Integration of Driving- and Trafficsimulation

Martin Scheuchenpflug

BACHELORARBEIT

eingereicht am

Automotive computing

Universal Computing

in Hagenberg

im Juni 2024

Advisor:

FH-Prof. Dr. Gerald Ostermayer

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere. This printed copy is identical to the submitted electronic version.

Hagenberg, June 25, 2024

Martin Scheuchenpflug

Contents

Preface

Abstract

This should be a 1-page (maximum) summary of your work in English.

Kurzfassung

An dieser Stelle steht eine Zusammenfassung der Arbeit, Umfang max. 1 Seite. ...

Chapter 1

Introduction

1.1 Motivation

1.1.1 Driving simulation

The use of driving simulators is common practice among researchers for studying driving behaviors in scenarios that may include dangerous and unlikely situations[15, 22]. The utilization of driving simulators is primarily driven by the significant reduction in time and financial resources for engineers and developers, as evidenced by reference[15]. Additionally, it facilitates expeditious research and evaluation of innovative human-machine-interface designs (HMI) within the automotive industry[13]. One such simulator represents the “me”-perspective by focusing the simulation on the vehicle that is driven by a human in the simulator, thereby ensuring that this vehicle is modeled with great precision. One illustrative example of a well-known driving simulator is CARLA, which is an open-source, free-to-use driving simulator with a feature-rich Python API[5].

1.1.2 Microscopic traffic simulation

The use of microscopic traffic simulators is also a common methodology among researchers studying the dynamics of vehicular traffic, particularly in the context of intelligent transportation systems (ITS)[16]. One such microscopic traffic simulator employs (among other models like: lane-change, fuel-consumption, . . .) car-following models to regulate the acceleration and, consequently, the movement of each vehicle[19]. A variety of car-following models are available to use for modelling different types of vehicles. One notable example of such a model is the IDM (Intelligent Driver Model) (see Section 2.1.3) described by Springer et al., which assembles one of the simplest complete and accident free models, that produces plausible acceleration profiles[19]. Moreover, as described by Treiber et al. these models are derived from a set of assumptions about real drivers such as keeping a “safe distance” from the leeding vehicle, driving at a desired speed or prefaring acceleration to be within a comfortable range. As all the mentioned Parameters are like sensor-values to a modern ACC (adaptive cruse control) System, models operating that way typically produce driving behaviors similar to those observed by the use of ACC systems[19].

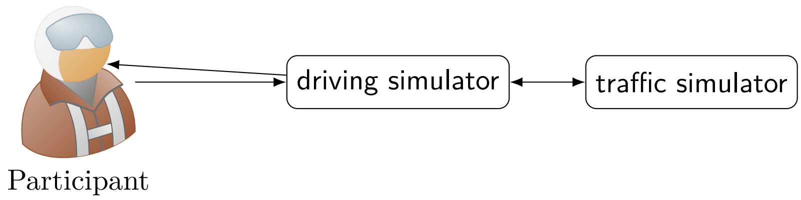

Figure 1.1: Integration architecture

1.1.3 Integrated simulation environment

With the goal in mind of investigating the impact of human drivers on traffic flow, one valid solution is to model the imperfections of a human driver. Using so called meta-models on top of typical vehicle-following models which modify the behavior to closely match a real human driver is common practice. Example for meta-models are for example the human driver model (HDC)[20] or the extended human driver model (EHDC)[9]. An additional potential methodology for the observation of human behavior is to engage directly in a scenario, which offers the benefit of eliminating the necessity for a model of the human driver. As a result of the direct participation of the human in the scenario, the impact of that behavior on the overall traffic flow can be observed. The aforementioned "me"-perspective is transformed into a holistic "we"-perspective through the utilization of an integrated simulation environment. This kind of integration has proven very useful[12].

In Essence there are two types of integrations:

- **Offline integration:** For this integration a driving simulator and a traffic simulator are used seperatly. For example one could build the same scenario in both simulators and a human driver could participate in the scenario inside the driving simulator. Afterward the behavior of the human driver is recorded and imported into the traffic simulator, so the impact of the recorded behavior can be observed. The key disadvantages of this integration are, that it lacks the possibility of observing the impact of the driving behavior in real time and the model-controlled vehicles are not able to adapt to the human driver's behavior during the simulation.
- **Online integration:** The aforementioned disadvantages are rendered obsolete by implementation of an online integration. The most straightforward structure for such an integration would be to have the participant interact with the driving simulator, which is then synchronized to the traffic simulator. This can be achieved by connecting a driving simulator with a microscopic traffic simulator, as illustrated in Figure1.1. By running the simulators in sync, it is possible for the simulation controlled (by the traffic simulator) vehicles to react to the driving actions of the vehicle controlled by a human driver.

The remaining work focuses on the online integration of the simulators, unless otherwise specified.

1.2 Objective of the work

In this thesis, an integrated Simulation Environment is proposed, consisting of the driving simulator CARLA[5] and the microscopic traffic simulator TraffSim[1, 2]. The primary objective of this integration is to achieve a robust and synchronized environment, where the real-time interactions between human-driven and vehicular-follow-model-controlled vehicles unfold seamlessly.

The key challenges of the proposed integrated environment are:

- **Synchronisation of integration components:** One of the primary objectives of this work is to ensure seamless synchronisation between the driving and traffic simulators. This synchronisation involves real-time data exchange, particularly for online integration, where the actions of a human driver in the driving simulator should immediately impact vehicle behaviours in the traffic simulator. A significant challenge in this regard is the minimisation of latency, which could cause inconsistent or delayed responses between simulators.
- **Road network compatibility and flexibility:** Synchronizing road networks across the simulators is essential for a coherent simulation environment, as inconsistencies can lead to inaccuracies in traffic flow or navigation. Therefore, a procedure for exporting, importing and exchaning road networks between the simulators was implemented by the use of the OpenDrive**??** Standard.
- **Handling human driving behavior:** Human drivers often exhibit unpredictable behaviors, creating a need for adaptable models that can respond effectively to erratic or non-standard actions. This objective was partially archived by extrapolating the acceleration and speed of human-driven vehicle, for further use, this objective has to be readressed. One potential solution for this problem could be to use a vehicle-following, which models a human driver[9], to predict the next action taken by the human in charge of the vehicle.

1.3 Structure

This thesis is structured into multiple chapters, each addressing discrete aspects of the integration of driving and traffic simulators and the associated challenges.

The initial chapter, entitled “Introduction”, provides an overview of the subject, outlining the historical development of driving and traffic simulation, the individual functionalities of each, and the rationale for integrating them into a unified environment. Furthermore, it delineates the objectives and scope of the work.

The second chapter, entitled “Related Work”, presents a review of existing research on driving and traffic simulation systems. It focuses on previously explored methods for integration, identifying both successful techniques and common obstacles encountered.

The third chapter, entitled “Methods”, provides a detailed account of the architectural design of the integrated simulation environment. It describes the technical framework and design decisions that have been taken in order to achieve synchronisation of the simulators, to facilitate real-time data exchange and to ensure compatibility between simulator platforms.

The fourth chapter, Results, presents the outcomes of the implemented integration, including a discussion of performance metrics, an analysis of the observed behaviours of the simulation components, and an evaluation of the solutions proposed for specific integration challenges.

The fifth chapter, entitled “Future Work”, identifies areas for further research and development. It discusses potential improvements in integration techniques and considers ways of expanding the functionality and realism of the integrated simulation environment. The thesis concludes with a summary and reflections on the insights gained through the integration process, its potential implications and the contributions made to the field of simulation in intelligent transportation systems.

Chapter 2

Related work

In the following chapter provides an overview of research and state-of-the-art in the domain of modelling and simulation of traffic systems. Section 2.1 examines foundational studies that have shaped current understanding of modelling traffic systems, focusing on the methodologies and findings that are most relevant to this thesis. Section 2.2 focuses on studies that are more closely related to the proposed simulation environment and the challenges that had to be solved in order to create one such integration. Finally, Section 2.3 summarises the findings of the literature review and identifies the used solutions for upcoming challenges encountered along the way of developing the integrated Simulation environment. By mapping out the state of the art, this chapter establishes the context for the subsequent discussion of the proposed approach and methodology in Chapter 3.

2.1 Foundational studies

The field of traffic science is divisible into multiple different sub-categories by the time span of the observed events. As described by Treiber et al. [19] there are Vehicle dynamics, Traffic flow dynamics and Transportation planning as the three major topics(see Table 2.1). These fields are divisible even further as stated in table 2.1. The proposed simulation environment would fall into the category of traffic flow dynamics using microscopic car following models. According to Treiber et al. this kind of simulation environments are best used to model reaction times, time gaps and the acceleration/breaking behaviors of vehicles in a traffic scenario. Driving behavior of vehicles in traffic flow dynamics are usually described by vehicle following (VF) models, which will be discussed further in the following sections.

2.1.1 Gazis-Herman-Rothery model

Vehicle following models have been researched for more than 70 years [11] and many such models have been developed since. One of the first widely known VF models is the Gazis-Herman-Rothery (GHR) model (see Eq 2.1), which described the acceleration of a vehicle n in a driving scenario with respect to the difference in speed Δv to the leading vehicle $n-1$ and the distance to the leading vehicle Δx, at a point earlier in

Time scale	Field	Models	Aspects of traffic (examples)
$\leq 0.1s$	Vehicle dynamics	Sub-microscopic	Control of engine and breaks
1 s		Car-following	Reaction time, time gap
10 s	Traffic flow dynamics	modells	Acceleration and deceleration
1 min		Macroscopic	Cycle period of traffic lights
10 min		models	Stop-and-go waves
1 h		Route	Peak hour
1 day	Transportation planning	assignment	Daily demand pattern
1 year		traffic demand	Building/changing infrastructure
5 years		Statistics age	Socioeconomic structure
50 years		pyramid	Demographic change

Table 2.1: Delimitation of traffic flow dynamics from vehicular dynamics and transportation planning[19]

time, with T being the reaction time of the driver[4].

$$a_n(t) = cv_n^m(t)\frac{\Delta v(t-T)}{\Delta x^{-1}(t-T)} \tag{2.1}$$

2.1.2 Gipps' model

Gipps states in the paper proposing his own VF model (Gipps´ model), that most VF models up until then (1981) have generally been in the form of EQ 2.2, where $\tau = T$. One example for this general form is the GHR model, some more of those models are found in publications dating back to that time: [3, 8, 10].

$$a_n(t+\tau) = l_n\frac{[v_{n-1} - v_n]^k}{[x_{n-1} - x_n]^m} \tag{2.2}$$

As pointed out by Philip A. Seddon, the fact, that the time interval between subsequent calculations was given by the reaction time was an undesired characteristic of these models[14], which could be overcome by storing a considerable amount of historical data, which was undesired at the time. The second pain point of these models, by today's standards maybe the more relevant point, was the existence of parameters l_n,k,m (EQ 2.2), that have no identifiable connection to driver or vehicle characteristics[6].

To address these deficits, Gipps proposed a new vehicle following model describing the speed of a vehicle at a point in time, in contrast to describing the acceleration of a vehicle at a point in time. The following form of the Gipps‘ model is not the form of the original publication, but the form from [19], which introduces a save speed v_{save} for simplification. As stated by Treiber et al. the model is conseptually unchanged. The GHR model calculates the speed of a vehicle with respect to the desired acceleration a, deceleration b, Desired speed v_0 and minimum distance s_0.

$$v_{save}(s, v_l) = -b * \Delta t + \sqrt{b^2 \Delta t^2 - v_l^2 + 2b(s - s_0)} \tag{2.3}$$

$$v(t + \Delta t) = min[v + a\Delta t, v_0, v_{save}(s, v_l)] \tag{2.4}$$

With the concept of the save speed depending on the distance to the front vehicle s and its speed v_l, the gipps‘ model assembles one of the simplest complete and accident-free models possible. The accident-free characteristic of the model is guaranteed with the assumptions that deceleration and reaction times are constant [19]. In essence the gipps‘ model chooses the minimum between the desired speed, the safe speed or the speed that max acceleration would result in the next time step. Problem with this model is, that as stated by Treiber et al. it produces an unrealistic acceleration profile.

2.1.3 Intelligent driver model

The time-continuous Intelligent driver model produces a realistic acceleration profile, since one of the requirements for forming the IDM is that the acceleration function $\dot{v}(s, v, v_l)$ is continuously differentiable in every three variables and thus producing smooth transitions between eg. breaking and acceleration phases. Another design criteria of the IDM is that the equilibrium distance[1]has to be larger than the “safe” distance $(v * T + v_0)$, with v being the current speed, T being the desired distance between vehicles in seconds and v_0 representing the “bumper-to-bumper” distance (min distance kept between vehicles), which ensures the accident-free characteristic of that model.

$$s^*(v, \Delta v) = s_0 + \max\left(0, vT + \frac{v\Delta v}{2\sqrt{ab}}\right) \tag{2.5}$$

$$\dot{v}(s, v, v_l) = a * \left[1 - \left(\frac{v}{v_0}\right)^\delta - \left(\frac{s^*(v, \Delta v)}{s}\right)^2\right] \tag{2.6}$$

The IDM (see EQ 2.6)is constructed from 2 pieces, the first part is comparing the current speed to the desired speed: $\left(\frac{v}{v_0}\right)^\delta$. The second part is comparing the current distance to the desired distance s^* (see EQ 2.5): $\left(\frac{s^*(v,\Delta v)}{s}\right)^2$ [17, 19].

The advantage of using an intuitive model like gipps‘ model or the IDM, is that they are easy to configure by some simple understandable parameters (including vehicle and driver specific numbers). For example the configuration of the IDM consists of six parameters controlling the behavior of vehicles, with default values used by TraffSim (see Table 2.2) [1, 2, 19]. The Implementation in TraffSim includes two additional parameters, determining the maximum possible acceleration a_{max} and deceleration b_{max}, which act as a hard limit for the acceleration function. The value of b_{max} for example models the physical limit of the breaks of a car.

[1]The Distance, that is required between vehicles to stay in a steady-state equilibrium, which means, in a homogenous convoy the distance between all vehicles and the speed is the same. Furthermore the acceleration has to be 0 for every vehicle to be in a steady-state eqilibrium. [17, 19]

[2]Dimensionless factor determining the rate at wich the vehicle‘s acceleration decreases as it approaches its desired velocity

Parameter	TraffSim default Value	Acceptable Range
Comfortable acceleration a	2m/s^2	$0 < a < a_{max}$
Comfortable deceleration b	-2m/s^2	$b_{max} < b < 0$
Desired Speed v_0	25m/s	$v_0 > 0$
Minimum bumper-to-bumper distance s_0	2m	$s_0 > 0$
Time gap T	1.5s	$T > 0$
Acceleration exponent δ [2]	4	$\delta > 0$

Table 2.2: Default Values for IDM

Figure 2.1: MovSim user interface [23]

2.1.4 Micro traffic simulation

The aforementioned vehicle following models are among others, implemented in a traffic simulator. Since one such simulator is a key building block of the desired simulation environment, the next section focuses on the micro traffic simulation environments available in literature. Special focus will be placed onto the micro traffic simulator TraffSim [1, 2], developed in house at Hagenberg.

The first notable example is MovSim [18], which is an open source, free to use traffic simulator developed by a team at TU Dresden. MovSim implements the IDM (see Section 2.1.3) among other vehicle following model which is responsable for the longitudinal movement of vehicles. For transversal movement of vehicles, MovSim utilizes the MOBIL model [7], which is a model for lane changing working alongside a variety of VF models. MovSim is primary source for sample implementations of VF models described in [19], a picture of the web based user interface is depicted in Figure 2.1.

2.2 Studies closely related to an integrated simulation environment

2.3 Summary of findings

As described by Treiber et al., both of these simulators are located in the field of traffic flow dynamics.

2.4 Microscopic traffic simulation

2.5 Driving simulation

Chapter 3

Methods

3.1 Architecture of the Integrated Simulation Environment

The proposed integration consists of the open-source Driving Simulator CARLA[5] and the microscopic traffic simulator TraffSim[1, 2], which was developed at the University of Applied Sciences Upper Austria. To connect the two simulators into one integrated environment, a gRPC[21] API was added to the microscopic traffic simulator, as well as a REST (Representational State Transfer) Endpoin. The gRPC-API is used to transfer data between the simulators during a running Simulation, as well as to transfer control signal (such as: Pausing the Simulation). The REST-Endpoint is used by the driving simulator to request the map used for the scenario. This map is requested in OpenDrive Format (see Section3.5).

3.1.1 Vehicles in TraffSim

In TraffSim there are a few different Types of Vehicles (see Figure

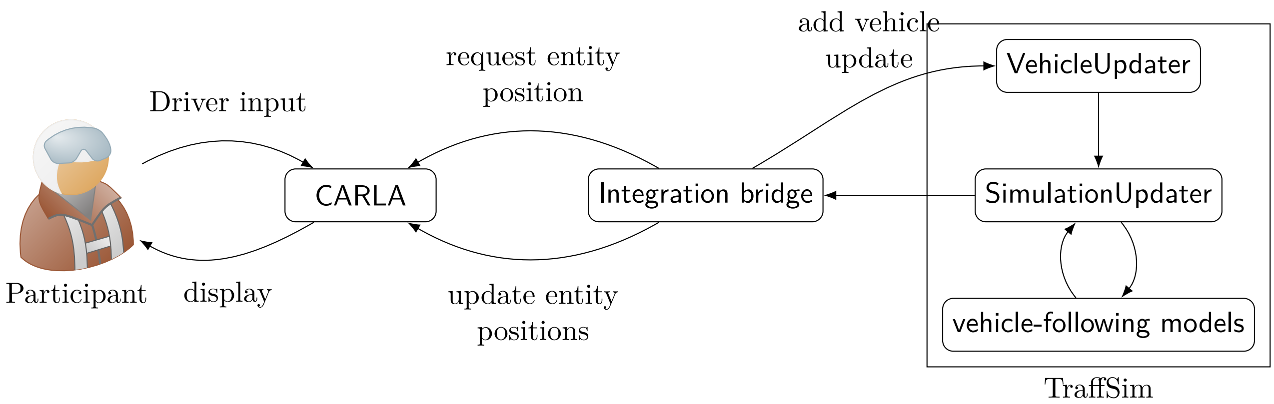

Figure 3.1: Data roundtrip

3.1.2 Simulation step synchronisation

The human driver employs the Carla interface to engage with the integrated simulation environment (see Figure3.1). It is essential that the driving behaviour of the simulation-controlled vehicle appears natural to the human driver. Therefore, the time between driver input and the visual output of the driving simulator must be minimised (data round-trip time). It is essential that the driver's input is accurately captured by Carla and that the human-driven vehicle is precisely simulated in order to create a realistic driving experience for the driver. As the two simulators are not synchronised with regard to their tick rates, the integration bridge requests the current position, speed and acceleration of the vehicle and a vehicle update is added to the update queue of the HumanDrivenVehicleUpdater within the traffic simulator. Upon calculation of the subsequent simulation step in the traffic simulator, all vehicle updates that have been queued are executed, thereby updating the human-driven vehicle. Therefore, the delta time between simulation steps should be selected as low as possible, in order to minimise the data round-trip time.

3.1.3 Real time simulation

although t must not be lower than the time it takes for the simulation step to be calculated T. To bind the simulation time to the real time, the easiest way would be to set the time step size to the time duration it took for the last simulation step to compute:

$$t_x = T_{x-1}$$

The problem with that approach is, that during a simulation in TraffSim it should be avoided to change the simulation step t, so another approach is used, by adding a delay after each simulation step to compensate for the difference between t and T. It's important for this technique to work, the condition $t \geq T$ has to be met. In every simulation step the added delay d has to be calculated and thus t can be static:

$$d_x = t - T_x$$

3.2 TraffSim API

To interact with the microscopic traffic simulator TraffSim, a gRpc[21] API was specified and developed, which allows the client to call the functions described in Listing3.1. The Api was designed to allow the client to control all the functions that are important during a simulation session and also to sync the two simulators.

Listing 3.1: TraffSim API proto

```
service TraffSimController {
    rpc requestVehicleData (ActiveSimulationRun)
        returns (VehicleData);
    rpc notifyFreeDrivingVehicle(FreeDrivingVehicleUpdateRequest)
        returns(VoidMessage){}
    rpc simulationAllowTswConnection (ActiveSimulationRun)
        returns (BoolMessage) {}
    rpc getAvailableFreeDrivingVehicles(ActiveSimulationRun)
```

```
        returns(AvailableFreeDrivingVehicles){}
    rpc getAvailableScenarios (VoidMessage)
        returns (AvailableScenarios){}
    rpc getOpenSimulations (VoidMessage)
        returns (ActiveSimulationRuns){}
    rpc getSimulationState (ActiveSimulationRun)
        returns (SimulationState){}
    rpc pauseSimulation (ActiveSimulationRun)
        returns (VoidMessage){}
    rpc startSimulation (ActiveSimulationRun)
        returns (VoidMessage){}
    rpc setSimulationResolution (SimulationResolutionRequest)
        returns (VoidMessage){}
    rpc setSimulationDelay (SimulationDelayRequest)
        returns (VoidMessage){}
}
```

3.3 Mapped dummy vehicles

In TraffSim, there are several different types of vehicles (see Figure).The human-driven vehicles are able to traverse the entire map and are not constrained to the road network, whereas the simulation-controlled vehicles are constrained to the road network.

In order for the vehicle-following models to react to the human driver, it is necessary to represent the human-driven vehicle on the “rail-like” road network. Dummy vehicles are placed on the road network at each time step to enable the model-controlled vehicles to interact with the human-controlled vehicle. A map-matching algorithm is employed to identify the locations where dummy vehicles should be placed.

3.3.1 Map matching

To determine the position, a dummy vehicle has to be placed on the road network, a map-matching algorithm is used.

3.4 Speed and Acceleration projection

In order to allow the vehicle-following models to work, the human driven vehicle‘s speed $\vec{v_h}$ and acceleration $\vec{a_h}$ need to be mapped onto the dummy vehicle. Therefore $\vec{a_h}$ and $\vec{v_h}$ are projected onto the tangent vector of the nearest road segment $\vec{R}$. Since a mapped vehicle is bound to the road network and thus traveling “on rails”, the speed and acceleration can be expressed as scalars (a_m and v_m), which act like the magnitude of the speed vector in the direction of the current Road segment. The idea behind this mapping is to allow model-controlled vehicles to react to human-controlled vehicles, even if they are not traversing the road network as expected. To achieve the desired behavior some cases have to be considered, the cases described below can be seen in Figure3.2 :

Case 1: Vehicle driving on the nearest road. In this case the speed and acceleration vector of the mapped vehicle should be equal to the speed and acceleration

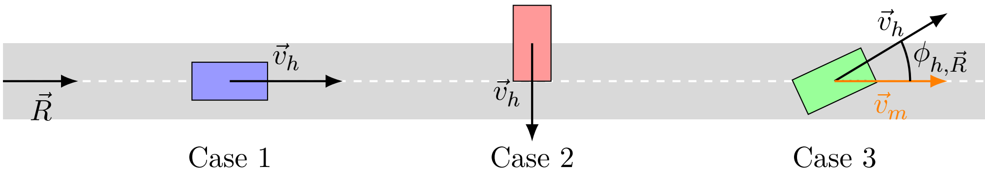

Figure 3.2: Cases of vehicles traveling on road network

of the human-controlled vehicle.

$$\vec{v_h} \times \vec{R} = 0 \Rightarrow v_m = |\vec{v_h}|$$
$$\vec{a_h} \times \vec{R} = 0 \Rightarrow a_m = |\vec{a_h}|$$

Case 2: Vehicle crossing the nearest road. In the case of a vehicle crossing the nearest road, the speed of the mapped vehicle should be zero. This is used to block the road for upcoming vehicles when one such human-controlled crosses the road.

$$\vec{v_h} \cdot \vec{R} = 0 \Rightarrow v_m = 0$$
$$\vec{a_h} \cdot \vec{R} = 0 \Rightarrow a_m = 0$$

Case 3: Vehicle moving on the road on an angle. Like in the 2 cases before the speed and acceleration of the mapped vehicle should correspond to the component of the speed and acceleration vector of the human-controlled vehicle (see Figure3.2) In this case the angle between the road and the vehicles speed and acceleration vectors has to be calculated.

$$\cos(\phi_{h,\vec{R}}) = \frac{\vec{v_h} \cdot \vec{R}}{|\vec{v_h}||\vec{R}|}$$

To combine all the cases above the speed and acceleration are projected onto $\vec{R}$:

$$v_m = |\vec{v_h}| * \cos(\phi_{h,\vec{R}}) = \frac{\vec{v_h} \cdot \vec{R}}{|\vec{R}|}$$
$$a_m = |\vec{a_h}| * \cos(\phi_{h,\vec{R}}) = \frac{\vec{a_h} \cdot \vec{R}}{|\vec{R}|}$$

3.5 Road network export and sync

Chapter 4

Results

Chapter 5

Future work

References

Literature

[1] Christian Backfrieder, Christoph F Mecklenbräuker, and Gerald Ostermayer. “TraffSim–A Traffic Simulator for Investigating Benefits Ensuing from Intelligent Traffic Management”. In: *2013 European Modelling Symposium*. IEEE. 2013, pp. 451–456 (cit. on pp. 3, 7, 8, 10).

[2] Christian Backfrieder, Gerald Ostermayer, and Christoph F Mecklenbräuker. “TraffSim-A traffic simulator for investigations of congestion minimization through dynamic vehicle rerouting”. *ResearchGate* 15.4 (2014), pp. 38–47 (cit. on pp. 3, 7, 8, 10).

[3] JG Bender and RE Fenton. “On the flow capacity of automated highways”. *Transportation Science* 4.1 (1970), pp. 52–63 (cit. on p. 6).

[4] Mark Brackstone and Mike McDonald. “Car-following: a historical review”. *Transportation Research Part F: Traffic Psychology and Behaviour* 2.4 (1999), pp. 181–196. DOI: https://doi.org/10.1016/S1369-8478(00)00005-X (cit. on p. 6).

[5] Alexey Dosovitskiy et al. “CARLA: An Open Urban Driving Simulator”. In: *Proceedings of the 1st Annual Conference on Robot Learning*. Ed. by Sergey Levine, Vincent Vanhoucke, and Ken Goldberg. Vol. 78. Proceedings of Machine Learning Research. PMLR, Nov. 2017, pp. 1–16. URL: https://proceedings.mlr.press/v78/dosovitskiy17a.html (cit. on pp. 1, 3, 10).

[6] Peter G Gipps. “A behavioural car-following model for computer simulation”. *Transportation research part B: methodological* 15.2 (1981), pp. 105–111 (cit. on p. 6).

[7] Arne Kesting, Martin Treiber, and Dirk Helbing. “General lane-changing model MOBIL for car-following models”. *Transportation Research Record* 1999.1 (2007), pp. 86–94 (cit. on p. 8).

[8] Gentry Lee. “A generalization of linear car-following theory”. *Operations research* 14.4 (1966), pp. 595–606 (cit. on p. 6).

[9] Manuel Lindorfer, Christoph F. Mecklenbräuker, and Gerald Ostermayer. “Modeling the Imperfect Driver: Incorporating Human Factors in a Microscopic Traffic Model”. *IEEE Transactions on Intelligent Transportation Systems* 19.9 (2018), pp. 2856–2870. DOI: 10.1109/TITS.2017.2765694 (cit. on pp. 2, 3).

[10] Gordon Frank Newell. "Nonlinear effects in the dynamics of car following". *Operations research* 9.2 (1961), pp. 209–229 (cit. on p. 6).

[11] Louis A Pipes. "An operational analysis of traffic dynamics". *Journal of applied physics* 24.3 (1953), pp. 274–281 (cit. on p. 5).

[12] Vincenzo Punzo and Biagio Ciuffo. "Integration of driving and traffic simulation: Issues and first solutions". *IEEE transactions on intelligent transportation systems* 12.2 (2010), pp. 354–363 (cit. on p. 2).

[13] Andreas Riegler, Andreas Riener, and Gerald Ostermayer. "From Me to We: Combining Driving Simulation and Traffic Simulation for Holistic Usability and Safety Research". In: *Adjunct Proceedings of the 15th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. 2023, pp. 294–296 (cit. on p. 1).

[14] Philip A Seddon. "A program for simulating the dispersion of platoons of road traffic". *Simulation* 18.3 (1972), pp. 81–90 (cit. on p. 6).

[15] Thomas Nguen That and Jordi Casas. "An integrated framework combining a traffic simulator and a driving simulator". *Procedia - Social and Behavioral Sciences* 20 (2011), pp. 648–655. DOI: https://doi.org/10.1016/j.sbspro.2011.08.072 (cit. on p. 1).

[16] Tomer Toledo et al. "Microscopic traffic simulation: Models and application". In: *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges*. Springer, 2005, pp. 99–130 (cit. on p. 1).

[17] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. "Congested traffic states in empirical observations and microscopic simulations". *Phys. Rev. E* 62 (2 Aug. 2000), pp. 1805–1824. DOI: 10.1103/PhysRevE.62.1805 (cit. on p. 7).

[18] Martin Treiber and Ame Kesting. "An Open-Source Microscopic Traffic Simulator". *IEEE Intelligent Transportation Systems Magazine* 2.3 (2010), pp. 6–13. DOI: 10.1109/MITS.2010.939208 (cit. on p. 8).

[19] Martin Treiber and Arne Kesting. *Traffic flow dynamics*. Vol. 227. Springer, 2013, p. 228 (cit. on pp. 1, 5–8).

[20] Martin Treiber, Arne Kesting, and Dirk Helbing. "Delays, inaccuracies and anticipation in microscopic traffic models". *Physica A: Statistical Mechanics and its Applications* 360.1 (2006), pp. 71–88 (cit. on p. 2).

[21] Xingwei Wang, Hong Zhao, and Jiakeng Zhu. "GRPC: A communication cooperation mechanism in distributed systems". *ACM SIGOPS Operating Systems Review* 27.3 (1993), pp. 75–86 (cit. on pp. 10, 11).

[22] Rachael A. Wynne, Vanessa Beanland, and Paul M. Salmon. "Systematic review of driving simulator validation studies". *Safety Science* 117 (2019), pp. 138–151. DOI: https://doi.org/10.1016/j.ssci.2019.04.004 (cit. on p. 1).

Online sources

[23] Gabriel Plassat. Feb. 2022. URL: wiki.lafabriquedesmobilites.fr (cit. on p. 8).

Check Final Print Size

— Check final print size! —

width = 100mm
height = 50mm

— Remove this page after printing! —