

## Article

# A VR Truck Docking Simulator Platform for Developing Personalized Driver Assistance

Pedro Ribeiro <sup>1,\*</sup>, André Frank Krause <sup>1</sup>, Phillipp Meesters <sup>1</sup>, Karel Kural <sup>2</sup>, Jason van Kolfschoten <sup>2</sup>, Marc-André Büchner <sup>1</sup>, Jens Ohlmann <sup>1</sup>, Christian Ressel <sup>1</sup>, Jan Benders <sup>2</sup> and Kai Essig <sup>1,\*</sup>

<sup>1</sup> Faculty of Communication and Environment, Rhine-Waal University of Applied Sciences, 47475 Kamp-Lintfort, Germany; andrefrank.krause@hochschule-rhein-waal.de (A.F.K.); Phillipp.Meesters@hsrw.org (P.M.); Marc-Andre.Buechner2@hsrw.org (M.-A.B.); Jens.Ohlmann@hsrw.org (J.O.); cr@hsrw.eu (C.R.)

<sup>2</sup> School of Engineering and Automotive, HAN University of Applied Sciences, 6826 CC Arnhem, The Netherlands; karel.kural@han.nl (K.K.); Jason.vanKolfschoten@han.nl (J.v.K.); Jan.Benders@han.nl (J.B.)

\* Correspondence: pedro.ribeiro@hochschule-rhein-waal.de (P.R.); kai.essig@hochschule-rhein-waal.de (K.E.)

**Abstract:** Professional truck drivers frequently face the challenging task of manually backwards manoeuvring articulated vehicles towards the loading bay. Logistics companies experience costs due to damage caused by vehicles performing this manoeuvre. However, driver assistance aimed to support drivers in this special scenario has not yet been clearly established. Additionally, to optimally improve the driving experience and the performance of the assisted drivers, the driver assistance must be able to continuously adapt to the needs and preferences of each driver. This paper presents the VISTA-Sim, a platform that uses a virtual reality (VR) simulator to develop and evaluate personalized driver assistance. This paper provides a comprehensive account of the VISTA-Sim, describing its development and main functionalities. The paper reports the usage of VISTA-Sim through the scenario of parking a semi-trailer truck in a loading bay, demonstrating how to learn from driver behaviours. Promising preliminary results indicate that this platform provides means to automatically learn from a driver's performance. The evolution of this platform can offer ideal conditions for the development of ADAS systems that can automatically and continuously learn from and adapt to an individual driver. Therefore, future ADAS systems can be better accepted and trusted by drivers. Finally, this paper discusses the future directions concerning the improvement of the platform.

**Keywords:** virtual reality; machine learning; driver assistance systems; personalization; semi-trailer truck



**Citation:** Ribeiro, P.; Krause, A.F.; Meesters, P.; Kural, K.; van Kolfschoten, J.; Büchner, M.-A.; Ohlmann, J.; Ressel, C.; Benders, J.; Essig, K. A VR Truck Docking Simulator Platform for Developing Personalized Driver Assistance. *Appl. Sci.* **2021**, *11*, 8911. <https://doi.org/10.3390/app11198911>

Academic Editors: Maria Torres Vega and Michele Russo

Received: 1 July 2021

Accepted: 15 September 2021

Published: 24 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The automotive sector became one of the largest investors in augmented (AR) and virtual reality (VR) technologies and is expected to reach about \$673 Billion by 2025 [1]. Particularly, the reduced time to market, the need for innovative products, application of new technologies and the need to continually improve quality in the automotive sector has led to an increase in the use of VR/AR applications for design, manufacturing, training and component evaluation [2,3]. Another important factor is the possibility to identify problems already in the early development process without the need for a time consuming and expensive implementation of a physical prototype. Technical improvements over recent years in hardware, software products and display technologies allow not only to model the design of the different parts of an automobile, but also to simulate their functionality in interactive and realistic driving simulations. These developments demonstrate that VR/AR technologies are the place where interdisciplinary fundamental research and engineering sciences meet to design and evaluate the techniques and processes of tomorrow.

Such environments are the perfect “what-if machine” for creating rich audiovisual-haptic immersions and situated experiences. They enable interactive systems that combine (real and virtual) sensory, motor and cognitive experiences and can be explored in an application and result-oriented manner, without the limitations of the technical feasibility of hardware components and the structure/space requirements for different laboratory environments and test car availability. In this way, designers, developers and customers can interact with products that are still in the development stage, with virtual rooms for physically correct visualizations and immersive simulations of driving dynamics in driving simulators. Acceptance and UX studies in automated vehicles have so far primarily been examined using surveys or flat-screen-based driving simulators. These simulators often evolved into VR simulators, providing a higher degree of immersion. Therefore, in recent times, VR simulators have become a helpful tool for researchers to investigate the influence of different factors on driver response behaviours under different environmental and driving conditions [4].

Another important development to study and predict driver behaviour became reasonable through the rapid developments in AI technologies. With the advent of the 3rd AI wave around 2020, AI started leading to machines capable of learning in a way that is much more similar to how humans learn. These systems will be able to learn from human behaviour, to understand context and meaning, and to use these capacities to adapt to different application fields (in contrast to current systems which only work well in a particular application environment). Modern AI approaches will also require fewer data samples for training, and will learn and function with less supervision as well as communicate with humans in a natural, adaptive and anticipative way, leading finally towards personalized and anticipative AI assistants. The three terms going along with the 3rd wave of AI applications are: explainability, interpretability and transparency. Explainable AI (XAI) means artificial intelligence in which the system outcome is made transparent so that it can be interpreted and understood by a human (in contrast to current systems where the input parameters and decision behaviour are hidden in a “black box”) [5,6]. With these novel competencies, XAI aims to promote a greater understanding and trustiness between humans and machines [7]. Therefore, future AI models must be able to explain their decisions, make their decision behaviour transparent, and accept corrective input from the user (augmented intelligence). The feedback from the user can also be used to further train the AI.

Bringing both technologies (XAI and VR/AR simulations) together opens up new perspectives in the development of intelligent vehicles which can learn from multimodal driver data recorded in VR/AR simulations, but can also use the insights to provide novice drivers with individualized and context-sensitive assistance if a deviation from expert behaviour is detected. This allows not only optimized training environments, but also the development of Advanced Driver Assistance Systems (ADAS), which can adapt individually to continuous human driver input and explain their decision behaviour, as well as provide adequate feedback. This will not only lead to a higher acceptance rate for future ADAS, but also increase the drivers’ trust in these systems. This is crucial, as drivers cannot be forced to make use of the system while logistics companies seek to limit the cost of damage and unnecessary delays. This paper illustrates the approaches toward such an intelligent VR-simulator platform for the design and evaluation of personalized ADAS for the truck docking process.

To explore these novel approaches, a specific driving scenario was chosen. In this scenario, truck drivers perform a rearward docking manoeuvre towards a loading bay (dock). The advantage of using this scenario lies in the limited operational design domain, which is less complex and well-structured compared with other well-known driving scenarios in which ADAS systems operate, such as lane change assistance or adaptive cruise control. The rearward docking scenario, by nature, limits the number of vehicles and the speed with which they operate. Additionally, by exploring this driving scenario, it is also possible to profit from the fact that the loading bays are static and their position

is well known. This driving scenario has been explored for several years, first within a project INTRALOG targeting the development of autonomous rearward docking [8] and currently through the VISTA project [9] (see Section 1.2). In the context of the VISTA project, a VR Truck-Docking Simulator was developed on which this paper is based and which is described in more detail in Section 2.

### 1.1. Background and Previous Work

Virtual reality technology has been applied in different areas of the automotive industry by diverse automotive manufacturers [10]. Marketing and sales are some of the application areas in which manufacturers explored the potential of VR technologies. For instance, VR environments that allow customers to configure a vehicle offer a more immersive and enhanced experience to customers. Additionally, the manufacturer benefits from a virtual showroom that saves resources such as space. Moreover, in the area of marketing and sales, the possibility to offer a virtual test drive was explored. The advantage here is that the automotive companies have a novel and powerful channel to advertise their product, even when the car is not yet released. In the application area of vehicle design, VR is used to support the development of the product. VR brings the advantage of improving the decision-making process and the product quality and it also allows rapid prototyping that might potentially reduce costs and the time-to-market.

In line with the application of VR by different automotive manufactures, the research and development of novel concepts have also been explored with the support of VR platforms. In fact, VR simulators have recently been used in the development and evaluation of ADAS. A good example is the application of VR to training. Educating users to interact with automated systems is considered highly important [11,12].

For example, learning about ADAS functionalities using a written manual has some drawbacks such as misinterpretation or forgetfulness. On the other hand, training in a real context might offer a richer and effective learning experience. However, there are not many automatic driving cars available for testing. Additionally, training drivers in a real setting is not risk-free: training drivers in some more dangerous scenarios (e.g., driver distraction) would be valuable, but such scenarios are naturally avoided. Hence, researchers explored VR simulators to train drivers. This provides drivers with the opportunity to have a safe and risk-free interaction with ADAS, promoting the experience of different scenarios such as the adaptive-cruise-control, or the automatic take-over request [13]. VR has several benefits: (1) prototypical traffic and environmental situations can be repeated as many times as necessary; (2) concepts such as a novel Human–Machine Interaction (HMI) approach can be simulated as digital prototypes and can be easily integrated into the VR simulation, saving time and money; (3) intensive tests with many participants might be executed, leading to valuable insights.

In a recent study, researchers investigated the effectiveness of a light VR-simulator to train automated vehicle drivers by comparing it against a fixed-base simulator [13]. This light VR system, composed of a Head-Mounted Display (HMD) and a game racing wheel, proved to promote an adequate level of immersion for learning about ADAS functionalities and offered the advantage to be portable and cost-effective. Despite reinforcing the idea that a VR-simulator is a valuable tool for training purposes in automated vehicles, the results of this study also suggest that participants preferred the light VR system in terms of usefulness, ease of use and realism.

Until now, researchers explored VR simulators and focused on the study of ADAS and automotive HMI on passenger cars. However, the literature has rarely mentioned VR simulators in the context of truck driving. Due to the increasing integration of ADAS functionalities in trucks, there is a necessity to find new ways of developing and designing ADAS for trucks. In a recent study, a light VR-simulator was used to examine different HMI designs with the intention to assess culture-related effects on the perception of and preference across German and Japanese truck drivers [14]. In this study, researchers highlighted that the light VR-simulator allowed the quick and efficient evaluation of HMI

designs. Besides this, researchers also suggested that the integration of eye-tracking might be valuable to better understand driving behaviour and HMI usage.

Recently, there has been an increase in the usage of neural network approaches for the analysis and prediction of driver behaviour based on input data retrieved from sensors within the car and from the outside world. To predict the driving behaviour, habits and intentions of the respective user in response to the external world and internal car-events, driver models were developed [15,16]. To date, these models are static systems, tailored to the average driver while not being sensitive to inter-driver and intra-personal differences, needs or preferences [17]. Therefore, each evaluation test has to be performed with several driver models representing the various driver types. A personalization of the system currently can only be done at the beginning of the drive or by manual driver interaction. It is believed that a continuous, individual adaptation to personal driving habits is crucial for a good driving experience: a non-personalized system might otherwise annoy the driver with too much or irrelevant information. Potential consequences might be that the driver gets sidetracked or even disables the system, as well as loses trust and confidence in the system. A continuous adaptation requires the synchronized integration of data streams from different components. Only a holistic system that continuously adapts to the individual user can reduce the number of accidents and road deaths by providing user- and situation-relevant feedback. This guarantees an optimal user experience and a reduction in the cognitive load for the driver. The application of neural network techniques for time-series prediction and classification of individual driver behaviour can open up new possibilities for the implementation of more flexible driver models.

Recurrent Neural Nets (RNN) are currently used in the automobile sector [18,19] and are mainly used for time-series prediction and classification. They can analyze data and behaviour over time, to identify temporal patterns and behaviours in the provided data sets. RNNs are used to track and predict paths of moving objects (e.g., pedestrians) and therefore to determine potential collisions. RNNs can also be used to predict human actions and potential future events. For example, Li et al. [18] trained an RNN with long short-term memory units to learn a human-like driver model that can predict future steering wheel angles based on road curvature, vehicle speed and previous steering movements. They argue that their approach provides more human-like steering wheel movements compared to preview-based models and can therefore increase the acceptance of autonomous vehicles if these will behave more like human drivers.

A research area that is gaining interest in the automotive research community concerns the individualization of ADAS. The concept aims at the adaptation of the ADAS' assistance functions to the drivers' preferences, skills and driving behaviour [17]. The authors suggest that individualization is a continuous process in which the ADAS' assistance functions adapt based on the driver behaviour. In that sense, the implementation of the individualization module must follow a driver-centric approach in which it continuously integrates the input of the driver. Darwish and Steinhauer [20] recently explored an approach that uses deep reinforcement learning to personalize the driving experience, focusing on a scenario in which the adaptive cruise control function is used to keep an optimal distance to the car in front. In this work, the authors emphasize that driver's behaviour and preferences are volatile, meaning that they can change rapidly depending on their experience and the novelty of the situations in which drivers have to operate. Hence, the individualization of ADAS is only achievable within a data-driven approach that is capable of performing adaptation in real-time and by relying on a limited amount of data. A personalized ADAS must be able to predict the driving behaviour, habits and intentions of the respective user. Additionally, external environmental data (traffic signs, pedestrians, obstacles, events) has to be available to the system using visual object recognition.

In fact, one of the greatest opportunities in exploring a VR-simulator in truck driving contexts is that multi-dimensional driver data can be recorded to inform the design and evaluation of ADAS functionalities, HMI concepts and driver models. For example, not only data about the vehicle (e.g., steering, braking, gear, truck position), but also data about

the driver can be recorded. For instance, the integration of a microphone can be useful to record verbal behaviour and hand-tracking allows the study of manual interaction with any vehicle device including HMI or non-verbal communication. Eye-tracking offers a multitude of possibilities, such as determining where the driver is looking or the evaluation of driving fatigue [21].

### *1.2. Content and Goals of the VISTA Project*

While performing the docking manoeuvre at the distribution centre, the driver has to operate a truck with one or even two attached trailers (truck combination) towards the loading bay. Handling the manoeuvre properly is a challenge: the driver has constrained visibility from the cabin and the manoeuvre area is limited in space. Additionally, the manoeuvre must be performed in a place that has a dynamic nature (e.g., other trucks manoeuvring, people walking, etc.), meaning that accidents can happen, resulting in significant costs due to collision damage. For that reason, the Intelligent Truck Applications in Logistics (INTRALOG) project developed and investigated the effect of an automated docking system [8]. Path tracking and path planning algorithms have been developed for logistic vehicle combinations with one or two articulations, for both forward and rearward docking manoeuvres.

Although some of the main problems of this driving scenario could be attenuated with the usage of sensors on the trailing vehicles and the usage of an automated docking system, it would also imply that all the trucks need the same technology integrated, resulting in significant costs. Moreover, trailers and trucks are often owned by different companies, which complicates the possibility for trailer instrumentation. Furthermore, logistic companies also have challenges in attracting skilled drivers, so they have a desire to provide support to less-trained drivers. This induces a desire for docking support functionality, suiting drivers with varying skills and experience. Based on these observations, a novel approach is currently being explored in the project VISTA (VIsion Supported Truck docking Assistant). The VISTA project aims to develop a framework integrating a camera-based localization system to track in real-time the position of truck and trailer(s) during the docking process at a distribution centre. Based on this localization system, the optimal docking path from the current position to the final unloading station is calculated and provided in the form of audiovisual assistive instructions (e.g., steering recommendation) either displayed as an appropriately coloured light array located above the windshield or on an HMI module displayed on a tablet placed on top of the cockpit in the truck cabin.

The target of the present research is to develop the VISTA-Sim, a platform that uses a VR-simulator as an environment to investigate driver performance, train drivers and develop and evaluate different forms of context-sensitive and personalized driver assistance feedback systems. The driving scenario in which this platform is studied refers to the docking of a truck combination towards the loading bay in a distribution centre. VISTA-Sim has two main goals. First, the simulator will serve as a tool that allows the validation of an HMI and different feedback components (such as lights or audiovisual hints, e.g., indication arrows or verbalized steering recommendations) designed for the VISTA project. This offers a driver-centred design approach that allows making decisions in an early stage of the project. The second goal is to use this platform as a tool to record data that can be used by machine-learning algorithms to learn about novice- and expert behaviours, allowing the implementation of a driver model. These insights can be used to estimate individual driving behaviour or to detect differences from expert driver performance, in order to provide context- and user-adequate feedback.

In summary, this paper proposes a novel and holistic approach to develop and evaluate personalized driver assistance using VR technology.

## **2. VISTA VR-Simulator Platform (VISTA-Sim)**

This section describes all the different components of the VISTA-Sim, as well as how they cooperate to provide an immersive truck-docking experience (see Figure 1). In VISTA-

Sim, the driver sits in front of a steering wheel with pedals and wears a Virtual Reality headset with an integrated eye-tracker (see Figure 2b). Integrated into the VR truck cabin is an HMI module showing the following conceptual features: bird-eye view, distance to the docking station and steering recommendation.

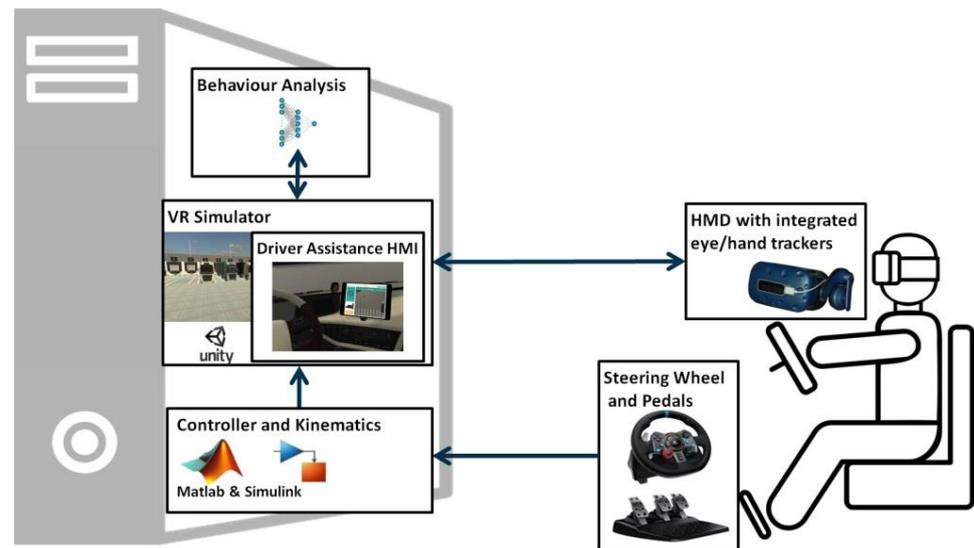


Figure 1. The high-level view of the VISTA-Sim architecture.



(a)



(b)

Figure 2. (a) Vista-Sim hardware setup, (b) Leap motion controller mounted on the HMD.

The system is composed of four main components: (1) path planner combined with the path tracking controller which runs in Simulink [22]; (2) the VR-simulator that uses the Unity 3D cross-platform game engine [23]; (3) the driver assistance HMI; and (4) the Behaviour Analysis Module.

The general functionality can be described as follows: a user wears VR glasses that display the environmental model of the driver's outlook from the vehicle cabin whilst operating in the distribution centre, and the HMI should support the user to successfully park the vehicle combination at the loading bay. The environmental model consists of a distribution centre, its surroundings, the vehicle combination and the integrated driver assistance HMI. The environmental model and HMI are powered by a kinematic model running in Simulink. The input for both the controller and the kinematic model is provided by the user through the physical steering wheel in terms of steer angle and pedal positions as a response to the visual inputs.

### 2.1. VISTA-Sim Hardware

This section describes the hardware setup (see Figure 2a) for providing the immersive experience of driving a truck, and specifically to perform docking manoeuvres at a simulated distribution centre. Following a similar approach to Sportillo et al. [13], there

was implemented a Light Virtual Reality Simulator that uses the Head-Mounted Display (HMD) HTC VIVE Pro Eye [24], which provides stereoscopic vision with a resolution of  $1440 \times 1600$  pixels per eye, a refresh rate of 90 Hz and a field of view of 110 degrees. The HMD is equipped with infrared sensors to track the user's position in real-time by detecting infrared pulses coming from two emitters. The HMD also comes with embedded eye-tracking and headphones. This HMD provides a highly immersive VR experience, while the user can freely observe the surrounding virtual world in any direction just by turning the head. Additionally, it was used the Logitech G29 driving system [25], consisting of a steering wheel and a set of pedals. With the driving system, the users can control the truck in the virtual world. Finally, the system also integrates the Leap motion controller [26], an optical tracking module to track user's hand poses in real-time. The Leap motion controller is mounted directly in front of the HMD, as can be seen in Figure 2b.

## 2.2. Path Planner and Path Tracking Controller

The main role of the path planner is to establish an optimal bi-directional reference path for the vehicle combination, which connects the initial pose and the terminal pose represented by the loading dock. On the other side, the role of the path tracking controller is to derive a steering angle based on the error between the actual pose of the vehicle and a previously established reference path. This error can subsequently be used to provide advice to the driver. Given the low operational speed (below 2 m/s), both subsystems are based on the assumption that the vehicle combination driving behaviour is perfectly kinematic, and thus no tyre slip is assumed. A high-level description of both subsystems follows.

### 2.2.1. Path Planner

The path planner uses a lattice-based approach in combination with motion primitives, where the environment is divided into a set of discrete states which can be connected and which can create the complete solution for the path. In the first step, the operational environment is described in terms of free and restricted space using polygons which specify the location, shape and size of the obstacles in the sensed environment. The blue polygon in Figure 3a depicts the available space, whereas the red polygons represent the obstacles, whose positions are also consistent with the environmental models in the VR. The state of the vehicle combination is defined by the position of the centre of the semitrailer axle in the global coordinate system  $x_2$  and  $y_2$ , the yaw angle of the semitrailer  $\theta_2$ , and the articulation angle  $\gamma$ . Thus, the path planning problem is reduced to finding the path between two discrete states in the path planning environment where the path segments, called motion primitives, are generated (see Figure 3b). This is achieved by solving an optimal control problem from one of the discrete states to the other, while using the kinematic equations of the articulated vehicle and differential constraints which limit the path curvature and steering angle rate and acceleration.

This guarantees that the created path segments are kinematically viable for the vehicle and are still negotiable by the human driver given human physiological limitations in actuation of the steering wheel. The detailed description of the optimal control problem with a desired cost function to optimize some parameters, the kinematic constraints and its solutions, can be found in [27,28].

Given a library of motion primitives, an algorithm is required to find an optimal combination of motion primitives to traverse from one point to another, using the free space within the operational environment and avoiding the obstacles. For this purpose, the graph search algorithm A star algorithm is used with a collision detection module, which checks for collisions while planning. The customized A star algorithm is also extensively explained in [27].

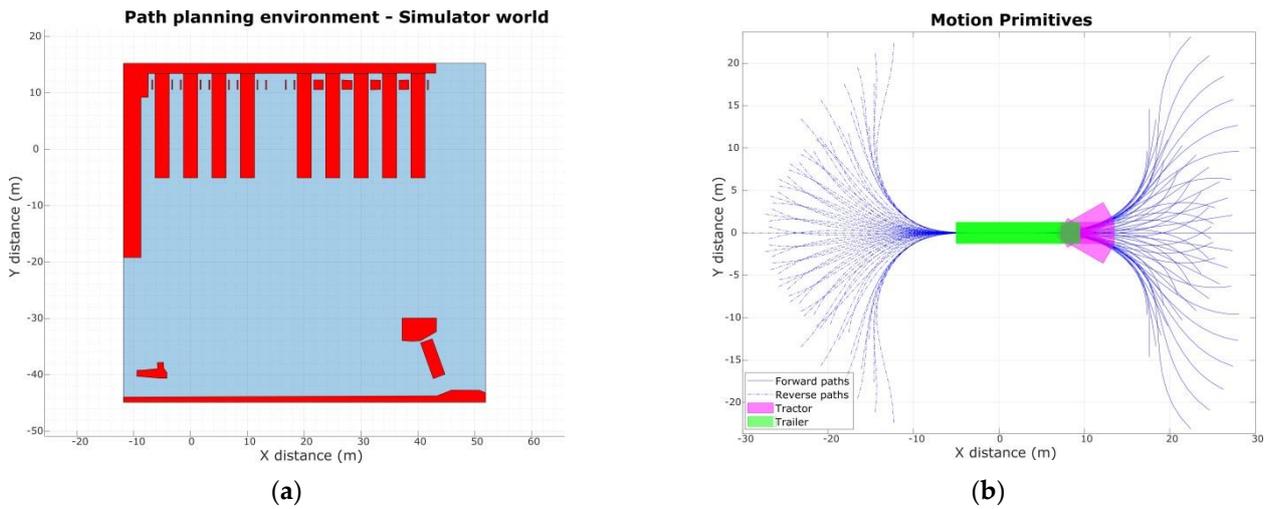


Figure 3. (a) Path planning environment with polygons; and (b) motion primitives of a truck semitrailer.

### 2.2.2. Path Tracking Controller

The main role of the path following controller is to navigate the central turn point  $U$  of the last vehicle unit (described by  $x_2$  and  $y_2$  in Figure 4) along the reference path, whilst actuating the steering angle of the hauling unit  $\delta$ . It is required that the controller is bi-directional, thus working for both the forward and reverse motion in the docking manoeuvre, and needs to be modular to be applicable to multiple vehicle combination layouts. The control problem is formulated as a path following problem and uses a technique of virtual tractor, fully described in [28]. Herewith, the vehicle combination is considered with respect to a set of known reference nodes  $R_i$ , which constitute the reference path generated by the path planner as shown in Figure 4. Each node  $R_i$  is defined by the position represented by  $(X_{ref(i)}, Y_{ref(i)})$  and orientation represented by  $\theta_{ref(i)}$ , all measured in the coordinate system  $e^{\rightarrow 0}$ . Although the reference path is characterized by discrete nodes, thanks to their high density, no interpolation is required. Subsequently, the lateral error  $e_{yU}$  can be obtained in the local coordinate system  $e^{\rightarrow 3}$  fixed to the controlled turn-point  $U$  of the rear-most vehicle unit.

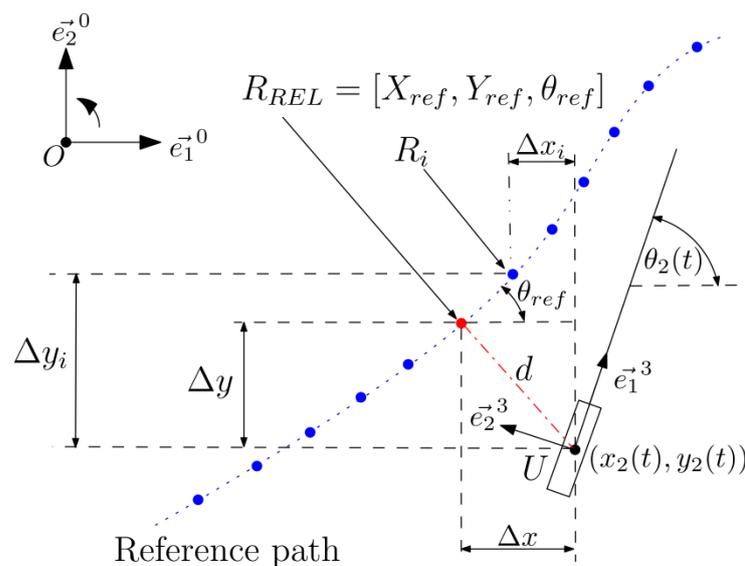


Figure 4. Determination of relevant reference point which is used to determine the tracking errors.

The lateral error reads:

$$e_{yU} = \Delta y \cos \theta_2 - \Delta x \sin \theta_2$$

Besides the lateral error, the controller also employs an angular error  $e_{\theta U}$  in order to ensure the rearmost vehicle has the same orientation as the relevant reference point  $R_{REL}$ . The angular error is defined by:

$$e_{\theta U} = f(x) = \begin{cases} \theta_{ref} - \theta_2, \wedge |\theta_{ref} - \theta_2| \leq \pi \\ \theta_{ref} - \theta_2 - 2\pi \text{sign}(\theta_{ref} - \theta_2), \wedge \text{otherwise} \end{cases}$$

To formalize the control problem, the classical model of nonlinear kinematic articulated vehicle can be rewritten as follows:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t), u(t)) \\ y(t) &= \begin{bmatrix} x_2(t) \\ y_2(t) \\ \theta_2(t) \end{bmatrix} \end{aligned}$$

In the model above,  $x = [\theta_0; x_0; y_0; \theta_2]^T$  are system states representing the pose of the hauling unit and yaw angle of the semitrailer,  $u = [\delta]$  is the control input representing the steering angle of the hauling unit, the velocity of the hauling unit  $v_0$  is known to be constant, which can be either positive or negative, and  $y$  is the output which gives the pose of the point  $U$  given by  $x_2, y_2$ , and  $\theta_2$ . To solve the control problem, the goal is to design the control input  $u$  such that closed-loop stability is guaranteed and the controlled output  $y$  follows the reference path, such that:

$$e(t) \begin{bmatrix} e_{y_U}(t) \\ e_{\theta_U}(t) \end{bmatrix} = h(y(t), R_i) \rightarrow 0, \text{ for } t \rightarrow \infty$$

In Figure 5, the control loop structure is depicted. As can be seen, it includes a known set of reference path nodes from which the relevant nodes are selected based on the current pose of point  $U$ , a controller, and a kinematic model of the articulated vehicle. The controller consists of three blocks, these being an error definition, a driver model and the inverse kinematic model, respectively. Further details are provided in [29], where the controller is also subjected to the stability analysis and the robustness is tested by deployment on the scaled vehicle lab-platform. Achieved results provided solid evidence that the controller framework may be employed also for “driver in the loop” applications, which is further exploited in the VISTA project and the VR simulator.

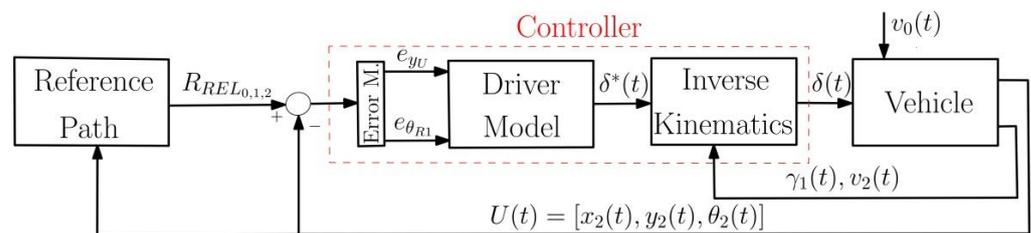


Figure 5. Controller structure.

### 2.3. Simulation Environment

The VR-simulator was developed using the Unity 3D game engine [23]. Unity is a powerful platform for the development of a VR simulator offering a simple VR integration. In Unity, the GameObjects are the building blocks that compose the 3D scenes. A GameObject is typically associated with a 3D object and aggregate one or more functional components which determine the behaviour and appearance of the GameObject.

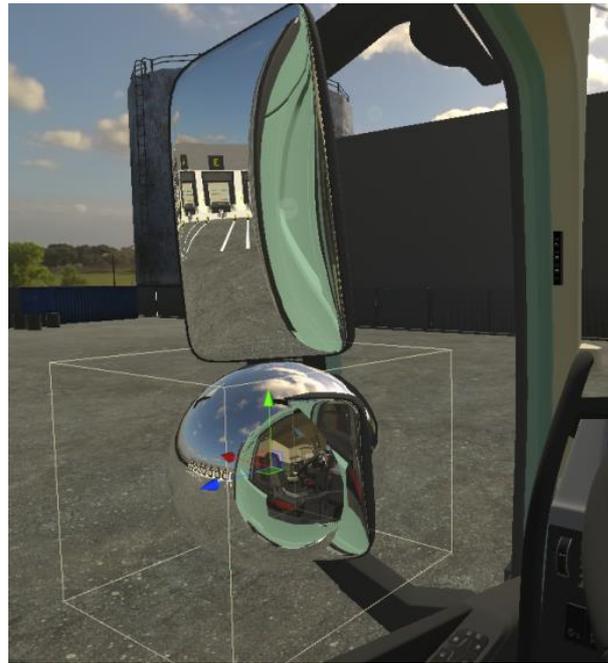
### 2.3.1. The Virtual Semi-Trailer Truck and the Distribution Centre

The VR-simulator allows the 3D representation of (i), the Volvo FH16 tractor unit and a trailer and (ii) the distribution centre environment. The distribution centre environment was modelled based on dimensions measured in the real world and consists of ten numbered loading docks and three parameterizable docking assist cameras. The floor of the distribution centre area has floor-marking guidelines aligned with the loading dock. Additionally, the texture of the floor can be easily changed whenever the simulator requires different conditions. The surrounding environment is visualized using a realistic skydome. Each loading dock also integrates a red and green light to inform the driver about the remaining distance of the truck to the docking door in order to support the docking process (see Figure 6a). Additionally, there is the possibility to determine the targeted loading dock and to park trucks in any lane (see Figure 6b).



**Figure 6.** (a) Distribution centre loading docks and the three docking assist cameras; (b) Distribution centre with a parametrized number of parked trucks.

Previous field studies with drivers wearing binocular mobile eye-tracking glasses while doing the truck docking process in the real world revealed that they mainly switch their gaze movements from the inside instruments to the mirrors. These results illustrate how important it is to simulate the mirrors realistically. For this, the curvature of the mirrors is of particular importance. To render reflections, reflection probes are currently used as a solution (see Figure 7). Although this technique offers a realistic experience, it also comes with some limitations. A reflection probe is similar to a camera that captures a spherical view of the 3D scene in all directions. This implies that the objects located outside the field of view increases the rendering effort per frame considerably. Since the truck has six mirrors, it can add up to significant effects on the frame rate. The solution adopted to overcome this problem was to dynamically deactivate or decrease the resolution and refresh rate of the reflection probes based on the current camera direction as well as the drivers' field of view determined from the recorded eye-tracking data. Note that the driver can also rotate each of the six mirrors in the simulator in order to meet his/her preferences.



**Figure 7.** Mirrors with real-time reflections using reflection probes.

### 2.3.2. Driver Assistance Human-Machine Interface (HMI)

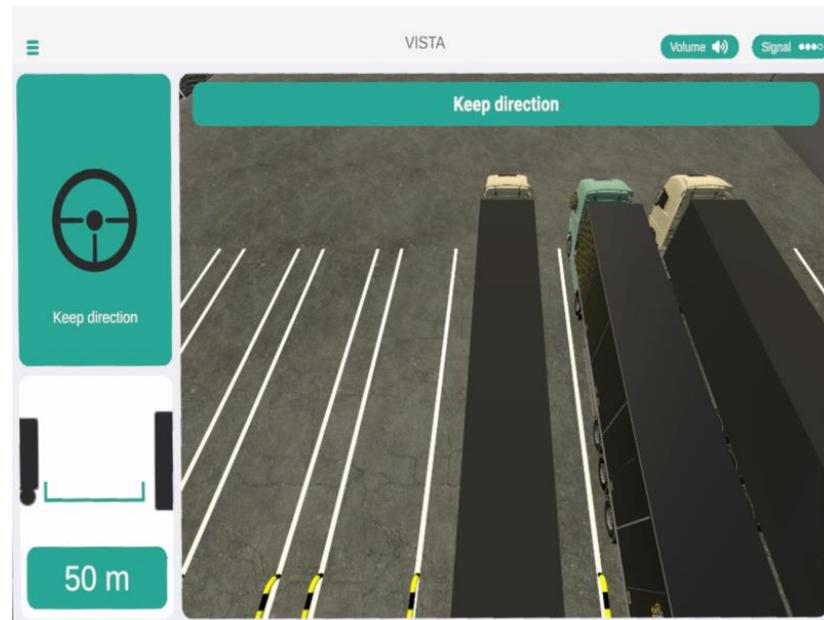
One of the central components of the VR-simulator that is integrated into the interior of the Volvo FH16 truck cabin is the driver assistance HMI. To be more specific, a 3D model of a tablet device was installed to run the driver assistance HMI and to present audio and graphical instructions to the user. The tablet can be placed at two different locations (centre or left side) (see Figure 8). Note that in project VISTA, the VR-simulator is used to evaluate a variety of HMI designs, not only limited to a tablet device-based HMI. However, for this research, a tablet-based HMI version is used underneath.



**Figure 8.** (a) HMI installed on the left side; (b) HMI installed in a central position.

Through the HMI, the driver can obtain valuable information (see Figure 9). For instance, the driver can check in which loading bay he/she has to park the truck, for which the HMI displays the identification number of the loading bay target. Additionally, based on the ideal path, the driver assistance HMI can inform in which direction the steering wheel has to be rotated. For this, two dynamic visual representations are used: (i) the stylized steering wheel complemented with an arrow and a text and (ii) the arrow-based instruction that is complemented with the text. Furthermore, the HMI also provides the distance to the loading bay and an embedded video feed of the camera system that overwatches the docking area. Finally, complementary audio cues are used to represent

steering angles and distance to the dock. This approach is employed to allow drivers to receive information without explicitly looking at the HMI screen.



**Figure 9.** Driver assistance HMI screenshot.

### 2.3.3. Integration with Path Tracking Controller

In order to implement the inter-process communication between the Simulink process and the Unity 3D process, a User Datagram Protocol (UDP) communication channel was established. With this approach, Simulink transmits in real-time: (i) the tractor's position and rotation; (ii) the trailer's position and rotation; (iii) the velocity; (iv) steering wheel angle; (v) the selected gear; (vi) the recommended steering angle; and (vii) the distance to the targeted docking station. These data are then received on the Unity side with the help of a script that processes the incoming packets and then distributes the information to the simulated truck. For example, based on the information received, the position and rotation of the truck are updated in real-time. In a similar way, the recommended steering angle and the distance to the targeted loading bay of the HMI's GameObject are updated in real-time and presented to the driver.

### 2.3.4. Tracking Systems

The VR-simulator uses the SRanipal-runtime to access the eye-tracking capabilities of the HTC VIVE Pro Eye [24]. The HMD has an integrated binocular eye-tracker with a sampling rate of 120 Hz, an estimated measured gaze accuracy of  $0.5^\circ$  as well as a trackable field of view of  $110^\circ$ . In the VR simulator, the main goal of eye-tracking is not only to determine in real-time which objects the driver is directly focusing on but also which objects he/she is looking at through the mirror (see Figure 10).

The simulator uses a combined gaze ray to determine which GameObject is focused on. The combined gaze ray originates between the centre of the left and right cornea and is cast through the intersection point of the left and right gaze ray (see Figure 11a). To use the combined gaze ray in unity, its normalized direction vector is used to cast a raycast from the VR camera against the scene. A GameObject that is hit by this ray is considered to be the current focus of the driver. In order to detect which GameObjects are focused in a mirror, the first ray is reflected off the surface it hits and is casted again in this direction (see Figure 11b). This happens only when the first ray hits a GameObject that is tagged as "mirror". Whenever the second ray hits any GameObject or Terrain, then a spherecast (i.e., a raycast that uses a sphere collider instead of a point) is cast. The spherecast moves

along the normal of the hit surface (see Figure 11b) and returns all GameObjects inside the sphere's radius as indirect focuses. A spherecast was used here to compensate for tracking errors caused by distortions in the mirror and imprecision of the eye-tracking algorithm over longer distances.

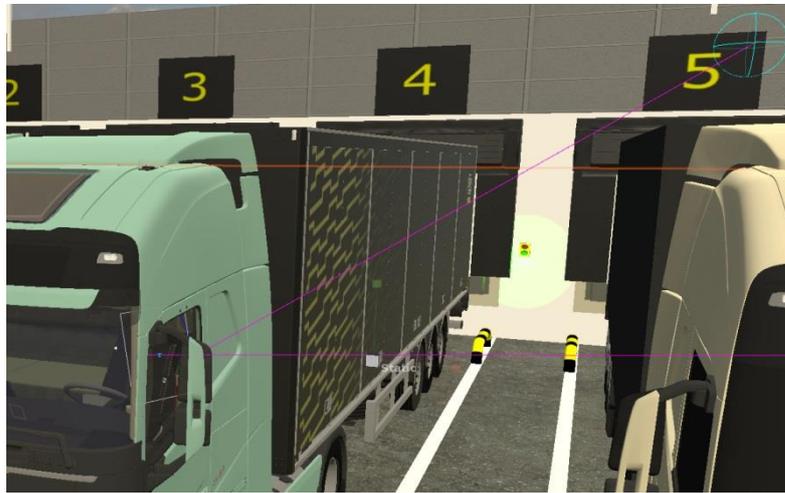


Figure 10. Using eye-tracking to identify the gazed object through a mirror.

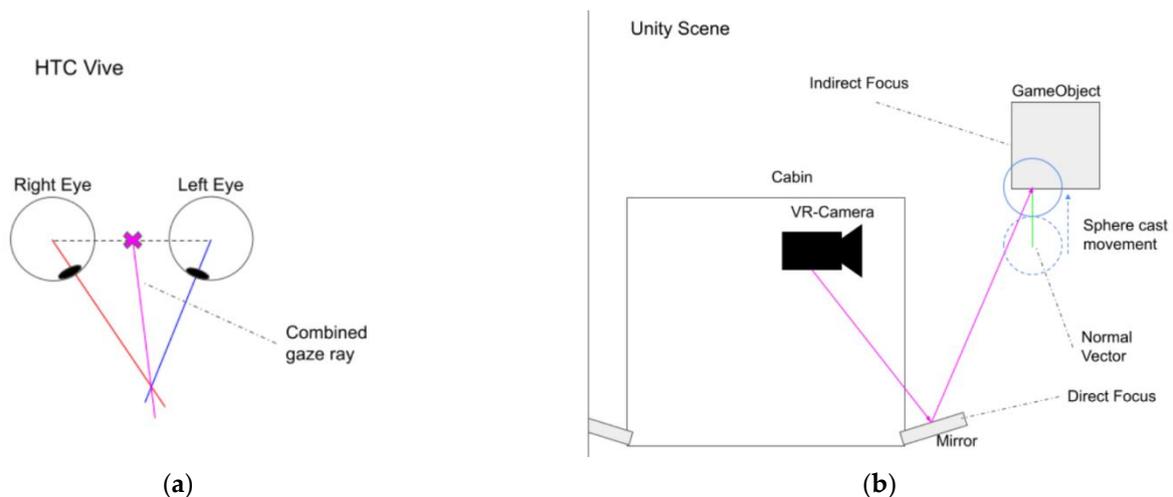


Figure 11. Object detection schematic (a) the combined gaze ray, (b) detect which object is gazed through the mirror.

With the gaze data, besides identifying the objects the driver is looking at, it is also possible to determine in which temporal order the driver picks up internal and external information for the truck docking process. As mentioned before, eye-tracking is also used to improve the simulator performance by dynamically toggling the mirrors and the camera used by the HMI. A Gaze-contingency-system is also controlled by eye-tracking that switches the texture of non-focused windows to solid black. This system aims to reduce the peripheral optical flow reducing nausea in the user [30].

Hand tracking is realized with the help of the Leap motion controller. The user's hand motions and gestures are captured and simultaneously mapped onto the virtual hands of the driver. In this way, the user controls the virtual hands inside the simulator and interacts with different objects, like the steering wheel or the HMI (see Figure 12). The properties of the virtual hands can be adjusted according to the user's needs (e.g., skin colour, arm length).



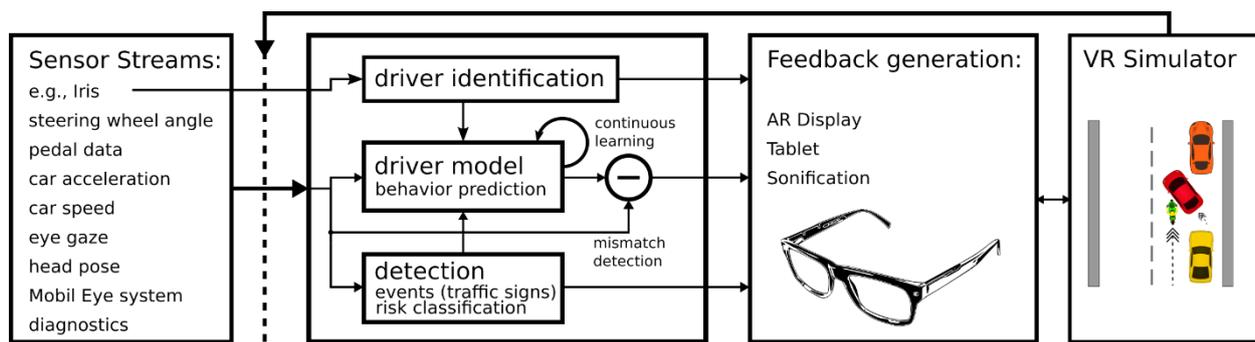
**Figure 12.** (a) Driver interacting with the steering wheel, (b) or with elements of the HMI.

### 2.3.5. Data Recorder and Player

Another interesting functionality of the VR-simulator software is the possibility to record and playback driving sessions. Concretely, it is possible to start the recording of a time-stamped list of sample values in a JSON file or the utterances captured by the microphone built in the HMI in an audio file at any time of a driving session. Specifically, in the JSON file the following data is recorded: (i) the information provided by the MATLAB/Simulink process (e.g., steering wheel angle, trailer's position and rotation), (ii) the eye-tracking data (e.g., pupil diameter, the direction of the gaze ray, as well as the list of focussed objects) and (iii) the user's head position and orientation. The recorded files can be used later for two different purposes: First, the recorded files can be loaded and played back in a unity program allowing the observation of the driver performance. The second purpose is to use the information stored in the JSON file as a dataset that can be applied for machine-learning research, e.g., to learn expert drivers' truck docking behaviour from the recorded simulator data.

### 3. Driver Behaviour Analysis

The VISTA VR-Simulator Platform is well suited to perform research on ADAS systems. One important goal of ADAS is to provide individualised and anticipative feedback that minimizes driver distraction and provides only situation-specific, highly relevant information. The system should adapt to the driver based on previous behaviour using learning techniques, e.g., machine learning. If the trained system detects unexpected driver behaviour in a familiar situation, it can trigger warnings or supportive feedback. In the context of the truck-docking process, an ML model was trained to predict the driver's expertise on-the-fly within a certain time-frame (see Section 3.3). If a situation of low expertise is detected, the system can provide context-sensitive feedback on the HMI to ease the truck docking process. Figure 13 provides a general scheme of such an adaptive and individualized ADAS.



**Figure 13.** ADAS concept. Sensor data are collected, classified and predicted by an individual driver model. Based on the model output, feedback is generated, for example, if mismatches between sensor streams and model predictions are detected or if situations of low driver performance are detected. The VR-simulator will be used for data collection, rapid prototyping of feedback components and evaluation studies.

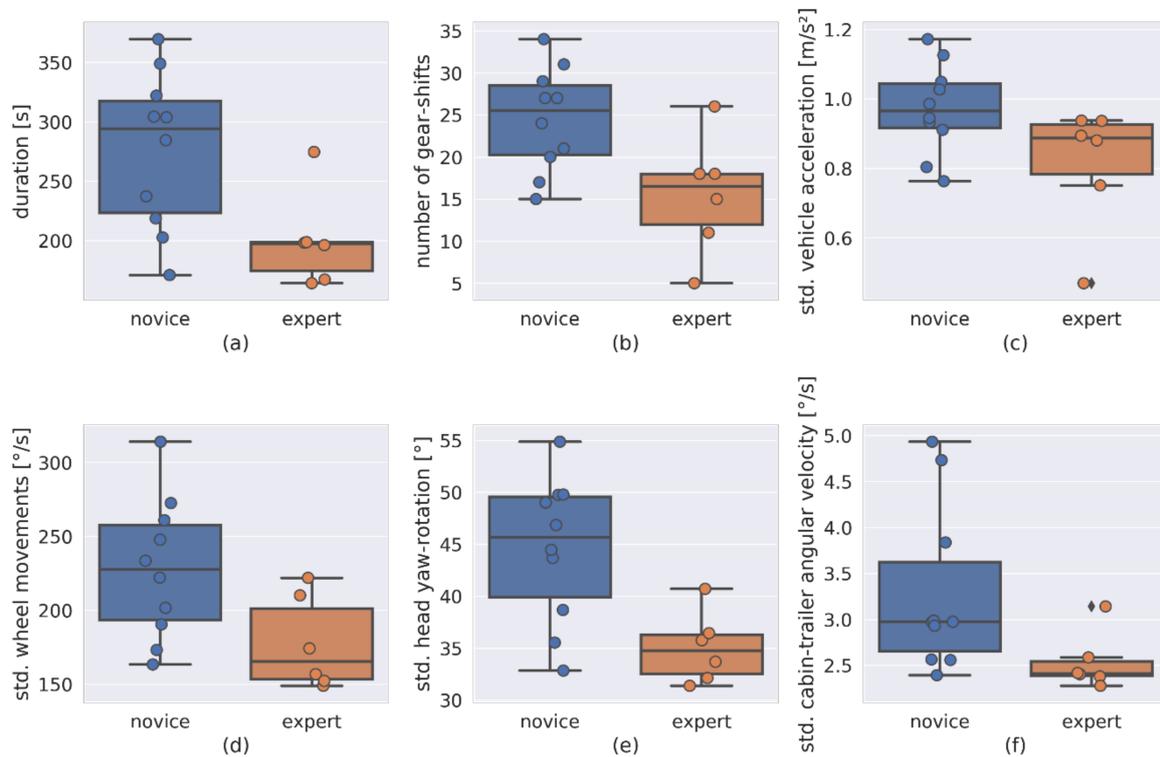
### 3.1. Driver Expert Interview

An interview was conducted with an expert driver to identify driver-performance indicators that help to distinguish between expert- and novice-drivers. The expert was asked to describe how experienced drivers would perform the docking manoeuvre step by step and where the differences to novice drivers become visible. According to the expert, the most important parameters to distinguish experienced from novice drivers are: (i) the relaxedness of the driver; (ii) the amount of steering movements; (iii) the amount of braking; (iv) the engine-rpm; (v) the frequency of gaze-switching between side-view mirrors; (vi) the number of collisions; (vii) the final parking angle; (viii) the overall time required for docking and the number of aborted docking attempts. The simulator allows to directly measure the amount of steering movement, the amount of braking and gear shifts, gaze-changes and head-movements, the parking angle and the overall docking-time.

### 3.2. Visualization of Expertise Performance Indicators

The required dataset for machine learning was recorded using the simulator. One subject performed one warm-up trial (excluded) and thereafter several training-trials ( $n = 16$ ) of the docking manoeuvre. For each trial, the subject provided a self-rating of the expertise level on a scale from one (worst) to five (best). The subject performed the docking task over the time span of a week, steadily improving in performance until a level comparable to experienced drivers was reached.

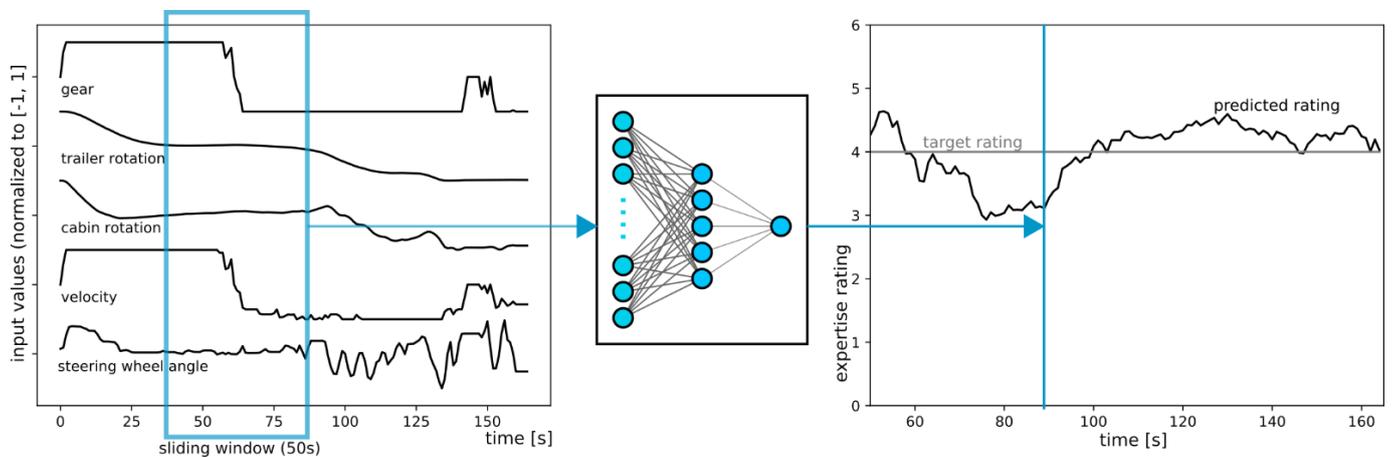
The recorded dataset was analysed with respect to several performance indicators extracted from the expert interview (Section 3.1). Trials were split into expert- and novice trials: all trials with a self-rating above 3 were tagged as expert trials, and the remaining were tagged as novice trials. Figure 14 shows that the analyzed performance indicators (docking task duration, number of gear shifts, standard deviation of vehicle acceleration, standard deviation of steering wheel movements and standard deviation of yaw head-rotation) indeed seem to correlate with the expertise level, visible as non-overlapping boxes in the box-plots. Expert trials show a shorter task duration, fewer gear shifts and a lower standard deviation in the remaining performance indicators. Further statistical analysis was not performed because of the comparatively small dataset.



**Figure 14.** The figure shows pronounced differences in (a) task duration, (b) number of gear shifts, (c) vehicle acceleration changes, (d) steering wheel movements, (e) head yaw-rotation and (f) trailer-cabin angular changes for two selected expertise levels (novice and expert).

### 3.3. Expertise Estimation Using Machine Learning

As shown in the previous section, there is a strong correlation between the expertise level of a driver and a number of performance indicators. Therefore, a small neural network should be able to estimate the expertise of a complete docking manoeuvre. Here, the question asked was whether such a network can also estimate the driver performance given a much shorter time-window from an ongoing docking manoeuvre. Such a network could then be used to inform a feedback component to provide support within a reasonable timeframe (see Figure 15).

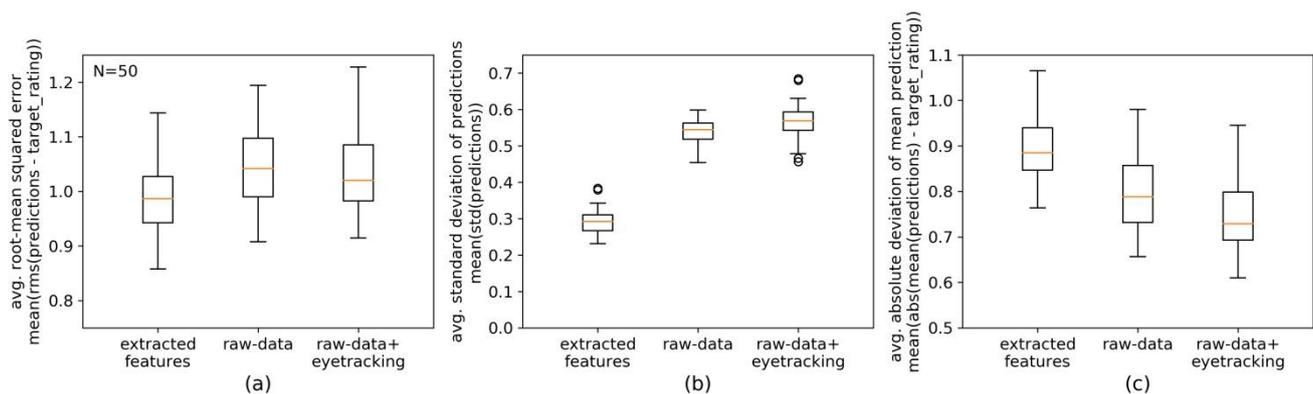


**Figure 15.** Driver expertise prediction. A neural network estimates the current expertise of the driver. Observable values like the steering wheel angle, the truck velocity and the current gear are aggregated over a sliding window (width = 50 s) and fed into the network. The network output provides an estimation of the current driver expertise within this time frame.

To test the idea, a multilayer perceptron was trained on the regression task to estimate the driver's expertise given only the following locally observable values: the steering wheel angle, the cabin-trailer angle, the truck velocity, the current gear, the drivers head yaw-rotation and optionally a list of gazed objects within the time-frame (acquired using eye-tracking, see Section 2.3.5). Locally observable values are comparatively easy to measure, are readily available in real vehicles (e.g., from the CAN-Bus) and do not require external cameras and sophisticated computer vision.

**Dataset pre-processing:** Each stream of local data in the dataset was individually scaled to the range  $[-1, 1]$ , to ease network training. Categorical like the current gear was one-hot encoded or multi-hot encoded (gazed objects from eye-tracking). The values were then down-sampled from 90 to 1 Hz using a block-wise average. It was found that using the block-wise average method gave slightly better results than another down sampling method (two-stage downsampling with an order 8 Chebyshev type I filter, see python package `scipy.signal.decimate`). An explanation for this might be that the closely related moving average method is optimal for filtering white noise while preserving sharp edges in data streams [31].

**Sliding window:** A sliding window (width = 50 s) was moved frame-by-frame (1 Hz) over each trial of the down-sampled dataset. The data within each window were either further processed by extracting the performance indicators shown in Section 3.2 and Figure 14, or concatenated into a long input vector for the neural network ("raw data" in Figure 16). For example, 150 training samples can be extracted from a 200-s trial.



**Figure 16.** Results of neural network-based expertise estimation given three different inputs: 1. extracted features, 2. raw-data and 3. raw-data + gazed objects (eye-tracking). The boxplots show  $N = 50$  repetitions of leave-one-out cross-validation (see text for more details). (a) Average root-mean-squared error of predictions from the target rating. (b) Average standard deviation of predictions. (c) Average absolute deviation of the mean prediction from the target rating.

**Network layout and training:** A small multilayer perceptron with one hidden layer ( $n = 5$  neurons, ReLU activation function) and a single, linear output neuron was implemented using Keras, a high-level neural network API for Tensorflow [32]. The single output neuron will provide an expertise rating with the same scale as that used for the self-rating, see Section 3.2. Hence, the network was trained on the regression task of estimating the driver expertise within the sliding-window time-frame. The specific training setup was: Adam Optimizer using gradient clipping (`clipnorm = 1.0`); a batch size of 25 and a fixed number of 15 epochs. The mean-squared error was used as the loss function.

**Network evaluation:** The performance of a trained network can significantly fluctuate mainly for two reasons: 1. local minima encountered during training (depending on initial weight values) and 2. the specific split of samples into the training- and test-dataset. For small datasets, a random splitting can be either advantageous or "unfair"; for example, if there are many difficult samples found in the test-dataset. To reduce such problems,  $n$ -fold cross validation is often used [33]. Here, leave-one-out cross-validation was used: one trial was excluded from the training-dataset respectively, until each trial was used

as the test-dataset once. The final result of leave-one-out cross-validation is the average mean-squared error over all the individual test-datasets.

### 3.4. First Evaluation of the Behaviour Analysis Module

Three different variants were tested for expertise-estimation using a neural network. They differed in the way the input values were presented to the neural network. Variant 1 used the performance indicators described in Section 3.2 as input features. Variant 2 used the “raw” data streams as described in Section 3.2, concatenated into a large, one-dimensional input vector. Variant 3 added a list of gazed objects (multi-hot encoded) to variant 2. To get a fine-grained insight into the effects of the different variants, a repetition of the cross-validation procedure described above was performed 50 times. The resulting boxplots (see Figure 16) indicate no substantial differences regarding the average root-mean-squared-error (RMSE) between the three variants, with variant 1 having a slightly lower error on average. The network can achieve almost the same RMSE given raw data streams, but the standard deviation of the predictions increased. Contrary to expectations, adding eye-tracking data did not improve the RMSE. A reason for this might be the dataset: given a very small dataset, adding more potentially helpful features might degrade network performance due to overfitting. This shows once again that machine-learning using small datasets is a challenging task. In summary, for all variants, the average RMSE is around one, meaning that the expertise estimation—even with this small dataset—can be used to trigger supportive feedback in an ADAS, e.g., if the estimation falls below a certain threshold.

## 4. Outcomes and Conclusions

This paper presented the VISTA-Sim, a platform that uses a VR-simulator and AI technologies to develop and evaluate personalized driver assistance. VISTA-Sim provides a holistic approach that allows: (i) the development of driver assistance concepts; (ii) the simulation of scenarios in which driver assistance is used; (iii) the training of driving skills in specific scenarios; (iv) the recording of multimodal interaction data from driving sessions; (v) the evaluation of driver assistance concepts and driver performances; (vi) the automatic learning from multimodal driver data; and (vii) the development of driver models capable of effectively describing different human drivers. This approach was realized by developing a custom platform and consequently benefits from the possibility to easily extend it through the integration of new modules, functionalities and scenarios. As such, this paper demonstrated the exploration of this approach within a very particular and controlled scenario in which truck drivers perform the rearward docking manoeuvre on logistics areas towards loading bays. This scenario offers ideal conditions for the exploration of this approach because it presents reduced levels of complexity when compared with other typical ADAS scenarios. A thorough search of the relevant literature yielded that this is the first attempt at using a VR-based holistic approach to develop driving assistance to this very specific scenario. Within the docking manoeuvre scenario, the potential of this platform was explored to allow the development of personalized driving assistance by providing means to learn from multimodal driver data, optimize training environments and ultimately guarantee that the automated driving support is effectively accepted and trusted. Additionally, it was reported that the conduction of a preliminary test in which the scenario of a docking manoeuvre was explored to record driver performances which were then used by the behaviour analysis module. The preliminary results indicate that the behaviour analysis module was able to identify the expertise level of each performance. This suggests that an improved iteration of the platform has the potential to perform a real-time and continuous evaluation of the driver’s expertise and adapt the driver assistance HMI to be personalized to the current needs of the driver. More concretely, these promising first results suggest that, despite the inherent technical challenges, this platform can evolve to be capable of automatically learning from small datasets, deal with label noise coming from a subjective rating, and ultimately learn and adapt ADAS systems to increase their acceptance and trustiness.

Finally, it is important to highlight that the exploration of a VR-simulator brings significant advantages when compared with the real-world, for instance, the data collection in a VR-simulator is a practical and simple process in which driving situations can be repeated in a very controlled way, guaranteeing access to clean multimodal driver data. Additionally, since a variety of tridimensional data (e.g., vehicle or gazed objects) is being recorded, these data can be at any moment revisited and analysed from different perspectives and for different purposes. It is envisioned that by exploring the data collection benefits offered by the VR-simulator and by employing machine-learning techniques, an opportunity is presented to endeavour through the unexplored field of building driver models using data-driven methods and ultimately pursue the creation of more accepted and trusted driving assistance systems.

## 5. Future Work

Our simulator is a constantly evolving platform for ADAS research that is continuously adapted to new technologies. As such, it offers opportunities for new research areas and technical improvements. Below is provided a non-exhaustive list of potential future work.

**Machine Learning improvements:** The machine learning results presented in this paper indicate that even small neural networks can estimate the driver's expertise. Therefore, efforts will be directed toward the extension of the ML approach to behaviour prediction (what will the driver do next?) and continuous learning concepts. The current generation of neural networks are often static and do not change once the training process has been completed. Therefore, they cannot adapt and change over time, contrary to the networks of living neurons found in human and animal brains. A new trend in machine learning, called "lifelong learning", strives to overcome this limitation [34]. Such an approach would be ideally suited to the task of building an ADAS that continuously learns from and adapts to the individual user. However, the implementation of "lifelong learning" poses several challenges, including effects like "catastrophic forgetting", where new, incoming training data tend to overwrite previous knowledge, causing a sudden drop in network performance [34,35]. Furthermore, it is planned to perform careful evaluation studies to test the functionality and the user acceptance of such a self-adapting system. For the behaviour analysis module, it is planned to use not only local variables to estimate the driver expertise. Training the ML model using global data (e.g., absolute trailer and cabin positions and angles with respect to an external world-coordinate system) will improve expertise estimation, as preliminary experiments with the dataset from Section 3.4 indicates. Global data are easily available within the VR-Simulator, but are much harder to acquire in real-world scenarios using, e.g., stationary cameras. A real-world tracking system is currently implemented in the VISTA project [9].

**Trust in algorithms:** Trust in algorithms and systems is another important future research topic. For example, an ADAS geared towards autonomous driving needs to explain its decisions to strengthen the trust of the user in the system. The effectiveness of an ADAS to establish trust and to provide proper, individualized feedback needs to be evaluated using psychometric tools (e.g., questionnaires) and physiological markers. Another method to increase acceptance and trust is to give the user the freedom of choice, for example, concerning the feedback modalities of the HMI [36] or the way the system adapts to the user. For example, the user should have the choice to use a static, non-adapting system, should be able to select an adaptation speed and should be able to reset the system to an initial state. Therefore, the system should be highly customizable and provide a way to create and select individual user profiles.

The VR-Simulator provides another interesting opportunity: Within VR, it is possible to test simulated AR devices. In that sense, the use of the VR-Simulator to test and evaluate hardware aspects of AR devices is intended (e.g., field of view) as are different concepts for AR-based HMI. This will allow us to get a better understanding of best practices regarding feedback generation and interaction with future and upcoming AR-Glasses.

VR-Simulator Rendering improvements: Existing high-end VR headsets (e.g., Varjo 3) and future, upcoming headsets will have a much higher resolution than the headset currently used (HTC VIVE Pro). Therefore, this headset will be even more demanding for rendering the VR world. Especially regarding the current approach for calculating reflections for the mirrors (implemented through reflection probes). Future work should explore alternative solutions for implementing the vehicle mirrors, for example using Ray Tracing or specialized shaders. Additionally, the overall performance can be improved using foveated rendering.

**Author Contributions:** Conceptualization and methodology, P.R., K.E. and A.F.K.; software, P.R., A.F.K., P.M., M.-A.B., J.O., K.K. and J.v.K.; validation, P.R., K.E. and A.F.K.; formal analysis, P.R., K.E. and A.F.K.; investigation, P.R., K.E. and A.F.K.; resources, P.R., K.E. and A.F.K.; writing—original draft preparation, P.R., K.E., A.F.K., P.M., M.-A.B., J.O. and K.K.; writing—review and editing, P.R., K.E., A.F.K., P.M., M.-A.B., J.O., K.K., J.B. and C.R.; supervision, K.E.; project administration, J.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is supported within the Interreg V-A VISTA project, co-financed by the European Union via the INTERREG Deutschland-Nederland programme, the Ministerium für Wirtschaft, Innovation, Digitalisierung und Energie of Nordrhein-Westfalen and the Province of Gelderland.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The behaviour-analysis dataset can be downloaded from: <https://hochschule-rhein-waal.sciebo.de/s/IFTgQWH461cOk72> (accessed on 23 August 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Liu, S. Global AR/VR Automotive Spending 2017 and 2025. Available online: <https://www.statista.com/statistics/828499/world-ar-vr-automotive-spending/> (accessed on 28 June 2021).
- Choi, S.H.; Cheung, H.H. A Versatile Virtual Prototyping System for Rapid Product Development. *Comput. Ind.* **2008**, *59*, 477–488. [[CrossRef](#)]
- Lawson, G.; Salanitri, D.; Waterfield, B. Future Directions for the Development of Virtual Reality within an Automotive Manufacturer. *Appl. Ergon.* **2016**, *53*, 323–330. [[CrossRef](#)] [[PubMed](#)]
- Taheri, S.M.; Matsushita, K.; Sasaki, M. Virtual Reality Driving Simulation for Measuring Driver Behavior and Characteristics. *J. Transp. Technol.* **2017**, *7*, 123. [[CrossRef](#)]
- Gunning, D.; Aha, D. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Mag.* **2019**, *40*, 44–58. [[CrossRef](#)]
- Sanneman, L.; Shah, J.A. A Situation Awareness-Based Framework for Design and Evaluation of Explainable AI. In *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*; Calvaresi, D., Najjar, A., Winikoff, M., Främling, K., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 94–110.
- Nam, C.S.; Lyons, J.B. *Trust in Human-Robot Interaction*; Academic Press: Cambridge, MA, USA, 2020; ISBN 978-0-12-819473-7.
- Kusumakar, R.; Buning, L.; Rieck, F.; Schuur, P.; Tillema, F. INTRALOG—Intelligent Autonomous Truck Applications in Logistics; Single and Double Articulated Autonomous Rearward Docking on DCs. *IET Intell. Transp. Syst.* **2018**, *12*, 1045–1052. [[CrossRef](#)]
- Benders, J. VISTA—Towards Damage-Free and Time-Accurate Truck Docking. Available online: <https://vistaproject.eu/> (accessed on 28 June 2021).
- Ihemedu-Steinke, Q.C.; Erbach, R.; Halady, P.; Meixner, G.; Weber, M. Virtual Reality Driving Simulator Based on Head-Mounted Displays. In *Automotive User Interfaces: Creating Interactive Experiences in the Car*; Meixner, G., Müller, C., Eds.; Human-Computer Interaction Series; Springer International Publishing: Cham, Switzerland, 2017; pp. 401–428, ISBN 978-3-319-49448-7.
- Körber, M.; Baseler, E.; Bengler, K. Introduction Matters: Manipulating Trust in Automation and Reliance in Automated Driving. *Appl. Ergon.* **2018**, *66*, 18–31. [[CrossRef](#)] [[PubMed](#)]
- Zahabi, M.; Razak, A.M.A.; Shortz, A.E.; Mehta, R.K.; Manser, M. Evaluating Advanced Driver-Assistance System Trainings Using Driver Performance, Attention Allocation, and Neural Efficiency Measures. *Appl. Ergon.* **2020**, *84*, 103036. [[CrossRef](#)] [[PubMed](#)]
- Sportillo, D.; Paljic, A.; Ojeda, L. Get Ready for Automated Driving Using Virtual Reality. *Accid. Anal. Prev.* **2018**, *118*, 102–113. [[CrossRef](#)] [[PubMed](#)]
- Diederichs, F.; Niehaus, F.; Hees, L. Guerilla Evaluation of Truck HMI with VR. In *Virtual, Augmented and Mixed Reality. Design and Interaction*; Chen, J.Y.C., Fragomeni, G., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 3–17.
- Lin, N.; Zong, C.; Tomizuka, M.; Song, P.; Zhang, Z.; Li, G. An Overview on Study of Identification of Driver Behavior Characteristics for Automotive Control. *Math. Probl. Eng.* **2014**, *2014*, e569109. [[CrossRef](#)]

16. Wang, W.; Xi, J.; Chen, H. Modeling and Recognizing Driver Behavior Based on Driving Data: A Survey. *Math. Probl. Eng.* **2014**, *2014*, e245641. [[CrossRef](#)]
17. Hasenjäger, M.; Heckmann, M.; Wersing, H. A Survey of Personalization for Advanced Driver Assistance Systems. *IEEE Trans. Intell. Veh.* **2020**, *5*, 335–344. [[CrossRef](#)]
18. Li, A.; Jiang, H.; Zhou, J.; Zhou, X. Implementation of Human-Like Driver Model Based on Recurrent Neural Networks. *IEEE Access* **2019**, *7*, 98094–98106. [[CrossRef](#)]
19. Sallab, A.E.; Abdou, M.; Perot, E.; Yogamani, S. Deep Reinforcement Learning Framework for Autonomous Driving. *Electron. Imaging* **2017**, *2017*, 70–76. [[CrossRef](#)]
20. Darwish, A.; Steinhauer, H.J. *Learning Individual Driver's Mental Models Using POMDPs and BToM*; IOS Press: Amsterdam, The Netherlands, 2020; pp. 51–60.
21. Xu, J.; Min, J.; Hu, J. Real-Time Eye Tracking for the Assessment of Driver Fatigue. *Healthc Technol. Lett.* **2018**, *5*, 54–58. [[CrossRef](#)] [[PubMed](#)]
22. MathWorks Simulink—Simulation und Model-Based Design. Available online: <https://de.mathworks.com/products/simulink.html> (accessed on 25 June 2021).
23. Unity, U. Unity 3D Platform. Available online: <https://unity.com/> (accessed on 9 June 2021).
24. HTC Corporation VIVE Pro Eye. Available online: <https://www.vive.com/eu/product/vive-pro-eye/overview/> (accessed on 9 June 2021).
25. Logitech Logitech G29 Steering Wheels & Pedals. Available online: <https://www.logitechg.com/en-eu/products/driving/driving-force-racing-wheel.html> (accessed on 9 June 2021).
26. Ultraleap Leap Motion Controller. Available online: <https://www.ultraleap.com/> (accessed on 9 June 2021).
27. Devasia, D. *Motion Planning with Obstacle Avoidance for Autonomous Docking of Single Articulated Vehicles*; HAN University of Applied Sciences: Arnhem, The Netherlands, 2019.
28. Kannan, M. Automated Docking Maneuvering of an Articulated Vehicle in the Presence of Obstacles. Ph.D. Thesis, České Vysoké Učení Technické v Praze, Prague, Czech Republic, 2021.
29. Kural, K.; Nijmeijer, H.; Besselink, I.J.M. *Dynamics and Control Analysis of High Capacity Vehicles for Europe*; Technische Universiteit Eindhoven: Eindhoven, The Netherlands, 2019.
30. Hettinger, L.J.; Riccio, G.E. Visually Induced Motion Sickness in Virtual Environments. *Presence Teleoper Virtual Env.* **1992**, *1*, 306–310. [[CrossRef](#)]
31. Smith, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing*; California Technical Publishing: San Diego, CA, USA, 1997; ISBN 978-0-9660176-3-2.
32. Chollet, F. Keras GitHub. Available online: <https://github.com/fchollet/keras>. (accessed on 26 June 2021).
33. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer Series in Statistics; Springer: New York, NY, USA, 2009; ISBN 978-0-387-84857-0.
34. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual Lifelong Learning with Neural Networks: A Review. *Neural Netw.* **2019**, *113*, 54–71. [[CrossRef](#)] [[PubMed](#)]
35. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.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](#)]
36. Coelho, J.; Duarte, C. The Contribution of Multimodal Adaptation Techniques to the GUIDE Interface. In *Universal Access in Human-Computer Interaction. Design for All and eInclusion*; Stephanidis, C., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 337–346.