementation.

In contrast to choosing the quantities for the observation vector, the assembly of the action vector was less complex. The force induced by the damper was controlled by the position of an electromagnetic valve, which itself was controlled by the current through its coil. On the damper control unit (DCU), the current flow could be controlled by adjusting the duty cycle of a pulse-width modulation (PWM). One option for the selection of the agent action would be to directly control the duty cycle of the PWM. However, this choice carries the risk of the current in the coil becoming too high. Since a proven current controller was available on the DCU, we selected the current setpoint as the action a for the RL agent.

4.2.3. Reward Function Design

The design of the reward function is a crucial degree of freedom in the application of RL. In general, RL algorithms are designed to maximize the expected discounted sum of future rewards. This implies that the reward should quantify desirable behavior, where favorable behavior leads to a higher reward compared to a smaller reward for undesirable behavior.

The main objective in vehicle vertical dynamics control is usually to optimize ride comfort as well as road-holding [41]. However, these two objectives are, to some extent, opposing each other. This means that at a certain point enhancing one metric yields a corresponding deterioration in the other. Previous experience has shown that it is much harder for the RL algorithm to optimize for comfort than for road-holding. Therefore, we designed the reward function to include ride comfort but neglect road-holding in the reward function to simplify the problem for the RL algorithm. Nevertheless, we integrated a safety module into the whole vertical dynamics control system to ensure driving safety during real-world road tests.

To enable the optimization of these objectives, a numerical quantification of the objective was necessary. In general, it is not trivial to measure ride comfort, since it describes a subjective perception. Different metrics to quantify ride comfort are compared in [42]. In this work, we assumed to enhance ride comfort by minimizing the vertical chassis motion. Even though ride comfort is usually calculated from chassis vertical accelerations, we found that including the chassis vertical velocity in the reward function leads to better results compared to using the chassis acceleration.

In previous experiments, we noticed several undesired agent behaviors: First, the agent actions tended to be very jittery, meaning that the action signal was very noisy from one timestep to another. As a second issue, we observed that the policy induced high current steps during high damper velocities. Given the damper characteristics depicted in Figure 3, this led to a sudden change in the damper force, which resulted in undesirable noises and might have harmed the damper due to increased mechanical stress. Additionally, it is desirable to keep a low damping characteristic as a default. This way, the damper can absorb high-frequency road disturbances, e.g., those induced by driving over a cobblestone road.

We designed the reward function such that (1) chassis motion was minimized, (2) high jumps in the damper force were avoided, (3) the action signal was smooth, and (4) the default damper current was low. To achieve this, we constructed the reward function out of different terms, each accounting for a different objective weighted by an accompanying weighting factor. The used terms and weights are listed in Table 3.

Table 3. Weights and terms used in the reward function.

Reward Weights		Reward Terms	
-	-	r_{fj}	Force jump reward term
k_{cm}	Chassis motion weight	r_{cm}	Chassis motion reward term
$k_{\Delta \mathrm{u}}$	Control signal jump weight	$r_{\Delta \mathrm{u}}$	Control signal jump reward term
k_{a}	Control signal weight	r_{a}	Control signal reward term

The individual terms are assembled as shown in the following equation:

$$r = r_{\mathrm{fj}}(v_{\mathrm{d}}, \Delta u) \cdot (k_{\mathrm{cm}} \cdot r_{\mathrm{cm}}(v_{\mathrm{c}}) + k_{\Delta \mathrm{u}} \cdot r_{\Delta \mathrm{u}}(\Delta u) + k_{\mathrm{a}} \cdot r_{\mathrm{a}}(a)) \tag{7}$$

Herein, the damper velocity is denoted as v_{d}, the chassis velocity as v_{c}, and the action as a. Δu describes the demanded current jump Δu, defined as

$$\Delta u = a - i_{\mathrm{d}}. \tag{8}$$

Herein, the current jump Δu represents the difference between the commanded current a and the measured current i_{d}.

The reward function r consists of two factors: The force jump reward term r_{fj} is multiplied with the weighted sum of all other reward terms listed in Table 3. Herein, the force jump reward term is defined as

$$r_{\mathrm{fj}}(v_{\mathrm{d}}, \Delta u) = \begin{cases} 1.0, & (v_{\mathrm{d}} < |\theta_{v_{\mathrm{d}}}|) \text{ or } (\Delta u < |\theta_{\Delta u}|) \\ clip\Big(1.0 - k_{\mathrm{fj}}(|v_{\mathrm{d}}| - \theta_{v_{\mathrm{d}}}) \cdot (|\Delta u| - \theta_{\Delta u}), 0.0, \quad 1.0\Big), & \text{else} \end{cases} \tag{9}$$

with the parameters k_{fj}, $\theta_{v_{\mathrm{d}}}$, and $\theta_{\Delta u}$ and the function clip$(\ldots, 0.0, 1.0)$, which limits the value range to $[0.0, 1.0]$. The force jump reward term r_{fj} takes the damper velocity v_{d} and the demanded current jump Δu as input. It has to be noted that r_{fj} can take values between 0 and 1. This means that the whole reward in Equation (7) is 0 if $r_{\mathrm{fj}} = 0$. If $r_{\mathrm{fj}} = 1$, then the other terms of the reward function come into effect. This way, the force jump reward term is prioritized over the other terms, since it is able to scale the other terms. This kind of hierarchical reward function design was adopted from our previous works [36,37]. Even though the main objective was to minimize the chassis motion, it was necessary to give the force jump reward term such a prominent position within the reward function. Adding the force jump reward term to the weighted sum in Equation (7) did not have a significant effect.

The shape of the force jump reward term is depicted in Figure 8. It can be seen that the function value becomes 0 if a large current jump is commanded at high damper speeds. From the damper characteristics in Figure 3, it is evident that a big jump in the current yields a high jump in the damper force if applied at high damper velocities. We encouraged the agent to avoid high force jumps by shaping the force jump reward term as proposed in Equation (9) and depicted in Figure 8.

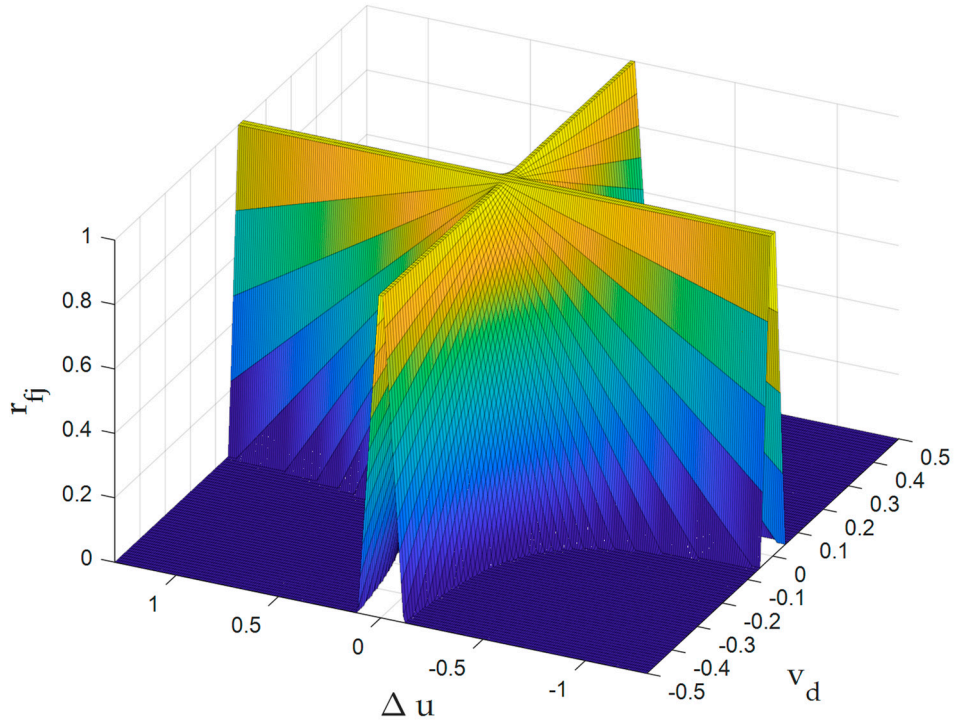


Figure 8. Illustration of the force jump reward term $r_{\mathrm{fj}}(v_{\mathrm{d}}, \Delta u)$ for parameters $k_{\mathrm{fj}} = 20$, $\theta_{v_{\mathrm{d}}} = 0.01$, and $\theta_{\Delta u} = 0.01$. The colors emphasize the values of r_{fj} beginning from 0.0 in dark blue to 1.0 in yellow.

We make use of the Gaussian-like function $g_\sigma(x)$ in the following, which we define as

$$g_\sigma(x) = e^{\frac{-x^2}{2\sigma}} \tag{10}$$

As observed in our previous works on applying RL, we found this function to be helpful within the design of the reward function. The function has two helpful properties, which we make use of: First of all, its values are limited: $0 < g_\sigma(x) \leq 1$. This way the contribution of each term to the reward function is normalized and the different terms can be weighted by additional multipliers. The second advantage is that in addition to $x = 0$, it has a non-zero gradient, which can guide the training.

The chassis motion term r_{cm} is designed to reward a minimized vertical chassis motion. Therefore, we designed the chassis motion term as

$$r_{\mathrm{cm}}(v_{\mathrm{c}}) = 0.8 \cdot g_{\sigma_{v_{c1}}}(v_{\mathrm{c}}) + 0.2 \cdot g_{\sigma_{v_{c2}}}(v_{\mathrm{c}}) \tag{11}$$

with the two chassis motion reward term parameters $\sigma_{v_{c1}}$ and $\sigma_{v_{c2}}$. It should be noted that the two terms in Equation (11) are weighted such that $r_{\mathrm{cm}}(v_{\mathrm{c}})$ is limited between 0 and 1. The combination of the two Gaussian-like functions in that way essentially broadens the peak compared to a single Gaussian-like function. In line with the results in [9], we found that designing the comfort reward term based on the chassis vertical velocity yielded better results in contrast to using the chassis vertical acceleration.

The next term in the reward function, i.e., the control signal jump reward term $r_{\Delta \mathrm{u}}(\Delta u)$, encourages a smooth action signal. To achieve this, we rewarded a small deviation between the commanded current setpoint and measured current via the Gaussian-like function defined in Equation (10).

$$r_{\Delta \mathrm{u}}(\Delta u) = g_{\sigma_{\Delta u}}(\Delta u), \tag{12}$$

with the parameter $\sigma_{\Delta u}$.

The last term in the reward function, i.e., the control signal reward term r_{a}, promotes a low default damping, which is desirable as discussed at the beginning of this section, i.e., Section 4.2.3. To implement this into the reward function, we used a simple linear term

$$r_{\mathrm{a}}(a) = m_a \, a + b_a, \tag{13}$$

with the parameters m_a and b_a.

The parameters that led to the most performant agent are listed in Table A4 in Appendix A.4.

4.2.4. Agent Performance Assessment

Most of the time, training a performant RL agent is an iterative process. This process consists of training an agent, evaluating the trained agent, changing the training setup, and then starting a new training. Different metrics are of interest during the evaluation of the agents. Since the reward compresses multiple objectives into one scalar measure, this reward may not be sufficient to assess the multiple objectives.

Therefore, we implemented a performant multi-objective agent assessment pipeline to evaluate the trained agents. This pipeline was composed out of different steps that were performed for each trained agent. In the first step, every trained agent was evaluated on predefined road excitations and all signals of interest were temporarily stored. Then, predefined metrics were calculated based on these signals and the resulting metrics themselves were also stored in a database. In addition to the metrics, all hyperparameters and other information about the environment and the training were stored. Keeping track of all the information allowed an in-depth analysis of the promising hyperparameters, reward functions, or environment implementations. Additionally, this pipeline enabled traceability and reproducible training results.

To rate the performance of the learned controller, we needed to relate the calculated metrics to a state-of-the-art controller. To achieve a fair comparison, we optimized an SH and a GH controller based on the optimal QVMs, which were also used during training. After that, the metrics of interest were calculated for the optimized SH controller, the optimized GH controller, and all trained RL-based controllers. To ensure a balanced assessment of the controllers, we evaluated each controller version on a set of different excitations including sweeps, wave shapes, and bump-like excitations. Following the calculation, we compared each trained agent to the better one from SH and GH and took the mean over all excitations. This resulted in a set of metrics that allowed us to compare the trained controllers to state-of-the art controllers.

In the controller assessment, we were especially interested in the comfort criterion, the road-holding criterion, as well as the smoothness of the actions. During the assessment, we applied a simplified version of the comfort criterion J_c from [43], i.e.,

$$J_c = \sqrt{\frac{1}{N}\sum_{k=1}^{N} a_{c,k}^2}, \tag{14}$$

with the vertical chassis acceleration a_c. To assess the road-holding $J_{rh,}$ we calculated the root mean square (rms) of the dynamic wheel load $F_{z,dyn}$

$$J_{rh} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} F_{z,dyn,k}^2}. \tag{15}$$

Finally, we measured the smoothness of the action signal J_s via a mean of the absolute change from one timestep $(k-1)$ to the next timestep k:

$$J_s = \frac{1}{N}\sum_{k=1}^{N} |a_k - a_{k-1}|. \tag{16}$$

In order to select agents for the evaluation on the real vehicle, we filtered all agents in which the comfort criterion was a maximum of 5% worse compared to the reference controller. The remaining agents were then sorted for smoothness in the action signal. Finally, we chose the smoothest four agents for evaluation on the real vehicle.

5. Control System Verification

After the training and selection of the RL agent described in Section 4, the obtained agent then needed to be deployed on the real vehicle. Nevertheless, an in-depth verification of the trained RL agent and the whole control system was favorable to support a smooth transition to the real vehicle. The aim of the verification process comprised three objectives: (1) check the time domain behavior of the ANN-based RL agent, (2) evaluate the interaction of the agent with the remaining part of the control system, and (3) assess the agent's behavior on a full vehicle.

The policy obtained during the RL training was, in our case, an ANN-based mapping from the observation space to the input space. In contrast to analytical control laws, it was hard to derive a priori properties for the trained ANN-based agent. Even though the selected agent performed well on the scalar metrics defined within the agent performance assessment described in Section 4.2.4, it was still possible that the agent exhibited undesired behavior in the time domain. If the assessment metrics defined in Section 4.2.4. did not cover all possible undesired behaviors, the agent might have learned a policy that was able to maximize the reward and perform well across all performance assessment metrics but still show undesirable behavior. Often, such undesirable policy patterns can be identified by evaluating the time domain signals generated by the agent.

The second reason to validate the agent within a full-vehicle model (FVM) setting is that the agent will be deployed in an entire control system, in which the RL policy is only one component. In addition to the agent, the control system was comprised of different modules, of which some will be described in Section 5.2. During the verification process, the whole control system was integrated into the FVM simulation. Thus, the interaction between different subcomponents as well as the implementation of their interfaces could be checked.

The last cause to perform the evaluation simulation with the whole control system integrated into an FVM is the approach from Section 4 to train the controller on QVM models. To check the behavior of the agent applied to the FVM, the following controller verification simulations were conducted.

It has to be noted that we did not conduct a quantitative performance analysis on the FVM. Since we were interested in the performance of the trained controller on the real vehicle, the verification simulations were just a preliminary investigation to enable real-world tests. Even though the FVM described in Section 5.1 was developed to fit the real-world dynamics as well as possible, each model was a just a replica of the real world and might not have been able to fully capture the underlying real-world dynamics. Therefore, we will demonstrate the performance analysis on the data obtained from the real-world experiments described in Section 6.

The remainder of this section is organized as follows: The full-vehicle model of the AFM demonstrator is presented in Section 5.1. The entire vertical dynamics control system composed of different submodules is then discussed in Section 5.2. The design and optimization of the benchmark controller is briefly discussed in Section 5.3. The integration of the developed control system into the simulation environment, which was later executed on the RCP platform in the vehicle, is described in Section 5.4. Verification simulations conclude this section in Section 5.5.

5.1. Full-Vehicle Model

The learned control agent was embedded into a multi-body-vehicle model for verification. This integration enabled extensive controller testing within a nonlinear full-vehicle model for both standard driving maneuvers and different road profiles, which can be either measured or artificially created.

For these purposes, the vehicle model was implemented in the object-oriented modeling language Modelica as a multi-body model, which enabled full spatial motion. Its chassis sub-model comprised advanced models of the front McPherson and rear integral-link suspensions. Further, it contained stabilizers, a steering assembly, and wheels that utilized the semi-physical tire model TMeasy [44]. A driver model and the vehicle environment, including the road, were provided to simulate driving maneuvers. The learned control agent was integrated as an FMU. For further details of the full-vehicle model of the AFM, see [7,20,24].

In addition to the benefits of multi-physical modeling, the implementation in Modelica offered the ability to implement the semi-active damper as a one-dimensional translational model and to use its parameterized front and rear variants in both the QVM and the full-vehicle model. The one-dimensional translational damper model, described in Section 3.2, was further extended with additional accessories, including bushings, bump, and rebound stops, depending on the desired level of detail. The control and necessary measured signals of the semi-active damper's model were all readily accessible via a virtual control bus.

5.2. Vertical Dynamics Control System

As described in Section 4, the RL controller was mainly designed to compensate the excitations induced by the road itself. In addition, there were also parts of vertical dynamics effects in real driving operation that were caused by driver inputs. These were, for example, the chassis rolling during cornering or the pitching during acceleration or braking. In order to take these excitations into account in the vertical dynamics control concept, the overall control system consisted of further modules in addition to the RL controller:

- Prediction module for vehicle body accelerations:

Lateral and longitudinal accelerations were calculated based on the brake pedal position and the steering wheel angle. A stationary single-track model was applied to neglect the system dynamics and, thus, enable time prediction.

- Feed-forward control:

An inversion-based approach was used to calculate the required damper forces, which would compensate for the resulting roll and pitch angle based on the predicted body accelerations.

Even though the RL-based controller was tested in simulation, a safe operation of a learning-based controller cannot be ensured a priori. To intercept unsafe driving conditions, we added a safety module for all driving tests conducted on real roads:

- Safety module:

As soon as critical roll and pitch rates of the vehicle body were detected or in case of ESC activation, the safety module was activated and switched to a certain constant damper current, which provided safe driving. Details regarding this concept can be found in [20].

Furthermore, there was an SH/GH controller included in the overall control system, which acted as a benchmark controller. The vehicle software allowed us to switch between the SH/GH and the RL controller. The design and parameterization of this reference controller is presented in the following section.

5.3. Design and Optimization of the Benchmark Controller

In order to assess the trained controller on the real vehicle, a benchmark controller was necessary (see [20]). In this work, we used a slightly modified version of the SH/GH controller originally presented in [2]. The original version of the SH/GH controller postulated a virtual damper between the sky and the chassis as well as a virtual damper between the road and the wheel mass. The virtual forces, which would result from the two virtual dampers, were then applied by the semi-active damper, if the damper characteristics allowed it. Otherwise, the forces were clipped to the maximum applicable forces. The parameters of the controller were the damping coefficients of both the virtual skyhook damper and the virtual groundhook damper.

Even though this concept is appealing, the application was highly dependent on the measurement quality of the damper velocity. The SH/GH implementation described above output a desired damper force, but the interface for controlling the damper was a desired damper current. To obtain the desired damper current for a given force, the damper characteristics depicted in Figure 3 had to be inverted. The calculation was then highly dependent on the damper velocity. We found that this dependency affected the applicability of the SH/GH controller implementation, in which the controller output a desired damper force.

We used a modified version of the SH/GH controller with a current interface to minimize the dependency on the damper velocity measurement. In this controller variant, the controller directly output a desired damper current, instead of a damper force. The structure of the benchmark controller is depicted in Figure 9, where the SH current demand i_{SH} is calculated as

$$i_{\mathrm{SH}} = \begin{cases} k_{\mathrm{SH}}\, v_{\mathrm{c}} & \text{if } v_{\mathrm{c}} \cdot v_{\mathrm{d}} \geq 0 \\ 0 & \text{else} \end{cases} \tag{17}$$

and the GH current setpoint i_{GH} is calculated as

$$i_{\mathrm{GH}} = \begin{cases} k_{\mathrm{GH}}\, v_{\mathrm{w}} & \text{if } v_{\mathrm{w}} \cdot v_{\mathrm{d}} < 0 \\ 0 & \text{else} \end{cases}. \tag{18}$$

In this equation, v_{w} denotes the vertical wheel velocity and v_{d} denotes the damper velocity. In our convention, $v_{\mathrm{d}} < 0$ represents a compression of the damper.

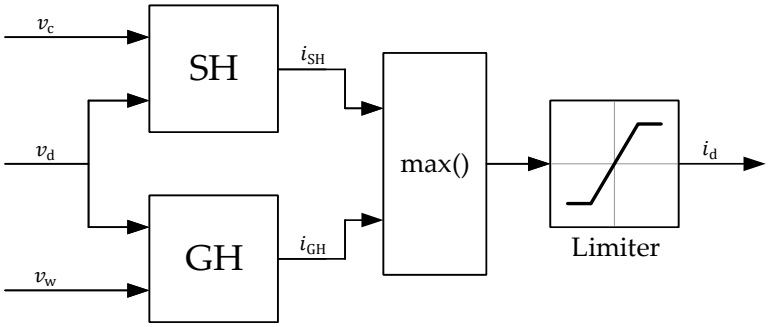


Figure 9. Structure of the benchmark controller.

To ensure a fair comparison, we optimized the controller parameters k_{SH} and k_{GH} in Dymola using the Optimization Library [26]. The optimization was performed on the QVM described in Section 3 and on different artificial road excitations. We optimized a separate parameter set for each vehicle corner using integral variants of the objectives J_c and J_{rh} from Equations (14) and (15).

5.4. Verification Toolchain

For the application on the real vehicle, the trained agents had to be embedded in the described vertical dynamics control system that is implemented in MATLAB/Simulink R2022b (cf. Section 5.2). The complete control system could then be deployed on embedded targets such as the AFM's RCP platform. Before the control system was tested in real-world experiments, it was favorable to verify the complete control system in a software-in-the-loop (SiL) simulation using the full-vehicle model. Therefore, it was necessary to first transfer the trained agents from the Python- and PyTorch-based training environment to the vertical dynamics control system in Simulink. The transfer from PyTorch to Simulink was realized by a custom code generation that implemented the whole neural network including its weights in plain C-Code. This C-Code could then be automatically wrapped into an S-Function by means of the MATLAB Legacy Code Tool.

In a second step, we transfer the whole control system to Modelica/Dymola in order to enable the SiL tests, as depicted in Figure 10. The transfer from Simulink to Dymola was carried out via the Functional Mock-up Interface (FMI), supported by both simulation tools. This automatized process ensured that the whole control system, which was deployed to the AFM's RCP, was verified in the FVM simulation setup.

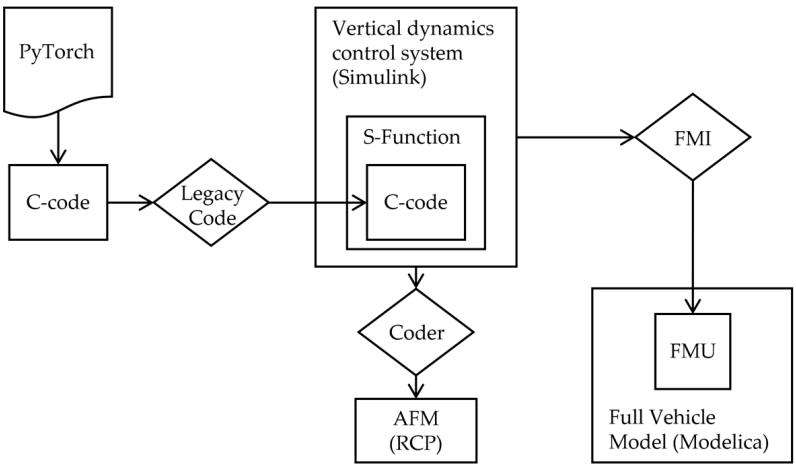
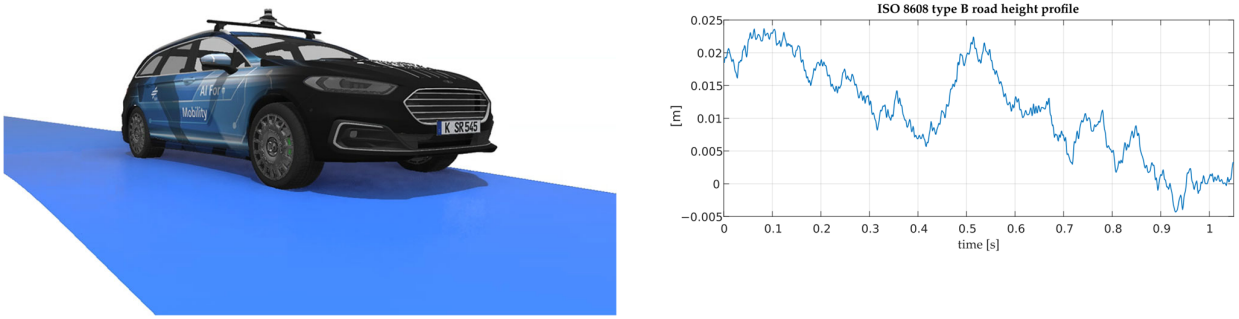


Figure 10. Verification and application toolchain for the trained RL agents.

5.5. Verification Simulation

After the integration of the whole control system into the FVM simulation setup as FMU, we conducted several simulations to verify the RL agent together with the control system. Additionally, the SH/GH benchmark controller was simulated as a reference. To verify the RL agent, several time domain signals were evaluated and checked for plausibility. Since we were mainly interested in the performance of the RL agent on the real vehicle, the verification simulations conducted on the FVM delivered only qualitative results and prepared for the real-world tests.

Figure 11a shows the rendering of the simulated vehicle and, in Figure 11b, the road height profile for an ISO 8608 [19] type B road is displayed. The feedback of the controller and the vehicle response of the FVM for the excitation with the selected type B road are depicted in Figure 12. The depicted signals are the dynamic wheel load, the body acceleration, and the damper currents calculated by the RL agent or the SH/GH, respectively. The depicted signals are all exemplary for the front left side of the vehicle. We selected the ISO 8608 type B road as excitation for the depicted verification simulation, as ISO 8608 road types A and B are the most common road types according to [45]. Since road type A is very smooth, we chose road type B as representative but still wavy enough to excite the controller. In the simulation, the vehicle was initialized with a velocity of 95 km/h, which is below the recommended maximum velocity for this road type, as analyzed in [45].



(**a**) (**b**)

Figure 11. (**a**) Rendering of the AFM's full-vehicle model simulation setup (adapted from [20]) and (**b**) part of the ISO 8608 type B road height profile used as excitation for verification.

Figure 12 reveals that the dynamic wheel load as well as the chassis acceleration were similar for both controller types. Additionally, it can be seen that the commanded signal calculated by the RL policy depicted in blue on the bottom subplot was reasonably smooth. This is an important check, as ANN-based policies obtained from RL sometimes tend to be very noisy. In contrast to the SH/GH controller, the RL policy made fewer interventions with high currents and did not induce big current jumps. The SH/GH controller relied on the direction of the damper velocity with respect to both the wheel velocity and chassis velocity. Therefore, it was prone to command current jumps in case one of the velocities changed direction.

Additional simulations verified the integrity of the whole control system together with the embedded RL agent. The application of the RL agent, which was trained on QVMs, in the FVM did not show concerning behavior. Therefore, we concluded that the control system was sufficient to be tested on the vehicle in real-world tests.

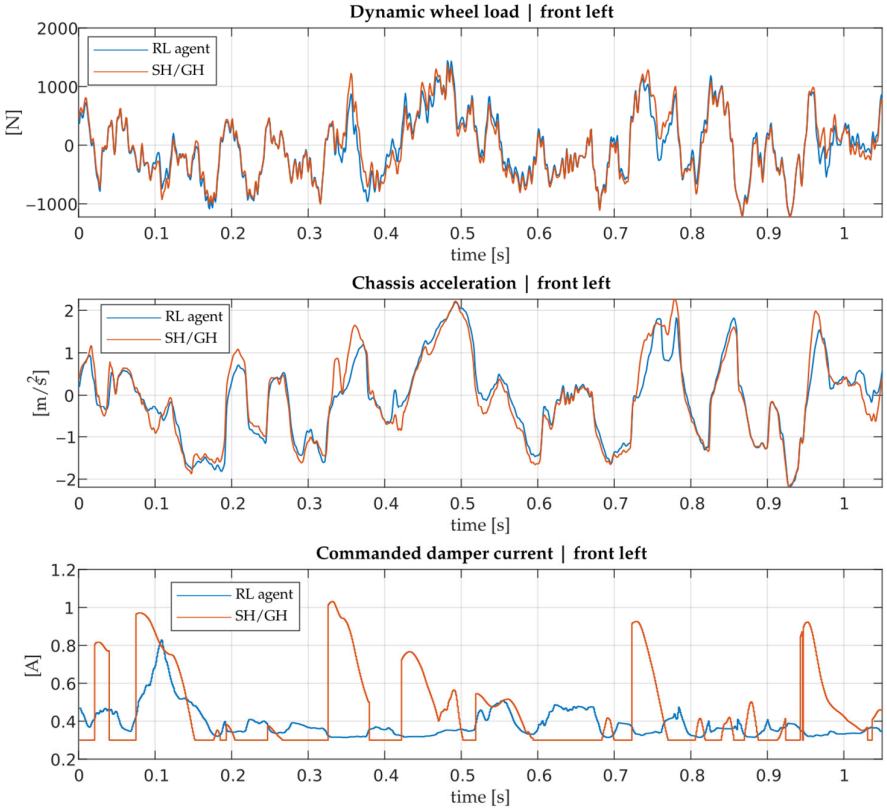


Figure 12. Time domain plots of the FVM simulation setup subject to excitation with ISO 8608 road type B with a velocity of 95 km/h. All signals are exemplary, shown for the front left side of the vehicle.

6. Real-World Test

After the SIL tests confirmed the integrity of the selected RL agent within the vertical dynamics control algorithm, they were tested in real-world experiments (also see [20]). For a quantitative analysis, tests were once again carried out using the four-post test rig, which enabled a precise repetition of excitation profiles and, thus, the direct comparison of different controllers. Further, high-velocity test drives were carried out on a real road with a rough surface and a significant ground bump. As these experiments were not exactly reproduceable, only a qualitative assessment of the controllers was possible.

In the conducted tests, the selected RL agent was compared to a baseline SH/GH controller that is described in Section 5.3. In the following, the experimental setup is described, the obtained results are both illustrated and discussed, and a concluding assessment of the RL agents is presented.

6.1. Results from the Test Drives

In addition to the tests on the test rig, the controller was also tested while driving on real roads, as depicted in Figure 13. The sensor setup during the test drives on the road reduces to the sensors which are mounted on the vehicle (cf. Table A2). In particular, the ground-based measurements, i.e., sensors integrated in the posts, were unavailable for these tests. As a result, the assessment of the controllers was limited to the subjective perception of an experienced test driver.

Figure 13. Real-world test drive on a bumpy road.

The tests were carried out on roads of different characteristics and with various driving maneuvers. In contrast to the test rig, no benchmark measurements were available for the road tests. Thus, the tests were limited to objectification by an experienced test driver and application engineer. The controller modules described in Section 5.2 and designed to ensure safe driving operation were able to intervene as expected. Driver-induced excitations were handled by the predictive feed-forward control and the safety module. The RL controller was able to respond to road-induced excitations and ensure a comfortable driving experience. While driving on a bumpy road, as depicted in Figure 13, the RL controller was able to ensure safe body control and safe driving stability even when extreme road excitations occurred. The test driver confirmed that the RL controller minimized the chassis movement and, thus, provided a comfortable driving experience.

6.2. Results from the Four-Post Test Rig

The final experiments on the four-post test rig, which was also used to obtain measurement data (cf. Figure 1), were conducted to quantitatively evaluate the trained controller. To set the performance of the RL-based controller into perspective, the offline-optimized benchmark controller described in Section 5.3 was also evaluated on the same excitations. The evaluation of the controllers on the four-post test rig provided several advantages: First of all, every excitation could exactly be repeated for every controller. Quantitatively evaluating different controllers on real roads requires a high accuracy for each repetition. Even small lateral displacements between two experiments might result in altered vertical excitation, e.g., if in one experiment the wheel hits a pothole and misses it in the next pass. The second advantage of the test rig was the additional sensor setup. Force sensors inside the posts could directly measure the wheel load, and high-quality acceleration sensors were additionally used to obtain the chassis acceleration. Measuring the wheel load in real-world driving tests requires additional modification of the vehicle and, thus, was not feasible. To enable a holistic analysis of the vehicle dynamics, all the quantities listed in Table A2 were measured for different vertical excitations.

We used a wide range of excitation types to evaluate the performance of the controllers. The first excitation category was sine sweeps ranging from 1 Hz to 30 Hz with exponentially increasing frequency and different constant post zero-crossing velocities of (50, 100, 150, 200, and 250) mm/s. The second type of excitations were synthetic road excitations of type A up to type D according to ISO 8608 [19]. Type A roads represent the smoothest roads and type D represents the bumpiest ones. The work presented in [45] concluded that type A and B roads typically represent roads such as motorways or other

high-quality roads. Roads of Type D and worse are often used to represent unpaved roads for which a maximum velocity of < 15 km/h is recommended [45]. We created the synthetic road excitations such that the vehicle started with a virtual velocity of 1 m/s and increased the velocity linearly up to a maximum velocity. Afterwards, we then decreased the velocity again down to 1 m/s. Additionally, two different versions of the synthetic road were tested: In the first version, all posts were excited with the same excitation, i.e., a synchronous excitation. For the second version, which excited the posts in an asynchronous fashion, the phase between the posts was randomly shifted. The last set of excitations are real-road excitations, which were recorded and reproduced on the testbench and represented both a rough road and a rough bump.

From the obtained measurement data, we calculated the road-holding criterion (cf. Equation (15)) and a comfort criterion inspired by ISO 2631-1 [43]. The norm ISO 2631-1 defines comfort metrics for humans, which are subject to whole-body vibrations. To calculate the vertical excitation comfort metric according to ISO 2631-1, the seat surface acceleration first had to be modified by specified filters. Afterwards, the RMS of this filtered acceleration was calculated. Since, in our test setup, the vertical acceleration of the seat surface was not measured, we calculated the filtered RMS for each of the three chassis accelerations and took the mean value for evaluation. After calculating the comfort and road-holding criterion for both the RL-based and the benchmark controller, a reference value was calculated by normalizing the metrics of the RL-based controller to the metrics of the benchmark controller. This resulted in a metric that was easy to interpret: Values below 1 indicated that the RL-based controller performed better on this metric, and vice versa. The obtained results are listed in Table 4 and depicted as a pareto plot in Figure 14.

Table 4. Normalized performance metrics of the trained controller on different excitations. Metrics smaller than 1, depicted in green, indicate a superior performance of the RL agent; metrics greater than 1, depicted in red, represent a superior performance of the benchmark controller (compare [20]).

Excitation	Type	Adapted ISO 2631 Comfort Criterion	Road-Holding Criterion
Exponential Sine-Sweep (1–30 Hz | 50 mm/s)	Sweep	0.971	1.017
Exponential Sine-Sweep (1–30 Hz | 100 mm/s)	Sweep	1.019	1.028
Exponential Sine-Sweep (1–30 Hz | 150 mm/s)	Sweep	1.083	0.921
Exponential Sine-Sweep (1–30 Hz | 200 mm/s)	Sweep	1.108	0.871
Exponential Sine-Sweep (1–30 Hz | 250 mm/s)	Sweep	1.092	0.842
ISO 8608 Type A Road (asynchronous)	Road-like	0.931	0.954
ISO 8608 Type B Road (asynchronous)	Road-like	0.967	0.980
ISO 8608 Type C Road (asynchronous)	Road-like	0.998	0.993
ISO 8608 Type D Road (asynchronous)	Road-like	1.024	0.991
ISO 8608 Type A Road (synchronous)	Road-like	0.947	0.943
ISO 8608 Type C Road (synchronous)	Road-like	0.993	0.995
ISO 8608 Type D Road (synchronous)	Road-like	1.032	0.991
Real Road Replay: Rough Bump	Road-like	0.924	0.982
Real Road Replay: Rough Road	Road-like	0.962	0.992

Figure 14 summarizes the test results in a pareto plot: All markers in the left bottom quadrant represent an excitation in which the RL agent performed better in both the road-holding as well as the adapted comfort metric. All markers with a road-holding criterion greater than 1 correspond to an excitation where the RL agent performed worse. Accordingly, markers with an adapted comfort criterion greater than 1 represent an excitation where the agent performed worse. The subplots of Figure 14 depict the metrics on different excitation types. Remark: Due to a corrupted measurement, the metrics for the road type B are missing on the synchronous road excitations.

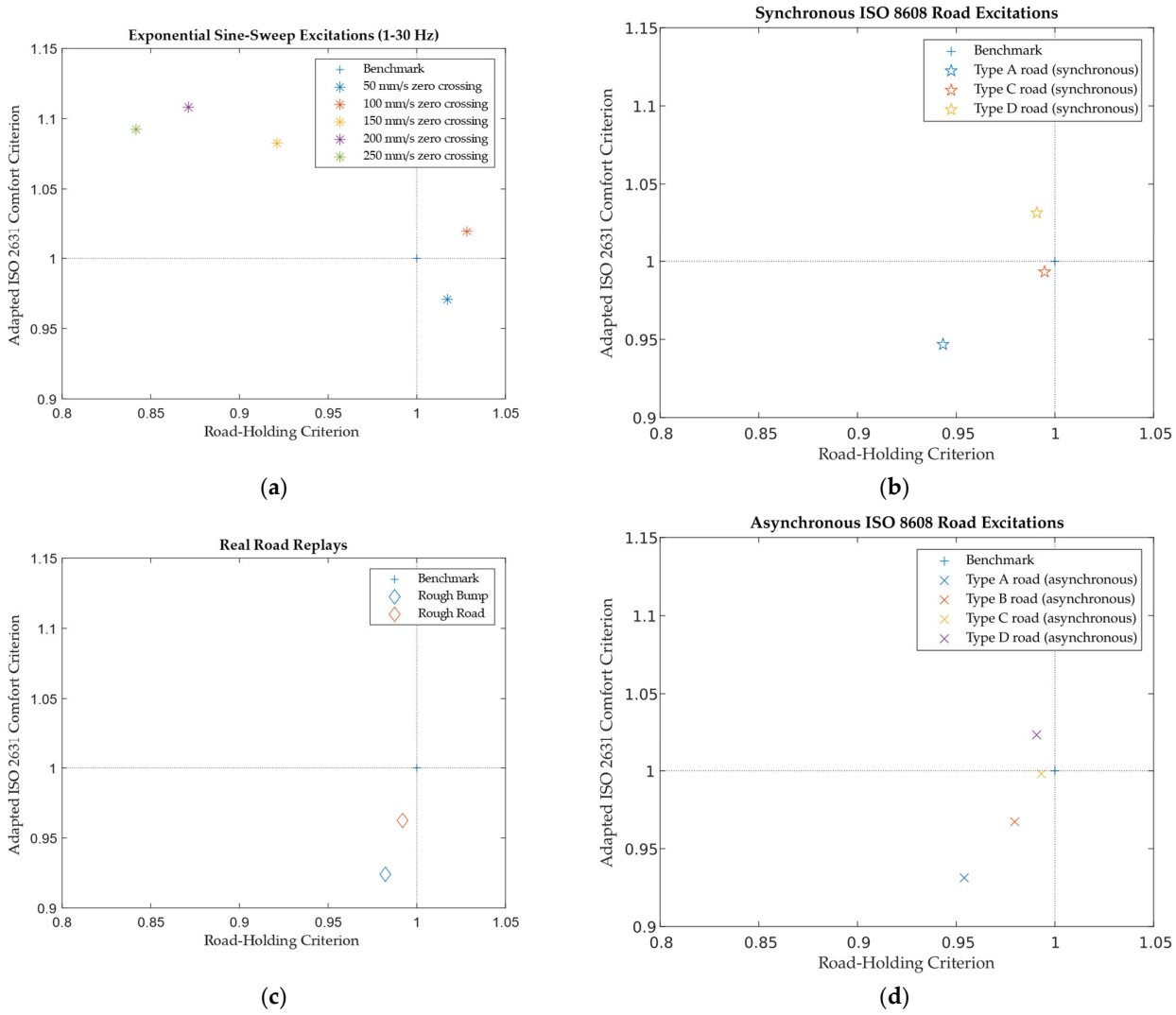


Figure 14. Normalized performance metrics of the trained controller as pareto plots on (**a**) sine sweep, (**b**) synchronous synthetic road excitations, (**c**) real-road replays, and (**d**) asynchronous synthetic road excitations. Metrics smaller than 1 represent a superior performance of the RL agent, and metrics greater than 1 correspond to a superior performance of the benchmark controller. (Remark: Due to a corrupted measurement, road type B is missing in subplot (**b**)).

Figure 14a shows that the RL-based controller can improve the road-holding for the sweeps with a zero-crossing velocity $\geq$ 150 mm/s but thereby deteriorates the comfort criterion. For the sweep with a 50 mm/s zero-crossing velocity, the RL agent can improve the comfort but deteriorates the road-holding. Finally, for the excitation with 100 mm/s, the benchmark controller outperformed the RL agent on both metrics. In contrast, Figure 14b–d reveal that the RL agent was able to improve the road-holding metric for all examined road-like excitations. Apart from two exceptions, the trained controller also outperformed the benchmark controller in the adapted ISO 2631-1 comfort criterion. In Figure 14b,d, the two excitations where the trained controller could only improve the road-holding metric were, in both cases, the road type D, which corresponded to an unpaved road [45]. Additionally, Figure 14b,d also show that the trained controller performed worse with a decreasing road quality. Nevertheless, the trained agent performed better than the benchmark controller on both metrics for all road types apart from type D. The observation that the trained agent was able to handle road-like excitations better than the benchmark controller was also backed by the tests with the road replay excitations, depicted in Figure 14c.

Since the control policy of the RL agent was obtained by a data-driven training process, it was not apparent why the RL agent performed well on some of the excitations and worse on others. Additionally, the semi-active vertical dynamics is a highly complex control problem for which an optimal solution is not trivial. This makes the reasoning of the learned RL policy intricate. In general, the comparison of the control signals of both controller variants on the sweep inputs showed that the trained controller applied higher damper currents, especially at higher frequencies. This way, the RL agent outperformed the benchmark controller in controlling the wheel movement and, therefore, improved road-holding for the sweep excitations with a zero-crossing velocity ≥ 150 mm/s. One explanation for the superior performance of the learned controller for higher frequencies lies in the choice of the RL *discount rate* $\gamma = 0.99$ together with the sample time $T_s = 1$ ms: in RL, the algorithm tries to optimize the *discounted future return* G, defined as the discounted sum of future rewards R at timestep i:

$$G_i = \sum_{k=0}^{N} \gamma^k R_{i+k+1}. \tag{19}$$

This means that the contribution of future rewards decreases exponentially with the discount rate γ. For $\gamma = 0.99$ and a sample time $T_s = 1$ ms, this means that for 0.23 s the reward contribution is already decreased to less then 10%. This results in a relatively short horizon and might explain the difficulties of the agent to deal with the lower frequencies during the 1–30 Hz sweep excitation. During the training stage, we also performed trainings with $\gamma = 0.999$. However, the results did not meet the desired performance requirements during the agent assessment, described in Section 4.2.4.

To summarize, the results of the four-post test rig evaluation of the trained controller, we can conclude that the trained RL-based controller is able to outperform the offline-optimized benchmark controller on road-like excitations, improving the comfort criterion by about 2.5% and the road-holding criterion by about 2.0% on average.

7. Summary and Outlook

In this work, we covered the whole RL controller design process for the semi-active vertical dynamics control problem. We derived a QVM-based enhanced training model, whose structure and parameters were obtained by optimization based on measurement data. We showed that our modeling approach approximates the real measurement data better than a standard QVM approach. Additionally, we derived a damper model and parametrized it by optimization on measurement data.

The obtained models were then utilized to train an RL-based control policy. We proposed one way of incorporating different objectives into the reward function, which included the consideration of penalizing big jumps in the damper force. This consideration was based on previous real-world observations in which high damper force jumps induced undesirable bump sounds. After the training, all obtained controller variants were compared in an agent performance assessment, and the best was selected for further evaluation.

The selected agent was then validated and tested in a high-fidelity FVM simulation setup. After the assessment proved the integrity of the trained controller, the agent was subject to quantitative and qualitative real-world tests. To ensure a fair comparison, a benchmark controller was optimized in simulation. The quantitative evaluation on a real-world four-post test rig revealed that the RL-based controller was able to outperform the offline-optimized benchmark controller on road-like excitations, improving the comfort criterion by about 2.5% and the road-holding criterion by about 2.0% on average. An additional qualitative driving assessment on real roads showed that the RL-based controller, together with a feed-forward module responsible for handling the driver, induced disturbances and ensured safe body control and safe driving stability even when extreme road excitations occurred. Future work will address the improvement of the RL-based controller on rough road excitations.

Author Contributions: Conceptualization, J.U. and J.B.; methodology, J.U., J.R., A.P., J.B., T.K. and J.T.; software, J.U., T.K., A.P., J.R., J.T. and J.B.; validation, J.U., J.B., J.T., T.K., J.R. and A.P.; data curation, J.U., T.K. and J.R.; writing—original draft preparation, J.U., J.R., A.P., J.T., T.K. and J.B.; writing—review and editing, J.U., J.R., A.P., J.T., T.K. and J.B.; project administration, A.P., J.B. and J.U.; funding acquisition, J.U., A.P. and J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by German Federal Ministry of Education and Research (BMBF, project *KIFAHR: KI-basierte Fahrwerksregelung*, grant number 01IS20010A). Additionally, the authors received DLR basic funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank KW automotive GmbH for excellent cooperation during the project. We want to highlight the support provided in the experiments with the test vehicle on the four-post testbench as well as the measurements of the individual components. Additionally, we want to thank KW for equipping the vehicle and their general expertise on vertical dynamics. We especially want to thank Michael Rohn, Udo Wahl, Steffen Klenk, Mario Koch, and Marc Ziegler. Our thanks also go to our colleague Daniel Baumgartner for supporting the preparation and execution of the real-world experiments. Additionally, we want to thank our former colleagues Christina Schreppel, Benedikt Helling, and Caspar Bieri, who contributed to the project with their preliminary work.

Conflicts of Interest: The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Appendix A

Appendix A.1. Results of the QVM Structure and Parameter Optimization

In Table A1, the results of the QVM parameter optimization described in Section 3.3 and the best QVM model structure are summarized.

Table A1. Resulting model structure and parameters from the optimization-based parametrization of the QVM. To set parameters into perspective, additional reference values are listed, where available.

	Front Left (FL)	**Front Right (FR)**	**Rear Left (RL)**
QVM Structure	Engine QVM	Engine QVM	Topmount QVM
Body mass [kg]	278	272	426
Wheel mass [kg]	52.0	51.4	45.8
Suspension spring stiffness [N/m]	$5.51 \cdot 10^4$	$4.78 \cdot 10^4$	$9.32 \cdot 10^4$
Tire spring stiffness [N/m]	$3.52 \cdot 10^5$	$3.64 \cdot 10^5$	$3.94 \cdot 10^5$
Tire damping [Ns/m]	$1.13 \cdot 10^3$	$1.23 \cdot 10^3$	$8.14 \cdot 10^2$
Spring ratio $i_{a,s}$	0.806	0.843	0.661
Spring ratio $i_{b,s}$	0.0	0.0445	0.0
Damper ratio $i_{a,d}$	0.805	0.744	0.710
Damper ratio $i_{b,d}$	0.0	0.0365	1.0
Engine mass [kg]	171	149	–
Engine bearing stiffness [N/m]	$4.36 \cdot 10^5$	$3.41 \cdot 10^5$	–
Engine bearing damping [Ns/m]	$2.42 \cdot 10^3$	$2.53 \cdot 10^3$	–
Topmount bearing stiffness [N/m]	–	–	$6.27 \cdot 10^5$
Topmount bearing damping [Ns/m]	–	–	$4.06 \cdot 10^2$
Damper friction force [N]	42.0	48.0	103

Appendix A.2. List of Measured Signals during the Four-Post Test Rig Experiments

Note: All measurands shown in Table A2 are one-dimensional and refer to the vertical direction.

Table A2. Overview of the measurement setup (compare [20]).

Measurand	No. of Signals	Sensor Type	Comments	Availability	
				Test Rig	Vehicle
Post position	4	N.A.	Integrated test rig sensors	✓	✗
Post velocity	4	N.A.		✓	✗
Post acceleration	4	Accelerometer		✓	✗
Wheel load	4	Load cell		✓	✗
Wheel acceleration	4	Accelerometer	-	✓	✓
Body acceleration	3	Accelerometer	At rear axle only one sensor on the left-hand side	✓	✓
Deflection wheel–chassis	4	Linear potentiometer	Only available on test rig	✓	✗
Engine deflection	3	Linear potentiometer	-	✓	✓
Engine acceleration	1	Accelerometer	-	✓	✓
Damper deflection	4	Rotary potentiometers	-	✓	✓
Tire deflection	2	Laser sensor	Single-sided, front and rear. Only available on test rig	✓	✗
Damper current	4	Hall effect sensor	-	✓	✓

Appendix A.3. Hyperparameters Used for Training the RL Agent

We used the stable-baselines3 [33] SAC implementation for the training. The agent was trained with a sample time $T_s = 0.001$ s and the hyperparameters listed in Table A3.

Table A3. Hyperparameters applied in training the SAC agent.

Hyperparameter	Value
n-timesteps	$3{\cdot}10^6$
policy	"MlpPolicy"
policy_kwargs	"dict(net_arch = [64, 64])"
learning_rate	$1{\cdot}10^{-5}$
buffer_size	$1{\cdot}10^6$
learning_starts	100
batch_size	256
tau	0.005
gamma	0.99
train_freq	1
gradient_steps	1
ent_coef	"auto"
use_sde	false

Appendix A.4. Parametrization of the Reward Function

The following table lists the parameters of the reward function, which resulted in the most performant real-world agent.

Table A4. Reward function parameters.

Parameter	Value
k_{cm}	5
$k_{\Delta \mathrm{u}}$	0.5
k_{a}	2
k_{fj}	20
$\theta_{v_{\mathrm{d}}}$	0.01
$\theta_{\Delta u}$	0.01
m_{a}	$\frac{1.6}{1.3}$
b_{a}	$-\frac{1}{1.3}$

References

1. Karnopp, D.; Crosby, M.; Harwood, R. Vibration Control Using Semi-Active Force Generators. *J. Eng. Ind.* **1974**, *96*, 619–626. [CrossRef]
2. Valášek, M.; Novák, M.; Šika, Z.; Vaculín, O. Extended Ground-Hook—New Concept of Semi-Active Control of Truck's Suspension. *Veh. Syst. Dyn.* **1997**, *27*, 289–303. [CrossRef]
3. Poussot-Vassal, C.; Spelta, C.; Sename, O.; Savaresi, S.; Dugard, L. Survey and performance evaluation on some automotive semi-active suspension control methods: A comparative study on a single-corner model. *Annu. Rev. Control* **2012**, *36*, 148–160. [CrossRef]
4. Savaresi, S.; Spelta, C. Mixed Sky-Hook and ADD: Approaching the Filtering Limits of a Semi-Active Suspension. *J. Dyn. Syst. Meas. Control.* **2007**, *129*, 382–392. [CrossRef]
5. Jenelten, F.; He, J.; Farshidian, F.; Hutter, M. DTC: Deep Tracking Control. *Sci. Robot.* **2024**, *9*, eadh5401. [CrossRef] [PubMed]
6. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.; Pineau, J. An Introduction to Deep Reinforcement Learning. *Found. Trends® Mach. Learn.* **2018**, *11*, 219–354. [CrossRef]
7. Ruggaber, J.; Ahmic, K.; Brembeck, J.; Baumgartner, D.; Tobolář, J. AI-For-Mobility—A New Research Platform for AI-Based Control Methods. *Appl. Sci.* **2023**, *13*, 2879. [CrossRef]
8. Howell, M.; Frost, G.; Gordon, T.; Wu, Q. Continuous action reinforcement learning applied to vehicle suspension control. *Mechatronics* **1997**, *7*, 263–276. [CrossRef]
9. Tognetti, S.; Savaresi, S.; Spelta, C.; Restelli, M. Batch Reinforcement Learning for semi-active suspension control. In Proceedings of the 2009 IEEE International Conference on Control Applications (CCA), St. Petersburg, Russia, 8–10 July 2009; pp. 582–587. [CrossRef]
10. Dessort, R.; Chucholowski, C. Explicit model predictive control of semi-active suspension systems using Artificial Neural Networks (ANN). In *8th International Munich Chassis Symposium*; Pfeffer, P., Ed.; Springer Fachmedien Wiesbaden: Munich, Germany, 2017; pp. 207–228. [CrossRef]
11. Savaia, G.; Formentin, S.; Panzani, G.; Corno, M.; Savaresi, S. Enhancing skyhook for semi-active suspension control via machine learning. *IFAC J. Syst. Control* **2021**, *17*, 100161. [CrossRef]
12. Ming, L.; Yibin, L.; Xuewen, R.; Shuaishuai, Z.; Yanfang, Y. Semi-Active Suspension Control Based on Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 9978–9986. [CrossRef]
13. Liang, G.; Zhao, T.; Wei, Y. DDPG based self-learning active and model-constrained semi-active suspension control. In Proceedings of the IEEE 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), Tianjin, China, 29–31 October 2021; pp. 1–6. [CrossRef]
14. Han, S.-Y.; Liang, T. Reinforcement-Learning-Based Vibration Control for a Vehicle Semi-Active Suspension System via the PPO Approach. *Appl. Sci.* **2022**, *12*, 3078. [CrossRef]
15. Kim, S.; Kim, C.; Shin, S.; Kim, S.-W. Deep Reinforcement Learning for Semi-Active Suspension: A Feasibility Study. In Proceedings of the IEEE 2023 International Conference on Electronics, Information, and Communication (ICEIC), Singapore, 5–8 February 2023; pp. 1–5. [CrossRef]
16. Wang, Y.; Wang, C.; Zhao, S.; Guo, K. Research on Deep Reinforcement Learning Control Algorithm for Active Suspension Considering Uncertain Time Delay. *Sensors* **2023**, *23*, 7827. [CrossRef] [PubMed]
17. Lee, D.; Jin, S.; Lee, C. Deep Reinforcement Learning of Semi-Active Suspension Controller for Vehicle Ride Comfort. *IEEE Trans. Veh. Technol.* **2023**, *72*, 327–339. [CrossRef]
18. Yong, H.; Seo, J.; Kim, J.; Kim, M.; Choi, J. Suspension Control Strategies Using Switched Soft Actor-Critic Models for Real Roads. *IEEE Trans. Ind. Electron.* **2023**, *70*, 824–832. [CrossRef]
19. *ISO 8608:2016*; Mechanical Vibration—Road Surface Profiles—Reporting of Measured Data. International Organization for Standardization: Geneva, Switzerland, 2016.
20. Ultsch, J.; Tobolar, J.; Ruggaber, J.; Pfeiffer, A.; Kamp, T.; Brembeck, J.; Baumgartner, D.; Ziegler, M.; Wahl, U.; Rohn, M.; et al. *Sachbericht zum Projekt "KI-basierte Fahrwerksregelung KIFAHR"*; DLR Institut für Systemdynamik und Regelungstechnik: Oberpfaffenhofen, Germany, 2023. [CrossRef]

21. Peng, X.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation, ICRA, Brisbane, Australia, 21–25 May 2018; pp. 3803–3810. [CrossRef]
22. Modelica Association. Modelica—A Unified Object-Oriented Language for Systems Modeling—Language Specification. Available online: https://specification.modelica.org/maint/3.6/MLS.html (accessed on 5 June 2024).
23. Modelica Association. Functional Mock-Up Interface Specification. Available online: https://fmi-standard.org/ (accessed on 12 June 2024).
24. Ultsch, J.; Ruggaber, J.; Pfeiffer, A.; Schreppel, C.; Tobolář, J.; Brembeck, J.; Baumgartner, D. Advanced Controller Development Based on eFMI with Applications to Automotive Vertical Dynamics Control. *Actuators* **2021**, *10*, 301. [CrossRef]
25. Fleps-Dezasse, M.; Tobolar, J.; Pitzer, J. Modelling and parameter identification of a semi-active vehicle damper. In Proceedings of the 10th International Modelica Conference, Lund, Sweden, 10–12 March 2014; Linköping University Electronic Press: Linköping, Sweden, 2014; pp. 283–292. [CrossRef]
26. Pfeiffer, A. Optimization Library for Interactive Multi-Criteria Optimization Tasks. In Proceedings of the 9th International Modelica Conference, Munich, Germany, 3–5 September 2012; Linköping University Electronic Press: Linköping, Sweden, 2012; pp. 669–680. [CrossRef]
27. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA; London, UK, 2018; ISBN 9780262039246.
28. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602v1. [CrossRef]
29. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347v2. [CrossRef]
30. Lillicrap, T.; Hunt, J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016. [CrossRef]
31. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, ICML, Stockholm, Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; PMLR: New York, NY, USA, 2018; pp. 1861–1870.
32. Liang, E.; Liaw, R.; Nishihara, R.; Moritz, P.; Fox, R.; Goldberg, K.; Gonzalez, J.; Jordan, M.; Stoica, I. RLlib: Abstractions for Distributed Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning, ICML, Stockholm, Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; PMLR: New York, NY, USA, 2018; pp. 3053–3062.
33. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *J. Mach. Learn. Res.* **2021**, *22*, 1–8.
34. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**, arXiv:1606.01540v1. [CrossRef]
35. Towers, M.; Terry, J.; Kwiatkowski, A.; Balis, J.; Cola, G.; Deleu, T.; Goulão, M.; Kallinteris, A.; Arjun, K.; Krimmel, M.; et al. Gymnasium. 2023. Available online: https://zenodo.org/records/8127026 (accessed on 5 June 2024).
36. Ultsch, J.; Brembeck, J.; de Castro, R. Learning-Based Path Following Control for an Over-Actuated Robotic Vehicle. In *VDI-AUTOREG*; VDI Verlag: Düsseldorf, Germany, 2019; pp. 25–46. [CrossRef]
37. Ultsch, J.; Mirwald, J.; Brembeck, J.; de Castro, R. Reinforcement Learning-based Path Following Control for a Vehicle with Variable Delay in the Drivetrain. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium, IV, Las Vegas, NV, USA, 19 October–13 November 2020; pp. 532–539. [CrossRef]
38. Raffin, A. RL Baselines3 Zoo. Available online: https://github.com/DLR-RM/rl-baselines3-zoo (accessed on 12 June 2024).
39. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. Soft Actor-Critic Algorithms and Applications. *arXiv* **2019**, arXiv:1812.05905v2. [CrossRef]
40. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
41. Savaresi, S.; Poussot-Vassal, C.; Spelta, C.; Sename, O.; Dugard, L. *Semi-Active Suspension Control Design for Vehicles*; Butterworth-Heinemann: Oxford, UK, 2010; ISBN 978-0-08-096678-6.
42. Enders, E.; Burkhard, G.; Fent, F.; Lienkamp, M.; Schramm, D. Objectification methods for ride comfort. *Forsch. Ingenieurwesen* **2019**, *83*, 885–898. [CrossRef]
43. *ISO 2631-1:1997*; Mechanical Vibration and Shock—Evaluation of Human Exposure to Whole-Body Vibration. Part 1: General Requirements. International Organization for Standardization: Geneva, Switzerland, 1997.
44. Bünte, T.; Rill, G.; Ruggaber, J.; Tobolář, J. Modelling and Validation of the TMeasy Tyre Model for Extreme Parking Manoeuvres. In *Advances in Dynamics of Vehicles on Roads and Tracks II*; Orlova, A., Cole, D., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 1015–1025. [CrossRef]
45. Múčka, P. Simulated Road Profiles According to ISO 8608 in Vibration Analysis. *J. Test. Eval.* **2017**, *46*, 405–418. [CrossRef]