DOCTORAL THESIS

Advanced Autonomous Vehicle's Functions for Safety Improvements in Urban Mobility Context

Ehsan Malayjerdi

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2022

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
46/2022

Advanced Autonomous Vehicle’s Functions for Safety Improvements in Urban Mobility Context

EHSAN MALAYJERDI

TALTECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Mechanical and Industrial Engineering
This dissertation was accepted for the defence of the degree 16/07/2022

Supervisor: Professor Raivo Sell,
Department of Mechanical and Industrial Engineering,
Tallinn University of Technology
Tallinn, Estonia

Opponents: Professor Kari Tammi,
Department of Mechanical Engineering
Aalto University
Espoo, Finland

Professor M. Ilhan Akbas,
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Florida, USA

Defence of the thesis: 29 August 2022, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Ehsan Malayjerdi ______________________
signature

ISSN 2585-6898 (publication)
ISBN 978-9949-83-881-3 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9949-83-882-0 (PDF)
Printed by Koopia Niini & Rauam

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
46/2022

Täiustatud autonoomsete sõidukite funktsioonid ohutuse parandamiseks linnaliikluse kontekstis

EHSAN MALAYJERDI

KIRJASTUS

Contents

List of Publications

The present PhD thesis is based on the following publications that are referred to in the text in Roman numbers.

I Raivo Sell, Eero Väljaots, Tengiz Pataraia, and Ehsan Malayjerdi. Modular smart control system architecture for the mobile robot platform. *Proceedings of the Estonian Academy of Sciences*, 68(4):395–400, 2019

II Ruxin Wang, Raivo Sell, Anton Rassolkin, Tauno Otto, and Ehsan Malayjerdi. Intelligent functions development on autonomous electric vehicle platform. *Journal of Machine Engineering*, 20, 2020

III Mohsen Malayjerdi, Bariş Cem Baykara, Raivo Sell, and Ehsan Malayjerdi. Autonomous vehicle safety evaluation through a high-fidelity simulation approach. *Proceedings of the Estonian Academy of Sciences*, 70(4):413–421, 2021

IV Raivo Sell, Ehsan Malayjerdi, Mohsen Malayjerdi, and Baris Cem Baykara. Safety toolkit for automated vehicle shuttle-practical implementation of digital twin. In *2022 International Conference on Connected Vehicle and Expo (ICCVE)*:1–6. IEEE, 2022

V Ehsan Malayjerdi, Raivo Sell, Mohsen Malayjerdi, Andres Udal, and Mauro Bellone. Practical path planning techniques in overtaking for autonomous shuttles. *Journal of Field Robotics*, 39(4):410–425, 2022

VI Igor Astrov, Andres Udal, Heiko Pikner, and Ehsan Malayjerdi. A model-based LQR control of an obstacle avoidance maneuver of a self-driving car. In *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*:473–478. IEEE, 2022

Author's contributions to the publications

I The author designed and constructed a control system, implemented an experimental validation methodology for the system using an unmanned ground vehicle and TalTech iseAuto, and contributed to the analysis and preparation of the paper.

II The author proposed the HAVI methodology, developed the TalTech iseAuto software to integrate the light controller with the planning algorithm and implemented the experiments with the selected method, and helped prepare and finalize the publication.

III The author developed a methodology for experimentally validating simulations of overtaking manoeuvres on the TalTech iseAuto, developed a simulation setup for Simulator in the Loop, and helped prepare and finalize the manuscript.

IV The author validated an overtaking scenario in the digital twin and developed a 3D map and a high-definition map for simulations. The author took part in evaluating the results by using the TalTech iseAuto as well as writing the paper.

V The author proposed the initial idea and prepared and finalised the paper for publication after developing the methodology, conducting the experiments on the TalTech iseAuto, analysing the results, discussing them, and drawing conclusions.

VI The author proposed the initial idea, the initial parameters of the nonlinear model evaluated for the TalTech iseAuto, and contributed to preparing and finalising the paper for publication.

Abbreviations

AD	Autonomous Driving
ADAS	Advanced Driver Assistance System
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
APF	Artificial Potential Fields
AS	Autonomous System
AV	Autonomous Vehicle
BDD	Block Definition Diagram
CNN	Convolutional Neural Network
DARPA	Defense Advanced Research Projects Agency
DLC	Discretionary Lane Change
DT	Digital Twin
ECU	Electronic Control Unit
GA	Genetic Algorithm
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HARA	Hazard Analysis and Risk Assessment
HAVI	Human And Vehicle Interaction
IBD	Internal Block Diagrams
IMU	Inertial Measurement Unit
ITS	Intelligent transportation systems
LiDAR	Light Detection And Ranging
LQR	Linear Quadratic Regulator
MLC	Mandatory Lane Change
MPC	Model Predictive Control
NN	Neural Network
NPC	Non-Playable Characters
ODD	Operational Design Domain
PID	Proportional Integral Derivative
RCS	Real-time Control System
ROS	Robot Operating System
RQ	Research Question
RTK	Real Time Kinematic
RViz	Ros Visualization
SC	Smart City
SCMPC	Scenario based Model Predictive Control
SDG	Sustainable Development Goal
SiL	Simulation in Loop
SMC	Sliding Mode Control
SOTIF	Safety Of The Intended Functionality
TB	Test Bed
UAS	Unmanned Aircraft System
UGV	Unmanned Ground Vehicle
V2X	Vehicle to Everything
VR	Virtual Reality

1 Introduction

1.1 Background

The automobile has become a primary means of transportation over the past century. The automotive industry has grown exponentially because of its capacity to mass-produce safe, reliable, affordable vehicles [1]. However, for moving from one location to another, these vehicles require a skilled driver. In recent years, technological advances have transformed old-fashioned and mechanical means of transport into smart and information-rich vehicles. By integrating electronics and software into automobiles, advanced driver assistance systems (ADAS) have been introduced. This feature has already contributed to saving lives and preventing injuries, thus improving overall safety and convenience for the driver and passengers.

The integration of Artificial Intelligence into Autonomous Vehicles (AV) is one of the three revolutionary technologies. Travel and modes of transportation are slowly changing due to this revolution [2, 3]. Researchers and academics are investing a great deal of time and energy into Autonomous Driving (AD) technology - the highest level of smart vehicles. Currently, most research focuses on passenger cars and commercial vehicles. AVs must be safe and reliable to drive on public roads. AD requires a robust hardware and software architecture.

As a result of using AD and ADAS, fewer accidents are expected to occur, they will reduce driving stress and emissions, and people with disabilities will be able to use cars much more [4]. Traffic-related issues have been a significant concern in society for decades. Road accidents are still the most feared incidents for drivers and passengers. Around 1.2 million people die and up to 50 million are injured each year in automobile accidents [5]. Road accidents cause a lot of economic damage to individuals, families, and societies [6]. Globally, these accidents account for about 3% of the world's gross domestic product (GDP). Consequently, automotive systems tend to minimize the impact of human factors on transportation safety. Therefore, the goal is to eliminate human involvement by automating the entire driving process. An AV, also known as a self-driving vehicle, is an emerging technology that can sense its environment and drive itself without the need for a human driver. Machine-learning techniques can be used to process the data generated by various sensors and take the control of actions to complete the driving process autonomously with the latest advances in computational power. These techniques allow for improved comfort, efficiency, and safety.

1.1.1 Improving efficiency

There are many ways to increase the efficiency of AVs. (i) The AV can be programmed to improve energy consumption [7]. ii) The AV can follow optimal trajectory paths that are free from inefficient and unnecessary manoeuvres. (iii) When most vehicles are fully automated, traffic efficiency may improve significantly. AVs can communicate and coordinate with one another on a scale not understandable to humans [8]. AVs can accelerate the shift away from private ownership to ride/car-sharing. Reducing the number of personal vehicles on the roads by approximately one-third would enable meeting personal mobility requirements. As a result, Sustainable Development Goals (SDG) may be achieved with lower environmental impact [8].

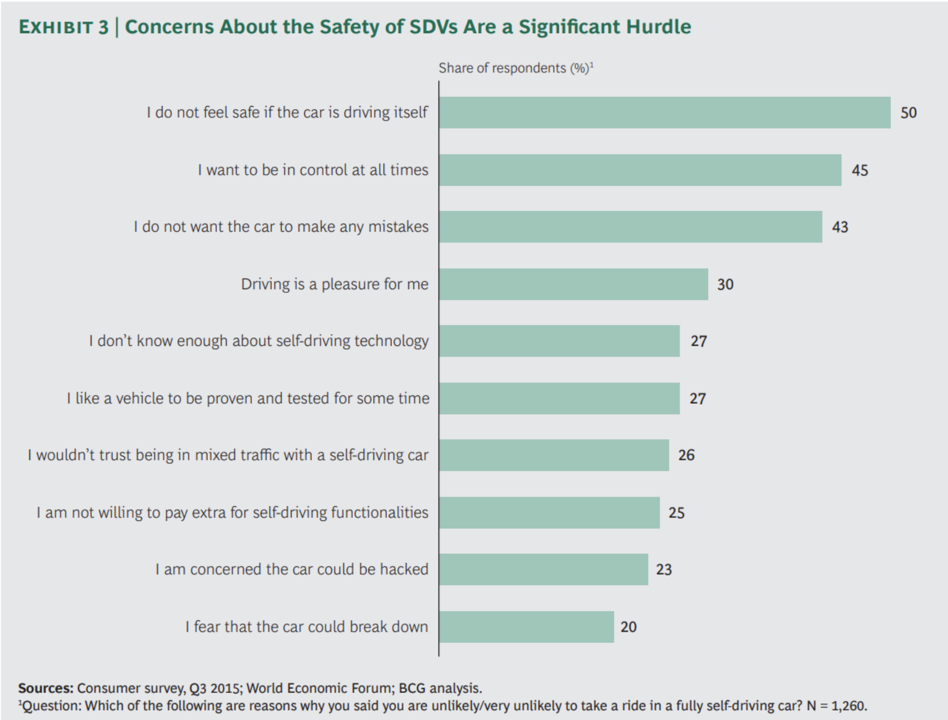

Figure 1: Consumers' main concerns about self-driving cars. Figure from [9]

1.1.2 Improving safety

Research and surveys show that safety in AVs is consumers' main concern [10, 11, 12](Fig. 1).

Intelligent Transportation Systems (ITS), which provide a variety of solutions for vehicles and traffic control, can reduce traffic congestion and accidents caused by human error while improving safety, efficiency, and passenger comfort [13, 14]. A variety of autonomous vehicle tasks can successfully be performed with limited human involvement using ITS, including lane-following, lane-changing, merging, platooning, and overtaking [15, 16]. Human drivers tend to make irrational decisions because of emotional reactions, which can result in road accidents. AVs can be programmed to perform driving tasks without endangering safety. These devices are equipped with sensors that have a better range and sensitivity. Unlike human drivers, who can only see in the visible spectrum, they can collect a lot more information about the surrounding environment.

Advancements in automated vehicle technologies have the potential to completely change the landscape for commuters, commercial fleets, and shared mobility providers while dramatically reducing fatalities and injuries from road traffic crashes. Although autonomous vehicles promise significant benefits and their complexities and wide range of algorithmic solutions introduce significant safety challenges, particularly for those with limited driving experience, many people still hesitate to trust technology.

With disruptive technologies that depend on the functionality of software and systems, safety becomes even more critical. To build confidence in autonomous technologies, innovators will need to understand and align with industry safety standards.

To ensure safety at all levels of autonomous vehicles, manufacturers must follow a multitude of standards and industry best practices. Here is a list of relevant standards:

ANSI/UL 4600 [17] A Standard for Safety for the Evaluation of Autonomous Products addresses fully autonomous products that do not require human intervention. It is used in autonomous vehicles, mining, agriculture, and Unmanned Aircraft Systems (UASs). In addition to that, it could be adapted for other uses [18].

ISO 26262 [19] The standard that regulates functional safety of road vehicles. To identify risky activities in a system and to specify safety goals that reduce the risks, the Hazard Analysis and Risk Assessment (HARA) method is recommended [20].

ISO 21488 [21] The standard that addresses a system's unintended behaviour on the safety of the intended functionality (SOTIF) when there are no ISO 26262 failures. A driver assistance system and an emergency intervention system are included in this standard. For higher levels of automation, additional measures may be required to achieve SOTIF, although ISO 21488 is complementary to ISO 26262, as defined by SAE International Standard, J3016 [22].

ISO/SAE 21434 [23] The standard covers road vehicles and their components and defines the requirements for cyber risk management. To reduce vulnerability to cyber-attacks, ISO 21434 encompasses the entire product lifecycle from concept to manufacturing, operation, and disassembly service [24].

1.2 Motivation and research gaps

There are different levels of autonomy available in AV technologies, from minimal autonomy (which most vehicles are today) to recent offerings from Tesla, Inc. and Waymo LLC, which allow AD in certain structured road conditions, but still require partial manual supervision [25].

A truly driverless system should be able to handle more complex situations than these scenarios. To determine the level of automation for a vehicle, the SAE Levels of Driving Automation can be used, ranging from "No Driving Automation (level 0)" to "Full Driving Automation (level 5)" [25, 26]. A detailed illustration of this can be found in Fig. 2. When autonomous vehicles are capable of handling all driving situations without human intervention, passengers' safety is ensured. Based on this information and literature review in section2, some of these issues are addressed in this dissertation and the following research gaps are identified:

Research gap 1 To achieve autonomous navigation and safe driving, more intelligent software and hardware is needed, including high-performance sensors like light detection and ranging (LiDAR), as well as cameras. Even though most ADs similarly perform different sub-tasks such as localization, global planning, and obstacle avoidance, they are still highly dependent on sensors and AV hardware. A major question is how to develop an integrated architecture for AVs for better performance and how to make this architecture sufficiently modular to be used on any kind of autonomous system (AV, UGV, mobile robots).

Research gap 2 To evaluate the reliability of AVs and place them on roads, they must be driven billions of kilometres. Considering the real-world crashes for AVs in the past, what could be the best way to test and validate the functionality of AVs before real-world testing? It is expensive to test the behaviour of an AV. Some companies test AVs on large testbeds, but in some research environments, it is difficult to find a

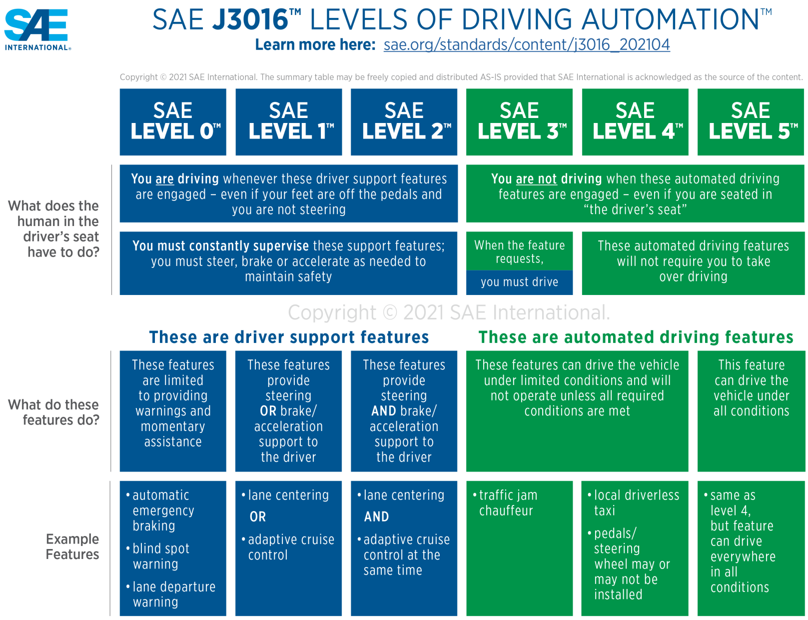

Figure 2: Five levels of AD by SAE. Figure from [26].

place for evaluating them. There is a lot of simulation software such as AutonoViSim [27], Roadview [28], LGSVL simulator [29], and Carla simulator [30]. It is still a great challenge to find a high-fidelity simulator with a 3D model of the AV that would be similar to the real AV, and a 3D environment similar to the testbed where AVs can be tested and validated.

Research gap 3 In a typical driving situation, a Fully Automated Vehicle (Level 5) should be capable of performing a wide range of complex driving manoeuvres, such as keeping lanes, switching lanes, overtaking, etc. Among these manoeuvres, overtaking is the most challenging one since being one of the riskiest for both human drivers and autonomous driver agents. These types of operations pose many challenges and issues for AVs. The technology is not yet mature enough to perform these manoeuvres safely. More experiments and improvements are required to improve reliability and safety.

1.3 Objectives and research questions

The main objective of this dissertation is to develop advanced AVs functions for safety improvements in the urban mobility context. These functions include:

- Introducing a modular control system for AVs and robots.
- Proposing an integrated Vehicle to Everything (V2X) communication module for the AV.
- Developing a high-fidelity simulation for safety evaluation.

- Developing digital twin for safety evaluations and testing the new overtaking algorithm on the simulation environment.
- Implementing a new overtaking method on the AV.

This publications-based dissertation concentrates on the summary of the following research questions (RQ):

RQ1 What is the optimal design of an autonomous vehicle control system?

RQ2 How to ensure proper communication between driverless cars and humans to ensure the pedestrians' safety?

RQ3 How to validate the safety of AV algorithms in particular traffic environments?

RQ4 How to improve the complex manoeuvres like overtaking to ensure higher safety, smoothness and reliability of the planning algorithm?

According to the six publications that have contributed to the current PhD thesis, the research questions have been addressed:

- Research paper I is a study focusing on the **RQ1**.
- Research paper II is a study focusing on the **RQ2**.
- Research papers III and IV focus on the **RQ3**.
- Research papers V and VI focus on the **RQ4**.

1.4 Research process and framework of the dissertation

During the related study, several methodologies were used. Ensuring the safety of operating AVs is the key to promoting AVs and is a prerequisite to achieving or even discussing their safety, mobility, and sustainability benefits. The main techniques used in this experiment-based study are literature reviews, the formulation of case study concepts, and the execution of the studies themselves. A concept of software and hardware architecture for AVs has been introduced including research related to AVs. A HAVI framework has been proposed for AVs and pedestrian communication supplemented with an experiment on the TalTech campus. Simulation environments with high levels of accuracy have been developed and a new AV overtaking method has been developed and experiments conducted to verify theoretical findings. Fig. 3 illustrates how the relevant dissertation is structured.

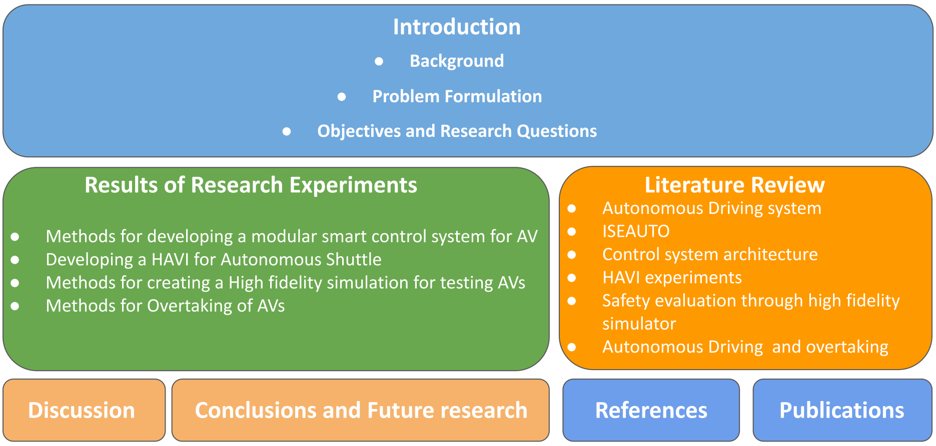

Figure 3: Providing an overview of the dissertation's framework

2 Literature Review

2.1 Autonomous Driving Systems

To create an AD agent, three main tasks must be completed: perception, planning, and control Fig. 4. Identifying all components of the surrounding environment is a key aspect: it may include traffic signs, obstacles, or other vehicles. Several models are introduced in the introduction chapter to transform the recognised components into predicted future states. Implementing and integrating past information is essential for predicting the future. The actual planning of future actions is one of the most challenging tasks. When planning future actions, it is necessary to be able to incorporate a model that can predict the environment and its dynamics. Hence, it is important to avoid unwanted situations and reach the destination safely [31].

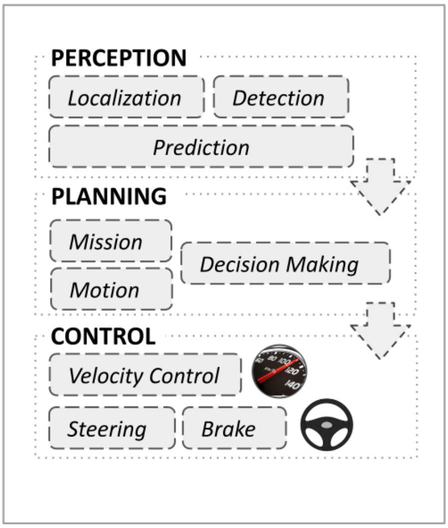

Figure 4: AD software architecture

2.1.1 Perception

To create a coherent, consistent and reliable representation of the environment, the information collected by the sensors needs to be extracted and further processed. Thus, the perception module is fundamental to any AD system. It provides information about obstacles, road layouts, traffic signs, and other features in the environment. Classical methods are not suitable for this understanding therefore, deep learning methods, particularly Convolutional Neural Networks, are practically standard for detection and recognition tasks. Additionally, the module provides localization of the vehicle based on data from IMUs and GPS, as well as visual odometry.

Localization There are two separate tasks involved in AD. Firstly, to ensure that the vehicle keeps following the correct path. The accurate location of the vehicle on the road is required to accomplish this task. It is still not possible to completely meet these constraints in dense urban environments, even using the most precise positioning systems based on the Global Navigation Satellite System (GNSS), GNSS with RTK correction, or GNSS and IMU fusion. Secondly, the ability to recognize and react to dynamic obstacles like cars, pedestrians, and traffic signs. While most localization algorithms use Kalman filters and similar techniques [25, 32], deep learning-based algorithms are currently being investigated that can combine localization with the estimation of and prediction of motions of the surroundings at the same time [25].

Detection Detection systems usually utilize various complementary sensors, among which LiDAR and cameras are two of the most common types. Cameras capture data in the form of a well-structured, but limited, 2D image with rich context information [33, 34]. The LiDAR sensor, on the other hand, provides low-context data in the form of a 3D point cloud with a high level of resolution and precision. Since LiDAR sensors are not limited by lighting conditions and can collect data in a more private manner than cameras, they are ideal both for technical and ethical considerations in a complex outdoor environment. In recent years, deep learning techniques have shown remarkable success in the real-time application of object detection in 2D image planes. As well as detecting objects in a 2D plane, certain classes of objects for AD applications, including cars, pedestrians, and cyclists, must also be detected in 3D space. An object detection system based on LiDAR 3D point clouds solely relies on the sensor data from this sensor. While LiDAR-based systems for AD applications are becoming more widely used, detecting 3D objects remains a challenge, as it requires both high detection performance and fast estimation times. LiDAR sensor data quality directly impacts effectiveness. Although LiDAR works without visible light, which enhances the environment perception at night and under low light conditions, extra noise introduced by adverse weather conditions affects the quality of the data, which impacts the ability to detect objects. Further, partial occlusions or self-occlusions can result in incomplete data, which when combined will reduce the quality of the data.

Prediction Increasing road safety and reducing road accidents are likely to be benefits of AVs [35]. AVs must, however, be able to improve their performance on roads not only by understanding the current state of local road users but also by predicting their future actions. Computer vision literature has discussed the subject of predicting pedestrian behaviour, which has been extensively studied [36, 37]. Several review articles have been written on pedestrian behaviour prediction including [38, 39, 40, 41]. Predicting the intended behaviour of other cars on the road is a problem as well. Vehicular inertia, driving rules, and road geometry are more constraining on vehicles than on pedestrians, which may help reduce the complexity of the problem. However, the interdependence of vehicle behaviour, the influence of traffic rules and driving environment, and the multi modality of vehicle behaviour present new challenges. It is also challenging to observe the surrounding environment because of practical limitations, in addition to the need for computational resources to execute the prediction algorithms.

2.1.2 Planning

In this module, the information derived from the perception layer, and the current state of the AV are used to generate a safe and optimal trajectory, which is based upon any dynamic constraints that assure the passengers' safety and comfort. Planning can be divided into the modules described below.

Mission Planning Particularly, path planning dictates the behaviour of an autonomous vehicle. There are global and local stages in autonomous vehicle planning algorithms. A digital map and a localization system are used to determine the global path and vehicle states during the global planning stage. In the local planning stage, the vehicle's local path can be generated based on the information gathered from sensors such as cameras and LiDARs [42, 43, 44]. There are several types of real-time path planning algorithms for AVs, including a sample-based method, graph-search method [45, 46], and geometry-based method [47, 48]. To navigate through the path plan, the AV must be able to interact with other vehicles and follow traffic rules and road conventions. The AV can do this by choosing from multiple driving modes. For instance, if another car is driving slowly in front of the AV, it can choose to overtake when there are no cars in the opposite lanes. Keeping a safe distance from the vehicle is crucial if it chooses the following mode. Information from the perception module is used to select the mode. It is more complex to make fast decisions in dynamic conditions. AV mission planning methods need to be fast, reliable, flexible, and accurate.

Motion Planning A trajectory (driving manoeuvre) chosen by the mission planning module must be trackable by the vehicle. Furthermore, it must keep a safe distance from obstacles while providing comfort for passengers and observing road geometry and rules. This can be achieved through the trajectory planner module, which moves the AV from the start point to the goal point. The vehicle should be able to track this trajectory without requiring user commands. Due to computation requirements, it is impossible to find exact solutions and, therefore, approximate numerical methods are used.

Most of these techniques are derived from mobile robotics and focus on local motion planning. Several approaches to local motion planning have been proposed in the literature. Specifically, local path planning, and velocity planning [49], are the two methods for motion planning. In comparison to traditional path planning methods, Dijkstra's algorithm generally does not use the velocity information in a static environment, making it more likely to search for a collision-free path between objects from the vehicle location to a temporary terminal location. Generally, numerous studies search for path planning or motion planning methods that are based on geometry, grids, sampling, meta-heuristics, artificial potential fields (APFs), and AI [50].

The geometry-based method presented by [51], uses the Voronoi cell algorithm to generate paths. Due to Voronoi edges being discontinuous, this algorithm performs well only in static environments rather than dynamic ones. Therefore, this algorithm would not be appropriate for nonholonomic vehicles. An algorithm based on Anytime Dynamic A* is presented in [52] to generate manoeuvres for high-speed AVs over large distances. This method is fast, but it does not yield a continuous path and it is not easy to find the heuristic rule. A sampling-based algorithm is widely used for motion planning. The state-space sampling method is used by [49] to develop a trajectory planning capability. In general, the paths connecting the initial state with the sampling terminal states are kinematically feasible and smooth. In a dense transportation environment, this method may fail to produce a path because the sample space for state sampling is so small. Using the rapidly-exploring random tree algorithm [53] propose a method for designing motion for AVs. This algorithm is can explore a space very quickly, but the resulting trajectory is non optimal, jerky, and does not follow the curvature of the space, so further smoothing is required. Using deterministic discretization of the state space, the state lattice [54, 55, 56] is another sampling-based method. To deal with static obstacles, the original version is non temporal, and the spatiotemporal state lattices are used to plan for moving obstacles; however, its accuracy and real-time performance are affected by the sampling density. Creating a lot of trajectory candidates and evaluating them all individually is time-consuming. The metaheuristic method can also be effective. A motion-planning method based on genetic algorithms is presented in [57]. The method shows high accuracy and the robot responds rapidly to changes in the environment. However, in terms of real-time applications, Genetic Algorithm's (GA) computation complexity can be a problem when used in AVs. There is also the classic APF which is widely used. The advantages of this approach are that obstacles can be avoided in real-time; however, the solutions may easily be stuck in local minima [58]. Applied AI methods have been developed for motion planning recently. There is also an end-to-end system presented in [59] that converts raw camera pixels into vehicle steering commands using convolutional neural networks (CNN). Motion planning using this technique is much more efficient and effective in certain situations than conventional methods. While the data collection and training process is hard, the robustness of the system can be improved. Using temporal optimization, a speed profile is calculated based on the timestamps for all waypoints along the path, and the path is separately planned [60]. Using both the spatial and temporal space, the path planner [61] identifies the best trajectory according to a set of cost functions. The authors of [62] combine the APF and model predictive control (MPC) to determine the trajectory automatically and predictably.

Decision Making Decision layers are typically categorized as path planning, behaviour planning, and motion planning. An autonomous vehicle should be able to decide on a path based on the passengers' requirements (e.g., shortest path), traffic conditions, and the road network from the current position to the destination. According to the current traffic conditions, such as traffic flow and signals, a behaviour planner can be used to determine a sequence of driving behaviours (e.g., turning, stopping, or driving straight) along the planned path. Vehicles can follow a safe (i.e., collision-free and stable) economic, human-like trajectory based on a predefined path and predetermined behaviour with motion planning. Based on the missions, the onboard sensors, and the online sensor information, an accurate plan of a global route and behaviour can be determined [63]. There must be both continuous and discrete uncertainty incorporated into the decision-making process, such as the actual position of the vehicles, which may change depending on whether someone is turning or trying to overtake another vehicle. During the Defense Advanced Research Projects Agency (DARPA) Urban Challenge project, traditional decision-making methods used predefined rules and handmade state machines [64, 65, 44]. Traditional motion planning methods are used to solve the decision-making task [66, 67, 68]. It is difficult to scale these methods to the complexity of real-world driving, even though they are successful in many situations because they are designed for the specific driving situations.

2.1.3 Control

By executing effective actuation commands, the control module should be able to track the trajectory generated by the previous module. [62] fuses the decision and control algorithms, which improves formulation but increases the computation time. Because of this limitation, most of the research in this field deals with the planning and tracking problems separately and in hierarchical manner [49]. A closed-loop control system controls steering, throttle, and brake commands by using a model of the vehicle. The controller must be robust and stable. Proportional integral derivative (PID) is a classic method used in AV. To achieve good tracking performance, these methods require expert tuning and result in more complex formulations, such as Sliding Mode Control and Adaptive PID [69]. As a result of recent advances in vehicle modelling and onboard computing capability, optimal control methods have been developed, such as MPC and Linear Quadratic Regulator (LQR) [69]. Proposed method in [70] uses an MPC with input and state constraints, recommending the yaw rate and sideslip angle be included in the MPC formulation.

2.2 TalTech iseAuto

TalTech iseAuto (Fig. 5) is the first autonomous vehicle operating in Estonia implemented at Tallinn University of Technology (TalTech) in collaboration with two companies in Estonia, AuVeTech and ABB. The TalTech iseAuto project aims to design and develop a self-driving vehicle. As a product of the cooperation and joint development, the pre-commercial AV shuttle can be used in university research and educational activities as well as a commercial vehicle in urban mobility pilot projects. Open-source software and a modular design enable the vehicle to be manufactured at a lower price. The vehicle was demonstrated in September 2018, and it has been a huge success since then.

Figure 5: TalTech iseAuto, the first self-driving shuttle in Estonia

2.3 Control system architecture in AVs

Automated vehicle architecture is like the architecture of a real-time, intelligent control system. Reference architectures of such systems have heavily been discussed in the literature of robotics and AI fields [71, 72, 73, 74]. These solutions are based on either behaviour-based systems or knowledge-based systems, where behaviour-based systems do not maintain an internal state of the environment [75]. ISO26262 standard [71] has played an important role in the development of AD systems by addressing such important aspects as system and component development, hardware and software testing, and safety. Furthermore, the issues regarding the safety in AD are an entire research field, including the aspects of the development of AVs as architecture, implementation methods, and the impact of reconfigurable and flexible hardware [76].

2.4 Communication between autonomous vehicles and pedestrians

Users must be able to trust automated systems to accept them, and even rely on them. According to [76, 77], increasing trust is influenced by three constructs: ability, benevolence, and integrity. Therefore, trust can be achieved through investigating and understanding the behaviour of the system and its underlying processes [78]. HAVI is a crucial component of AD systems because it provides information regarding the performance of the vehicle. Different interface prototypes have been created by researchers [79, 80] and designers [81] to demonstrate how pedestrians and automated vehicles might communicate in the future. According to Malmsten Lundgren et al. [82], even if eye contact is lost due to vehicle automation, pedestrians' perception of safety could be maintained if they are given the appropriate information as an external interface to the vehicle. As Keferböck and Riener [83] showed, pedestrians have different levels of trust and confidence based on automation and AVs must communicate actively with pedestrians. The use of VR environments in traffic studies and the assessments of crossing behaviour is a growing trend today. As well, VR provides a controlled test environment that can be easily repeated and can be used to monitor people's motion, eyesight, and behaviour [84]. Numerous studies demonstrate the benefits of interacting with Virtual Reality (VR). Boeckle [80] explains that it has a positive impact on pedestrians. According to Lundgren et al. [82], their system can sufficiently replace human-human interaction. According to Chang et al. [85], pedestrians made faster decisions. The participants also reported feeling safer as they crossed the street. According to De Clercq et al. [86], pedestrians prefer displays with text. The interaction display created by Matthews et al. reduced deadlock situations by 38 % [79]. According to Mahadevan et al. [87], when interacting with a car, the main signal is speed, but displaying information to the VR is still helpful. It is not enough to inform the VR that they are detected. Pedestrians want to know more about AVs intentions. Therefore, they recommend a variety of easily interpreted visual signals. Clamann and colleagues [88] found that only 12% of their participants said that the display affected their decision. The distance and speed were the major factors. About half of the participants, however, believed that the display helped them in their decision. This is like what Li et al. [89] found. Other communication methods are recommended at night. The study found that position, speed, and the environment all play a role in a pedestrian's decision to cross the road [90, 91, 92]. Researchers have not yet reached a consensus on whether AVs should be equipped with displays to facilitate pedestrian interaction. Nonetheless, they are helpful for pedestrians.

2.5 Safety evaluation through a high-fidelity simulation

As part of the development process, safety is one of the most critical elements that cannot be overlooked. Although safety is the most important concern in AV deployment, ordinary users and most industry stakeholders think that safety is one of the most critical importance. Several surveys conducted in recent years reveal that people are concerned about safety and security around the world [93, 94]. Over 70% of respondents considered safety issues important. Therefore, AV technologies will become popular once they become safe. In recent years, industry and academia have been concentrating more on safety aspects resulting in developing standards and procedures for evaluating, validating, and verifying safety. Along with passengers and drivers, it also includes pedestrians, road traffic, micromobility traffic, etc. Therefore, all safety concerns must be addressed satisfactorily for the public. In addition, many topics regarding automated driving, the aspects of verification and validation should be addressed. Some of these topics are covered in detail in SAE Edge research reports [95, 96, 97]. This dissertation describes the safety evaluation toolkit for evaluating use cases, and the first step of the process involved creating a digital twin of the real use case environment. The digital twin is a model of a real environment that can be used to simulate or experiment in various ways. The following study is the advancement of simulation case studies [98], and digital twin developments [99] in TalTech AVs research group and cooperation with Florida Polytechnic University, Advanced Mobility Institute.

2.6 Autonomous Driving and overtaking

In modern cars, a variety of sensors and electronic systems provide emergency assistance (e.g., ABS, traction control, stability control), ADAS (e.g., cruise control, lane-keeping, blind spot detection, etc.), and navigation assistance (e.g., route planning, regular traffic updates, etc.) [100]. However, future intelligent vehicles will be able to drive independently in a variety of driving scenarios because of the increased capabilities [101, 102]. One of the most dangerous driving tasks is overtaking, so an autonomous vehicle must be capable of determining how, when, and whether it should perform this manoeuvre. While overtaking, AVs need information from other vehicles, such as their position, speed, interdistance, or angle. AVs rely on their state as well as the information they gather from their environment to determine their trajectories or future state. Because overtaking manoeuvres are not standardized and categorizable, they are difficult to categorize. Thus, policymakers and automotive manufacturers are exploring sustainable car tech to increase safety and efficiency. A successful overtaking manoeuvre involves proper completion of three sub-manoeuvres, namely: **(1)** lane change from the original lane to the overtaking lane, **(2)** cruising in the opposite lane to pass a faster-moving (or stopped) vehicle travelling in the same direction (lane-keeping), and **(3)** returning to the original lane [103]. The lane changing sub-manoeuvre in the first and last step is categorized into two: 1) Discretionary Lane Change (DLC) and 2) Mandatory Lane Change (MLC) [69]. In the DLC, changing lanes is executed when the traffic situation in the overtaking lane is better than the current lane, i.e., the lane change is performed expecting an improvement in the driving conditions. However, as opposed to DLC, the MLC sub-manoeuvre is enforced by traffic regulations (e.g., blocking the road by a halted vehicle for road constructions). Correspondingly, due to the driving environments such as traffic conditions, traffic regulations, and road conditions, the overtaking manoeuvre is not a standard method, and overtaking is unique in real scenarios regarding the number of vehicles, duration, relative velocities,

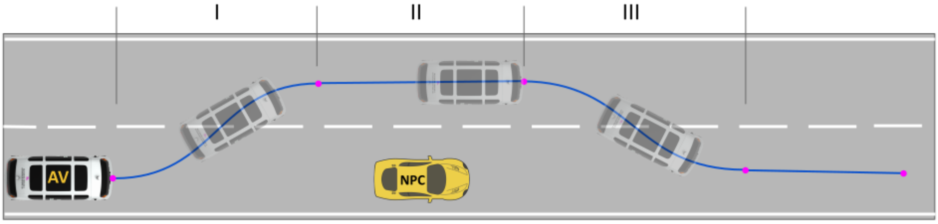

Figure 6: Overtaking sub-manoeuvrers. Figure from article V.

and the distance between vehicles [104]. In AV, an overtaking manoeuvre is evaluated based on the state of the vehicle and the surrounding environment to enable a discrete outcome to facilitate tactical decisions that are a part of decision-making and planning procedures. Fig. 6 shows a schematic representation of an overtaking manoeuvre. Each sub-manoeuvre is indicated by a Roman numeral.

Autonomous overtaking manoeuvres are critically dependent on the planning module. The current section explores control architectures and overtaking manoeuvres in related works.

2.6.1 Trajectory planning for autonomous overtaking

There are four well-known techniques for trajectory planning for overtaking applications, namely: potential fields, cell decomposition, interdisciplinary methods and optimal control [105]. Several topics are covered in this section, such as computational performance, safety, feasibility in overtaking at high speeds, as well as real-world applications.

Potential field algorithm defines an obstacle as a repulsive field and a safe zone as an attractive field. Then an algorithm is applied to calculate the trajectory along the steepest potential gradient in the resulting field [106, 107]. There is a guarantee that a path generated by this algorithm will follow the lowest potential (i.e., avoid all collisions) in a particular space, but its accuracy and safety heavily rely on the accuracy of the generated potential field (i.e., knowing the position of stationary and moving obstacles). Despite this, the method has only been experimentally proven to work for low speed (i.e., urban) manoeuvres due to high computation costs and the requirement for very accurate environmental information [107]. As well, the algorithm is not capable of handling vehicle kinematic constraints, potentially resulting in safety issues when driving at high speeds [48].

Cell decomposition algorithm is used to plan a collision free path [108]. It is possible to modify the algorithms to meet vehicle constraints, but the modifications have computational and memory costs. Increasing traffic density and frequency of road curvatures increases the computational complexity of such algorithms, creating an onboard computation problem for AVs on busy roads [108]. In addition, the paths generated by RRTs are unstable, and tracking such a trajectory can be uncomfortable for the passengers [109].

Multidisciplinary approach has also been developed for trajectory planning [110, 111]. A novel approach proposed by [112] involved the use of motion primitives (combinations of steady-state equilibrium trajectory and pre-specified movement). The experimental results demonstrated that collision-free and feasible trajectories can be generated in real-time using this approach [112]. A method developed by Ghumman et al. based on the rendezvous guidance technique (passing vehicle aligns with shadow target in real-time during overtaking manoeuvre) derived from missile guidance systems [110, 111, 113] proposed an approach for overtaking manoeuvres based on the tracking of virtual reference points that are placed at known distances from the leader vehicles. These simulation results demonstrate that both approaches can generate feasible trajectory generation in real-time, but tracking performance was confirmed using computer simulations with low-order models. Thus, it is difficult to conclude the efficacy of such methods in the absence of experimental validation.

Optimal control method minimizes a performance metric (e.g., change in kinetic energy [114]; jerking ; and lateral acceleration [48, 115]) while meeting a set of constraints (e.g., vehicle longitudinal and lateral limits, environment constraints, and surrounding vehicles). The literature shows that the proposed method is successful at generating collision-free trajectories without requiring many computational resources [48].

Unfortunately, most of these techniques do not account for the non-linearity in the vehicle and tire dynamics, resulting in unfeasible vehicle trajectory paths at high speeds and low friction on the road, which presents a safety risk for AVs [116]. These trajectory planning methods have limited potential unless used in highly structured or controlled environments, and therefore open-loop single stage optimisation cannot consider uncertainties in dynamic environments.

Recent research has used MPC methodologies for local trajectory planning because they can handle nonlinearities and system constraints more effectively. As part of determining which sequence of inputs minimizes (cost function) a performance index (cost function), the receding horizon principle is applied to a constrained finite-time optimal control problem [117]. While navigating in a dynamic environment, the difficulty of solving the optimisation problem in real-time is compounded by (i) dynamics of the vehicles, and (ii) changing input and state conditions [117]. To reduce the computational complexity arising from nonlinear dynamics, researchers have employed the following methods: (i) point mass vehicle models [67, 118, 119]; (ii) a linear model of bicycles [120, 121]; and (iii) the linearisation of the nonlinear vehicle model in iterative steps, in the prediction model [117]. Collision avoidance constraints are not convex, which means they cannot be guaranteed to be feasible and unique. For a controller to be unique and reduce the computing and memory requirements, researchers have devised several methods [67]. In the experiments, these approaches were able to generate collision-free paths around static or moving obstacles (i.e., overtaking manoeuvrers); however, it should be noted that to compute these safe paths, high-performance computing platforms are required to maintain state information, obstacles, and to maintain trajectory information. A major advantage of MPC in trajectory planning is its ability to accommodate system dynamics and constraints while adhering to receding horizons which may allow the planning of feasible paths over a wider operating window.

All methods discussed above assume that the trajectory planning system has access to accurate information on the environment and the lead vehicle state when needed. Table. 1 summarizes the benefits and drawbacks of the various abovementioned trajectory planning methods.

However, the following situations may occur because of the limitations of sensors. First, the measurements of the lead vehicle state (e.g., position, velocity, speed, and heading) could be inaccurate, lacking information, or inaccurate in general, resulting in inaccuracies in the representation of the environment. Furthermore, variations in external

Table 1: Benefits and drawbacks of the various trajectory planning methods. Table from [105].

Trajectory planning algorithms	**Strength**	**Weakness**
Potential fields	• Path search is guaranteed to be optimal • collision-free	• Costly computation • System constraints cannot be handled • Uncertainty about the environment is not considered systematic
Cell decomposition	• Ensure collision-free trajectory	• Requirements for computing are affected by traffic density • Calculated paths are unstable • There is no systematic way to evaluate environmental uncertainties
Interdisciplinary techniques	• As trajectory planning is converted to a reference tracking problem, collision avoidance becomes simpler • Real-time	• Untested experimentally • Don't consider uncertainty in perception of the environment when generating reference points
Optimal control	• Trajectory generation without collisions • Support for kinematic constraints	• Tire slip angles are too large for high-speed driving maneuvers
Model Predictive Control (MPC)	• Consider vehicle dynamics • Constraints and uncertainties are handled in a systematic way • Environment-independent computational requirements	• Tire dynamics are not considered • High-order system models, non-linearity, and nonconvexity of constraints increase computation complexity

conditions (such as road legislation, road surface condition, road width, weather, etc.) may affect the dynamic limits of the AV (such as lateral acceleration, longitudinal speed, etc.). In case of environmental variations and sensor inaccuracies, planning methods are not robust, posing a risk to safe reference trajectories and posing a major safety problem, particularly during high-speed driving. Based on the methods discussed above, trajectory planning techniques can be applied to deal with uncertainty in current environment perception and limited ability to predict the future. Using potential fields and cell decomposition methods, additional buffer zones are assigned to obstacles which will result in a less constrained space for searching feasible trajectories [122]. Similarly based on [110, 111], target virtual points will be conservatively calculated based on the relative velocity of the subject and lead vehicle. A type of MPC control technique called Scenario-Based MPC (SCMPC) is also proposed in the literature to mitigate unpredictability arising from traffic interactions in a systematic manner [123, 124, 125]. Several studies have demonstrated the practicality of the SCMPC trajectory planning technique for the generation of safe lane change manoeuvres, as well as its real-time capability [121]. A large quantity of actual traffic data, however, must be obtained to make this method effective.

2.6.2 Trajectory tracking for autonomous overtaking

There have been several comparisons of tracking controllers for AVs [126, 127, 128]. The following examples of tracking controllers for autonomous overtaking are discussed in conjunction with some relevant observations from these comparisons.

Geometric controller Two popular geometric controllers are pure-pursuit and Stanley [127, 128]. In pure-pursuit, a vehicle is continuously following a virtual moving point in front of it, and the Stanley controller is based on a non-linear geometric controller that calculates steering angle corrections based on heading and lateral error [126]. Pure-pursuit controllers, while easy to implement, are suitable only for applications that do not require consideration of vehicle dynamics. This approach lacks systematic control parameter tuning, thus making it difficult to strike a balance between tracking performance and stability [127]. Pure-pursuit and Stanley controllers are both affected by over-turning during manoeuvres.

Kinematic controller This type of feedback controller considers the kinematics of the vehicle (yaw rate, longitudinal velocity, etc.). In some cases, kinematic controllers have improved tracking performance over geometric controllers, though these improvements are insufficient to warrant the additional effort involved in designing and tuning the controller [126]. Due to the lack of consideration for vehicle dynamics in these methods, they should not be used in critical driving situations (high-speed driving, extreme path curvature, etc.).

Classical controller Various classical control algorithms are also found in literature, such as PID and sliding mode. Despite good performance, tracking controllers using classical approaches (PID) faced major challenges due to vehicle and tire nonlinearities. It has been demonstrated that sliding mode control (SMC), which is a well-established classical nonlinear state-feedback controller, is a good choice for tracker design due to its non-linearity [129]. Despite its benefits, it has some shortcomings, including (i) sensitivity to the sampling rate of controllers, (ii) noise issues, (iii) only for sliding surfaces, and (iv) it requires prior uncertainty and disturbance information [129].

Dynamic state feedback In comparison to geometric and kinematic controllers, dynamic state feedback (linear and nonlinear) based methods exhibit superior performance because they consider the dynamics of the vehicle and the tires when computing the control law. It is easy to design control laws based on linear quadratic regulators, but achieving error-free tracking requires feedforward control when tracking trajectories with varying curvatures. Due to the addition of feedforward control, the tracking controller becomes sensitive to discontinuities in the reference trajectory, which requires additional tuning to attenuate [130]. However, optimal control methods can produce accurate trajectory tracking even at high speeds, but this is possible only if certain preconditions have been fulfilled (e.g., the vehicle velocity should stay the same throughout the optimization process). The inversion immersion (II) method has been applied to controllers for tracing the trajectory of vehicles in recent years. The results of the first studies show that this method provides robust closed-loop tracking performance, although the controller is sensitive to uncertainty in parameter values [129]. As part of the same study, a Proportional-Integral (PI) with a non-linear gains controller for trajectory tracking was also described. According to simulation results, the controller is comparable with SMC and II controllers in terms of tracking performance and insensitivity were parameter uncertainty. The controller gains can become excessively high at the expense of the actuators when operating in non-linear regions of vehicle dynamics or in the presence of large curvature variations. actuators.

MPC Several advanced model-based control techniques, such as MPC, have also been applied to vehicle trajectory tracking [67, 112, 116, 117, 118, 119]. In a study conducted by [131], nonlinear MPC was found to offer accurate tracking, but as a result, it suffers from high computational requirements. Researchers use linear vehicle models to reduce the computational cost, however, these controllers are only applicable to linear regions of vehicle and tire behaviour [121]. To extend the operating range of linear MPC controllers for trajectory tracking, an MPC framework based on iterative linearization of a non-linear model has been proposed and experimentally validated [117]. This approach enables meeting the demands of computational accuracy and modelling errors.

Neural network Fuzzy logic Research has been published to demonstrate the tracking performance similar to LQR controllers using neural networks and fuzzy logic, but due to the absence of formal stability and error handling documentation, such implementations cannot be recommended for use in real-world situations [132].

Table. 2 summarizes the advantages and disadvantages of the various controllers. It is difficult to make a direct comparison between overtaking manoeuvres since they are not standardized and each researcher demonstrates their tracking controller under unique conditions. To simulate an overtaking manoeuvre done at 120 km/h, [127] designed five different trajectory tracking controllers (the Stanley, the LQR, the SMC, the Fuzzy, and the MPC). In this setup, the control algorithms could be directly compared since they were applied to identical systems. To analyse tracking performance, lateral and angular errors were compared. During the manoeuvre, the steering angle was also used to compare actuation effort. These preliminary results (i.e., trajectory tracking and steering angle actuation) showed that MPC produced the smallest errors during tracking (i.e., lateral position and heading angle) with smooth steering angle actuation.

All the controllers discussed above have been validated in well-controlled environments where environmental uncertainty (e.g., headwind, tailwind, etc.) and parameter variations (e.g., vehicle mass, a moment of inertia, road friction, etc.) are limited. This allows researchers to benchmark different controllers, but most of these controllers are operational within a narrow operating window, which cannot be considered a realistic representation of driving in real life. To increase the operating window of a controller subject to large variations in system dynamics, three approaches can be taken: (i) ensure robustness; (ii) create a set of controllers to cover different operational regimes; and (iii) maintain performance by updating parameters continuously.

Table 2: Trajectory Tracking summary. Table from [105, 126, 128, 129, 133].

Trajectory Tracking algorithm	**Strength**	**Weakness**
Geometric and Kinematic	• Performance across a range of conditions that are not affected by disturbances (e.g., by wind, by road banks) (experimentally validated) • At moderate speeds (slow, but not jerky), the tracking system exhibits good performance and robustness	• Vehicle dynamics are not considered • High-speed driving leads to higher steady-state error (e.g., geometric errors) • Cannot be used at high speeds owing to the disregarded dynamics (e.g., kinematics) • Reference trajectories must be smooth and continuous
Classic	• Non-linear systems can be studied with an established method that performs well • A robust closed-loop performance (e.g., SMC) when dealing with uncertainties and noise	• Tuning controller parameters (like PID) can be challenging • Performs well only in some scenarios (e.g., SMC) • Path curvature variations affect control law (e.g., SMC)
Dynamic state feedback	• Calculate the control law by considering the dynamics of the vehicle • Optimization shifted off-line, resulting in simple control law implementation	• In order to obtain vehicle states (e.g., wheel forces, slip angles, torques, etc.), a complex procedure must be followed. • Path curvature variations affect control law (e.g., LQR) • The availability of state measurements (e.g., side slip angle) is critical for the computation of the control input
Neural Network	• The automated car can feel more natural with proper training to make the behavior very human-like	• It is necessary to simulate large amounts of real-world (training) data in order to tune controllers • It is impossible to explain failures
Fuzzy Logic	• Closed-loop systems operate much like human drivers (because their rules mimic human behavior)	• No formal stability analysis is conducted when tuning controllers • When there are a lot of variables, rules can get unmanageable
Model Predictive Control (MPC)	• An approach to systematic design • Constructive consideration of system and actuator constraints • Dynamics of vehicle and tire included in control problem	• In high-speed driving environments, non-linear MPCs are not suitable due to their high computing requirements • A prediction model's accuracy has a direct effect on tracking performance • Compared to industry standard PID, a larger range of tuning parameters is available

3 Result of research

This section summarizes and evaluates the main results from the six articles, answering the questions raised in Chapter 1.

3.1 Modular smart control system architecture for the AVs platform

Based on the functionality and robustness of the system, the modular hardware and software architecture were developed. The goal of the software and hardware architecture was to develop a stable, scalable, and easy-to-reconfigure system where every module and microcontroller had a specific task for controlling each unit (steering, driving motors, brakes) and different sensors across the platform. Two communication protocols were used in the whole control system: the Controller Area Network (CAN) bus, and Universal Datagram Protocol (UDP). Several factors played a role in choosing these protocols, including speed, reliability, and robustness. These protocols utilized CAN bus communications. CAN bus was selected due to its high communication reliability, real-time capabilities, and robustness. Due to its speed, efficiency, and reliability, UDP messages were used in communication between the master controller and the computer. Fig. 7 shows a SysML Block Definition Diagram (BDD) and Internal Block Diagrams (IBD) for a general modular architecture. Diagrams showing how messages are routed from the main computing unit to the lower-level controllers and main data flow parameters were shown.

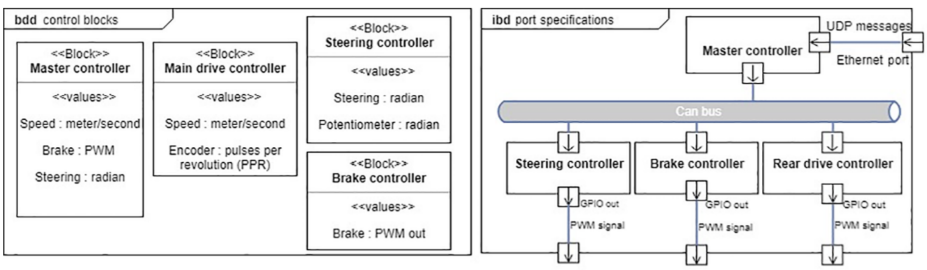

Figure 7: General modular architecture. Figure from article I.

Figure 8: Universal UGV platforms;1) Agronaut, 2) UKU. Figure from article I.

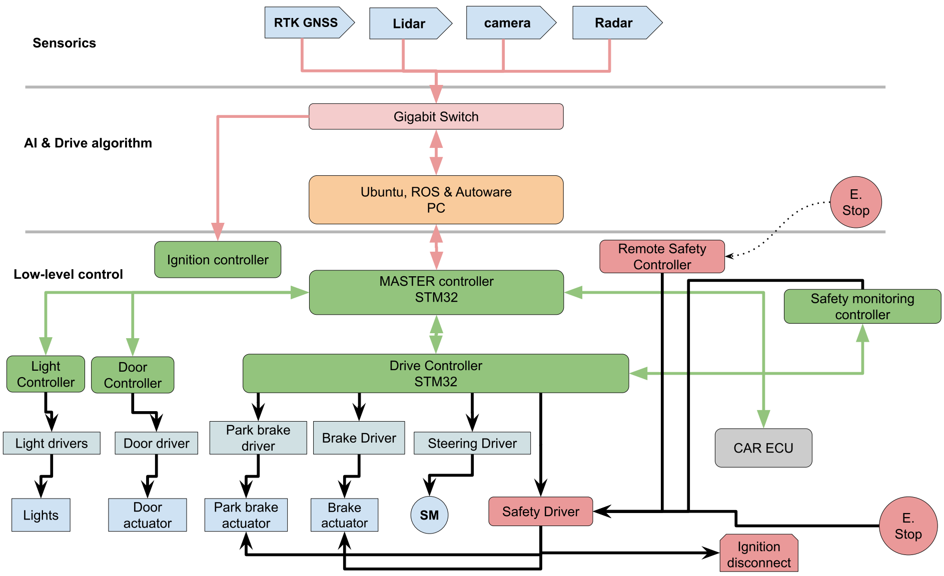

Figure 9: Hardware architecture implemented on the UGV UKU

3.1.1 Concept of modular hardware architecture

As described earlier, modularity is applied to the base platforms. In a mobile robot called UKU (Fig. 8), the architecture consists of three levels of hardware control and two levels of software control layers. In Fig. 9, the hardware architecture implementation is described in detail. Navigation and obstacle avoidance are handled by high-level sensors. While fulfilling its main purpose, the Robotic Operating System (ROS) middleware and open-source components run on the main computing unit, directly connecting the high-level sensors.

The master controller mediates driving commands for low-level controllers based on high-level ROS algorithms. The master controller assigns priorities and translates messages between the high-level and low-level controllers. Controllers at low levels are responsible for direct motor control, such as the PID controller and safety algorithms. Safety is controlled by a separate safety controller that is independent of both high-level and low-level controllers. When a problem occurs, it can stop the vehicle and monitor output and CAN messages. Furthermore, the system provides online data streaming between the server and the operator over 4G/5G mobile networks. To simplify, logically explain and make it configurable in future for different systems, the control software was designed similarly for each control unit of the mobile robot. For instance, the front and rear locomotion units of the robot have the same PID regulators with different controller inputs and coefficients.

3.1.2 Software architecture

The Robot Operating System (ROS) is the basis for the high-level software architecture of the system, shown in Fig. 10. Several factors led to this decision, including the availability of open-source drivers for multiple sensors, simplicity of integration of third-party software like Autoware [134] and Yolo [135], as well as multiple device drivers. Sensors

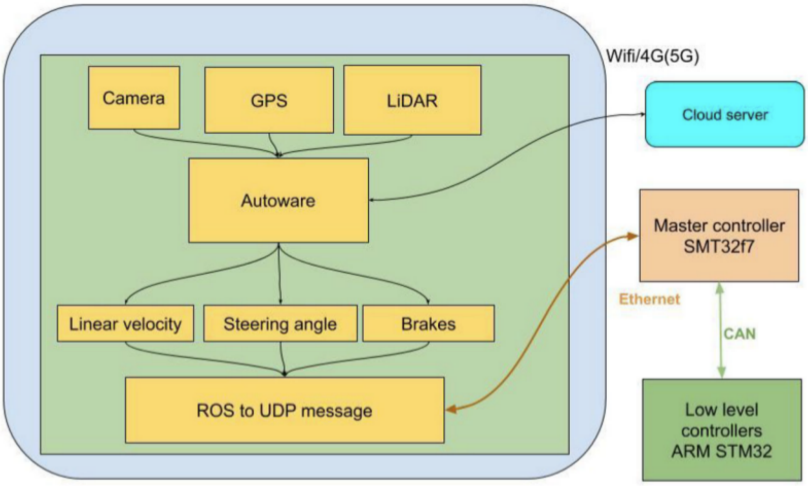

Figure 10: Software architecture and message flow. Figure from article I.

provide input to the mobile robot according to its current architecture.

Sensors provide several inputs to the current mobile robot architecture. The GPS is used for localization and path following. Safety and obstacle detection is handled by 2D (LiDAR) and camera inputs. Output commands are sent to the low-level controllers over UDP messages, which are steering angle, brake and linear velocity. Currently, Autoware [134] is the main software used for computation of the current architecture, which is a self-driving car open-source library. This library includes many advanced features such as lane following, obstacle avoidance, traffic light detection, lane detection, etc. ROS is a highly modular and scalable platform due to its master/slave architecture.

ROS communicates through the publish-subscribe method, which allows the running of separate nodes on different platforms that can easily interact with each other. The bridge converts ROS messages into UDP messages so that ROS can communicate with lower-level controllers. This high-level software architecture is modular due to key principles of ROS and the implementation of multiple software libraries as the building blocks of the primary product.

In this chapter, a modular architecture to achieve flexibility is proposed for off-road UGVs. Hierarchical hardware and software structures are specified, including three-level hardware and a two-level software structure, each of which can be adapted to the platform specificities. A practical example is shown in the first part of the paper using the off-road universal robot platform UKU. Moreover, another similar robot platform is considered to achieve comparable results. Although the two UGVs are similar in size and purpose, their power and locomotion methods are very different. A test drive of UKU provided a detailed evaluation of the hardware controller concept. The research demonstrates that the proposed architecture can provide stable navigation in dynamic and unknown environments while having fast and flexible implementation. Experiments with the modular control system show that the proposed modular architecture can easily be implemented for similar unmanned robots, such as Agronaut (Fig. 8). In this subchapter, **RQ1** was answered, and it was decided to use this control architecture on TalTech iseAuto for further development of the AV.

3.2 Human and Autonomous Vehicle Interaction (HAVI)

The main objective of this research is to propose a method for HAVI on the TalTech iseAuto platform. A matrix of RGB LEDs is used to implement sidebar lights where each individual LED is independently controlled. Therefore, the vehicle can easily switch its lights from red to white on both ends, as well as illuminate custom figures separately. HAVI devices communicate with humans mainly via LEDs.

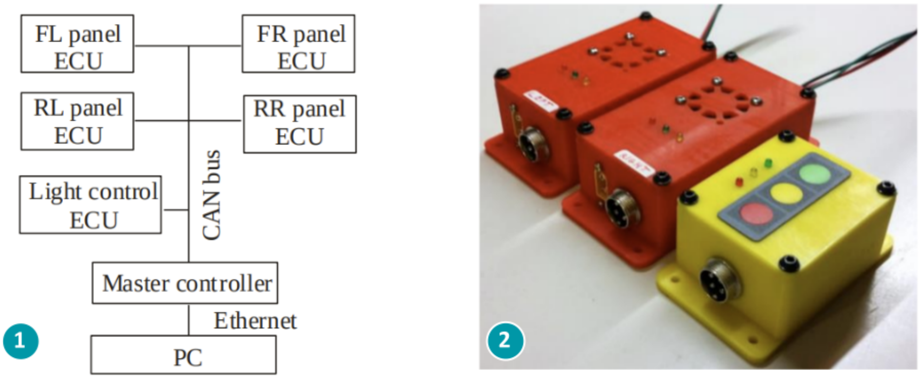

Figure 11: The hardware architecture (1) of two electronic control units (ECUs) with light control devices (2). Figure from article II.

In addition to the regular lights, the platform comprises lights designed to display different patterns for humans to give them a better notion of what the vehicle is doing or intends to do. It is achieved by using 512 red, green, and blue (RGB) light-emitting diodes (LEDs) as LED matrix modules.

Table 3: Illustration of the HAVI design pattern. Table from article II.

Pattern	Zebra Line	Arrows	Cross
Action	Pass	Pass	Stop
Visualization			

Light control ECU (Fig. 11) sends messages to the individual LED modules through a PC connected to a master controller. Based on the existing robot platform and the current research [136, 137], it is essential to further explore the interaction between humans and AV. The purpose of this experiment was to explore the following:

- What are people's opinions on the general safety of AVs?
- What do people think about their interaction with AVs?
- How to create a universal language that human-machine systems can use for interacting and communicating with each other?

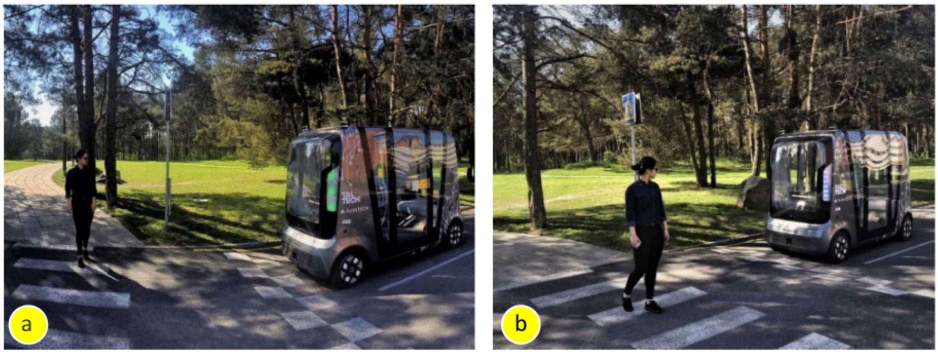

Figure 12: Experimenting with: Arrows pattern (a) and Zebra Line pattern (b). Figure from article II.

Due to the 16*8 pixels illumination of a single LED panel, three patterns were selected to be transferred onto the AV platform as shown in Table. 3. In Fig. 12, one can see the results of the designed line patterns. The arrow pattern (Fig. 12(a)) indicates a human should enter the crosswalk, whereas the zebra pattern (Fig. 12(b)) indicates that the human is crossing the road.

A questionnaire was created to collect feedback from humans that interacted with robot platforms, where most of the factors were derived from previous studies on AV.

In Fig. 13, you can see an experimental study plan for HAVI on an AV platform. Most of the test driving was conducted during the daytime on the roads of the university campus. The signs were marked near the zebra crossing at the intersection where people and vehicles interacted. When humans were crossing the road, randomly selected participants filled out a questionnaire. As the experiment was conducted on campus, the results were somewhat site-specific. HAVI provides an approach to improving road safety for AV by

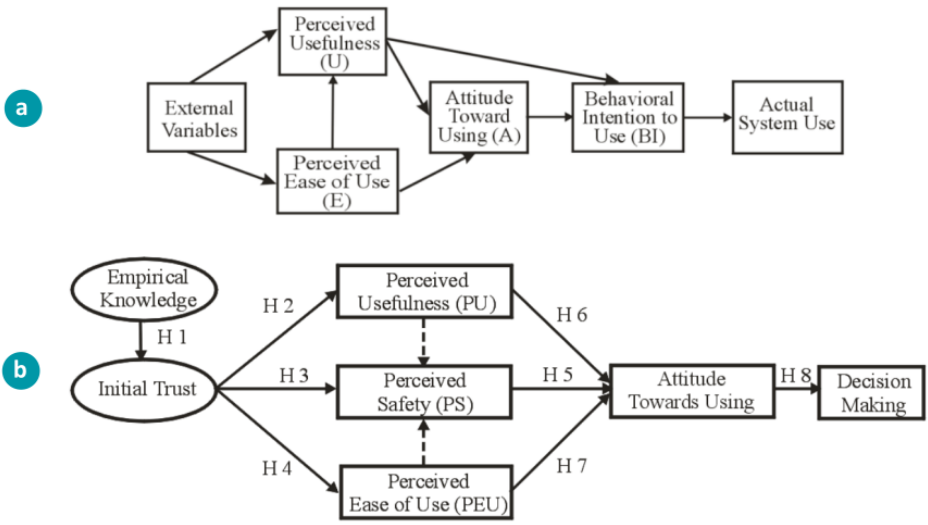

Figure 13: TAM's original contents (a). Figure from [138] and HAVI's model proposal (b). Figure from article II.

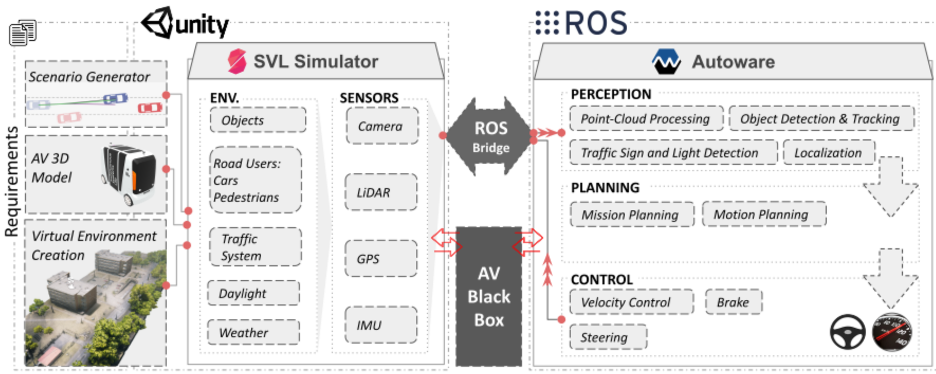

Figure 14: Safety validation workflow. Figure from article III.

taking advantage of its educational and research vehicle capabilities. It also allows developers to see people's reactions to the idea of creating a common language of interaction for humans and AV.

In practice, a remotely controlled multi- AV environment, such as found in [139], can also be used for larger mobile autonomous robots, such as those used in the HAVI experiment. To ensure the safety of robotised AVs, more effort and commitment are needed from all sectors. This paper aims to develop new on-vehicle light designs to tell humans about a robot's real-time decisions. Sensors on an AV should identify humans correctly and provide clear information about their movement in real-time. This approach partly answers the question raised in **RQ2** in this dissertation. To ensure better road safety, experiments were conducted using this architecture to determine an effective method of interaction between AVs and humans.

3.3 Safety Validation

Currently, autonomous cars are tested in three ways: by simulation, on the track, and on the road [140]. Simulation has proven to be much safer, cheaper, faster, and more reproducible than any other testing method [141] in all studies. This PhD thesis presents a simulation-based safety evaluation platform.

3.3.1 Safety toolkit for autonomous vehicle

Several open-source and proprietary simulators are available for AVs. There are low-fidelity and high-fidelity simulators, which provide different levels of simulation detail, based on the requirements of the user. Game-engine simulators, such as CARLA [30] and LGSVL [142], are based on Unreal and Unity engines, respectively, and use a powerful physics engine to simulate end-to-end systems. To achieve accurate and reliable results, the LGSVL simulator was chosen to enable controlling the ego with AV software on the vehicle. A ROS bridge was used to connect Autoware, the open-source ROS stack for AD to the simulator.

Fig. 14 gives an overview of the validation platform, including the requirements, LGSVL simulator, and Autoware stack, and shows how they relate. A simulation can be started when three requirements have been met. First, a simulation must have an execution plan. This step involves creating various conditions that endanger ego safety. Secondly, a 3D version of the ego is required to define realistic sensor configurations and to provide similar dimensions and dynamics. For a virtual simulation of real-world situations, a copy of the virtual environment is needed. Digital twins are a way of creating a replica of the ego and its working environment inside simulations, which will be discussed in the following chapter. Based on these requirements, the simulator runs predefined scenarios while providing sensor data for perception algorithms and getting control commands from an Autoware motion controller. Furthermore, a monitoring block records all the ego's actions and behaviour during each scenario for later analysis. This block is represented in Fig. 14 as a black box.

3.3.2 Safety evaluation through a high-fidelity simulation

Vehicle manufacturers use simulation extensively, especially for mechanical and dynamical analysis. However, the complexity of AVs makes it more challenging. Through simulation in complex scenarios and environments that include a variety of road users and sensors, decision-making algorithms can be verified. Gazebo is an example of a robot simulation platform. Based on ROS, it uses physics engines, and a variety of sensors appropriate for autonomous systems. Gazebo lacks modern game engine features like Unreal or Unity, which are capable of creating complex virtual environments and realistic rendering. As for CARLA and LG SVL, they are open-source simulators that are based on Unreal and Unity respectively, which are both compatible with AV stack Autoware. An overview of the simulation workflow and its relationship to Autoware is presented in Fig. 14. Simulator imports the vehicle 3D model and virtual environment, which are created in Unity. The simulator enables users to customise the environment, adjust the time of day and weather and add or remove other road users. Virtual sensors provide real-time information about the surroundings. In this study, the perception algorithms for localization and detection use this information via a ROS bridge. Autoware uses perception results in the planning section for vehicle control commands. To navigate the vehicle, the ROS bridge sends control commands back to the simulator.

To map an area, aerial imagery from a drone needs to be taken. This is done by flying in a grid pattern. It ensures that the sides of a subject are captured. The flight path is followed three times at constant altitude but at different camera angles to ensure maximum coverage. Aerial photography is an essential component of mapping, as it will determine the outcome of the process as well as the amount of work to be done to process those images. Additionally, external factors could affect the quality of the images taken from the ground. Lighting conditions and weather conditions can affect the quality of photos, which may interfere with photogrammetric processes. An RTK device can be used to mitigate error and shift in positioning data, if necessary, since drone images are georeferenced. By using the IMU onboard, the images can be oriented for later stitching together and processing to produce photogrammetric data. From the captured images, third-party software creates a dense point cloud. To remove unwanted objects and vegetation from the dense point-cloud, segmentation and classification of the points are required. As shown in Fig. 15, there are three main steps involved in generating the Unity train from geospatial data.

Using the simulation architecture shown in Fig. 14, the AV can run inside a virtual environment that was developed as a safety validation environment for the TalTech campus pilot road with Florida Polytechnic University and Embry Riddle Aeronautical University.

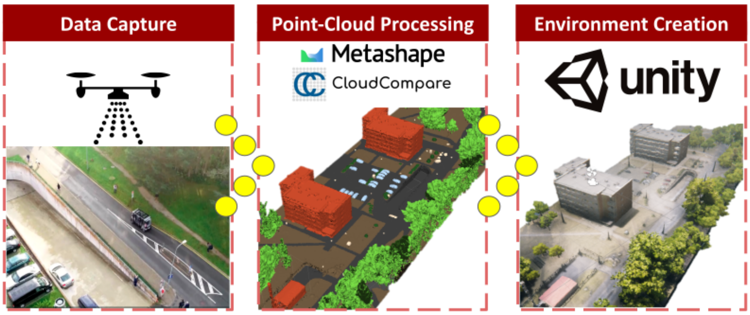

Figure 15: Creating a virtual environment. Figure from article III.

A high-fidelity simulator now allows evaluating the performance and safety of the control algorithm by simulating different scenarios close to reality. To accomplish this, LGSVL provides a Python API for spawning different objects, including pedestrians and cars, within the virtual environment. TalTech iseAuto faces an NPC vehicle that has appeared in front of the AV in Fig. 16. LGSVL is represented in Fig. 16(a), while Ros Visualization (RViz) is displayed when LiDAR is used in Fig. 16(b). Overtaking is a challenging aspect of algorithms for self-driving vehicles. A study is being conducted to find out how the AV should decide on this mission and the risks it faces. Trying to pass an object or an NPC has brought this topic into focus. Simulators can help improve both the perception and detection systems, as well as the mission and motion planning for a safe overtake. With the help of this testing scheme, the safety and performance of Avs can be improved.

3.4 Overtaking for AVs

Overtaking is one of the greatest challenges for autonomous vehicles due to its riskiness. The final objective of this chapter is to identify ways to improve the safety of this manoeuvre in autonomous vehicles.

3.4.1 Optimized sigmoid based overtaking

The research describes an overtaking manoeuvre that uses smooth sigmoid curves to improve manoeuvring by using a two-phase overtaking. Based on perception, optimal low-level steering, and trajectory planning parameters, a sigmoid function is created based on the AV kinematic model for fast, smooth and safe generation of overtaking paths.

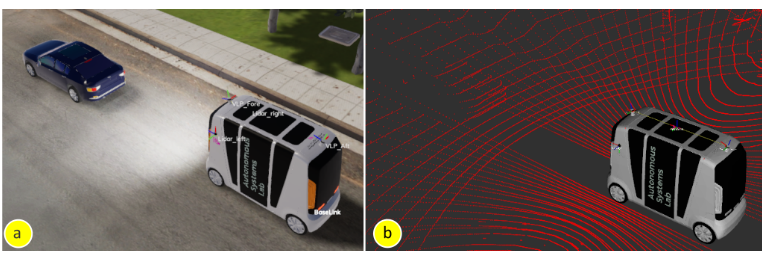

Figure 16: LGSVL environment (a), point-cloud visualization in RViz (b). Figure from article III.

Fig. 17 depicts the two phases of the overtaking manoeuvre: (I) lane change and passing obstacles, and (II) return to the original lane and continuing. Furthermore, this paper presents a high-fidelity simulation testbed for verifying algorithm performance on the AV before implementing it on the actual AV.

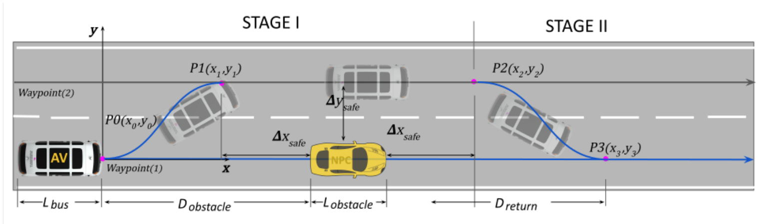

Figure 17: A description of an overtaking manoeuvre for a static obstacle. Reference points for each sigmoid path are $P_0 - P_1 - P_2$ and $P_2 - P_3 - P_4$ respectively. Figure from article V.

The TalTech iseAuto was used in a long-term experiment. The research's contributions of our work are as follows:

- A method that allows for fast overtaking decisions and path planning within seconds.
- A method that illustrates a safe generated path via a verification process.
- A human-machine interface communicating overtaking intentions.
- A method for safe overtaking by generating a traffic-law compliant path that is validated via simulation and then implemented using a real AV demonstrating its safety.

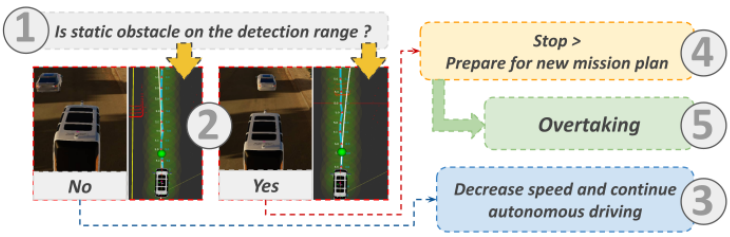

Figure 18: Five decision-making steps of the overtake algorithm in the case an obstacle was detected by the AV. Figure from article V.

The first step in overtaking is to detect stationary objects blocking the desired trajectory to the next waypoint. In some cases, the object stands on beside the road, but in others, it is directly blocking the road lane. If the object blocks the desired trajectory and it is in the detection range, the AV must stop. The detection range is approximately rectangular and predefined along the waypoint. It is bigger than the width of the vehicle in general (green area in Fig. 18 step 2). All objects detected in this area make the AV stop. When an obstacle is detected outside the AV detection range, the AV lowers speed to be prepared to stop immediately. The AV continues driving if the object remains outside the range. Fig. 18 illustrates these steps.

One of the main limitations of the current overtaking algorithm is ensuring the smoothness of steering angle changes while preserving the kinematic feasibility of generated

Table 4: Kinematic parameters used in the optimisation and experimental setup . Table from article V.

Param.	Val.	Description	Param.	Val.	Description
$D_{obstacle}$	8-15 m	Distance to obstacle	$L_{obstacle}$	4 m	Obstacle length in x-dir
W_{road}	6 m	Road width	$W_{obstacle}$	2 m	Obstacle width in y-dir
Δx_{safe}	3-5 m	Safety distance in x-dir.	$D_{ret.1\&2}$	5-15 m	Travel distance to the initial lane
L_{bus}	3.4 m	Shuttle length	$Lo_{distance}$	3-8 m	Pure Pursuit Look-ahead distance
L_{wheel}	2.55 m	Shuttle wheel base	Δy_{safe}	2 m	Safety distance in y-dir
v	10 km/h	Shuttle constant speed	$k_{abruptness}$	3-8	abruptness factor

paths. It is necessary to ensure passengers' comfortable riding experience and to prevent steering motor overload. One approach to ensuring smooth turning and steering during an overtaking manoeuvre involving two lane changes (see Fig. 17) is to use mathematically defined sigmoid functions. As the first step in sigmoid path generation, STAGE I in Fig. 17, is described by Eq. 1, including the exponential term:

$$y_{sig2}(x) = y_2 - a(y_4 - y_2) + b\frac{y_4 - y_2}{1 + e^{\frac{x_{23}-x}{D_{23}}}} \tag{1}$$

where y_0 is the actual y-coordinate of stage 1 starting point, y_2 is the desired final y-coordinate of stage 1, the sigmoid centre point x-coordinate $x_{01} = \frac{x_1+x_0}{2}$ is defined by midpoint between points P_1 and P_0, and sigmoid abruptness $D_{01} = \frac{x_1-x_0}{k}$ is calculated by the difference of the x-coordinates of points P_1 and P_0. Analogously, the sigmoid trajectory for the second stage stated by the Eq. 2:

$$y_{sig2}(x) = y_2 - a(y_4 - y_2) + b\frac{y_4 - y_2}{1 + e^{\frac{x_{23}-x}{D_{23}}}} \tag{2}$$

where y_2 is the actual y-coordinate of stage 2 starting point (endpoint of previous stage 1), y_4 is the desired final y-coordinate of stage 2, the sigmoid centre point x-coordinate $x_{23} = (x_2 + x_3)/2$ is defined by midpoint between P_2 and P_3, and sigmoid abruptness $D_{23} = (x_3 - x_2)/k$ is calculated by the difference of the x-coordinates of points P_3 and P_2. It may be noted that abruptness parameter definition could also use a larger denominator (e.g., 8), but then more rapid heading angle changes are expected in the middle of stages. However, the connection of consecutive steps should be smoother.

Finding the best parameter values for the sigmoid curve is essential to assess the sigmoid curve utilisation, as well as to attain a comfortable ride with accurate trajectory tracking (smooth steering). That way, an overtaking trajectory has been simulated using a virtual model provided by MATLAB and a GA optimisation algorithm has been used to find the best fit values. The kinematic model of the automated vehicle and the navigation controller simulation was done with the "Vehicle Body 3DOF Dual Track" block.

From the curve definition (Table. 4 includes all curve parameter values and description information, including the ranges of validity), the following five parameters were chosen: abruptness factor $k_{abruptness}$, distance to the obstacle D_{obs}, the safety distance along the longitudinal direction Δx_{safe}, as well as the two longitudinal travel distances to return to driving lanes D_{ret1}, D_{ret2}(see Fig. 17). In addition, the pure-pursuit controller's lookahead distance parameter $Lo_{distance}$ was included in the optimisation process to increase the trajectory following accuracy. By minimising the following error, precise movement can be achieved, which is crucial for manoeuvres such as overtaking. Using 500 simulation runs, GA found the optimum values shown in The parameter values listed in Table. 5 were suggested as initial values to better understand the optimisation result. Following that, two simulations using the initial and optimal data were run, while steering changes and trajectory tracking were recorded.

Table 5: Initial and optimized parameter values. Table from article V.

Param.	Init. Val.	Optim. Val.	Param.	Init. Val.	Optim. Val.
$Lo_{distance}$	5 m	3.88 m	Δx_{safe}	3 m	3.49 m
$k_{abruptness}$	10	3.62	$D_{ret.1}$	13 m	13.04 m
$D_{obstacle}$	13 m	14.28 m	$D_{ret.2}$	13 m	10.93 m

In Fig. 19, the performance of sub-optimal and optimised parameters for tracking and steering were conpared. Fig. 19(a) illustrates the trajectory that was not optimized (red dots) and how the vehicle followed the path (black circles), whereas Fig. 19(b) depicts the optimized trajectory.

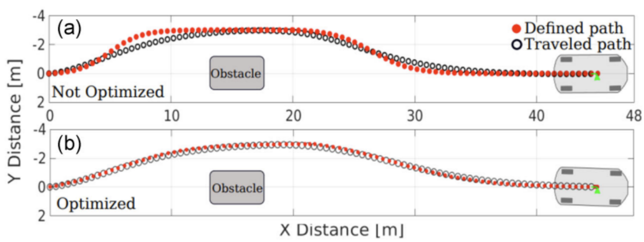

Figure 19: A comparison of trajectory following a not-optimized(a) and a optimize(b). Figure from article V.

It is evident that the AV followed the optimized trajectory more precisely and that the steering changes were smoother (see Fig. 20).

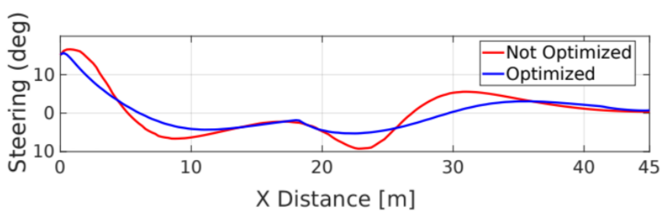

Figure 20: Comparison between the optimized and non-optimized steering angles. Figure from article V.

Figure 21: a) The LGSVL simulator simulates an overtaking scenario. b) A ROS visualization shows vehicle sensors, waypoints, and position on a map. Figure from article V.

LGSVL and CARLA are two of the realistic car simulators powered by AV control software. By using modern game engine features like Unreal and Unity, they can create complex virtual environments and render them realistically.

A sigmoid-based overtaking algorithm is evaluated using the LGSVL simulator. The simulator runs on the Unity game engine that provides a variety of environments and car models. A detailed TalTech iseAuto 3D model with the LiDAR inside Unity was implemented and the engine for evaluating the manoeuvre in the simulator was assigned. Within the LGSVL simulator, a simple overtaking scenario was created (see Fig. 21). This scenario involves placing an NPC car in the middle of a waypoint and observing how AV decides. In Fig. 21(b), the ROS visualization software displays the point cloud of the simulated environment, as well as the AV's desired straight path. In Fig. 21, the AV detects an object within the detection range of its waypoint (green area) and decides whether or not to overtake the object. The red line is displayed as a stop indicator before the NPC. Fig. 21(a) shows the simulation environment in LGSVL, including a stopped NPC and the TalTech iseAuto 3D model.

In the next step, the AV generates a smooth waypoint based on the sigmoid curve. As shown in Fig. 22, four-time instants (frames) are shown from the start of the overtaking

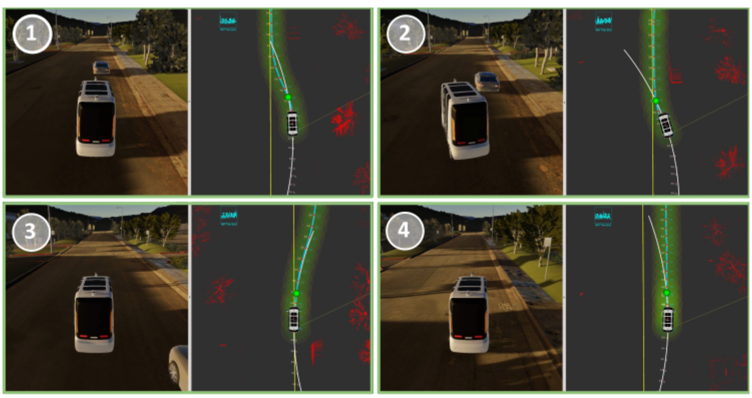

Figure 22: Four different time instants in LGSVL and corresponding ROS simulation environment illustrate the smooth sigmoid-based overtaking manoeuvre. Figure from article V.

operation through to its end inside the simulator and the corresponding ROS visualization.

As illustrated in Frame 1, the new path, presented by Eq. 1, is generated for STAGE I of overtaking Fig. 22. In frame 2, the AV moves toward the new waypoint. By using Eq. 2, the AV generates a new path for STAGE II of the mission after passing the object. Finally, it returns to its original waypoint in the last frame.

A major practical problem in the implementation of different overtaking algorithms and scenarios was ensuring smooth steering angle changes to avoid irritating passengers and overloading the steering motor. The summary of the performed sigmoid-based steering simulation is presented in Fig. 23. This result confirms that the desired smooth changes of steering angle may be achieved and the maximum values of necessary steering angles remain below 8°.

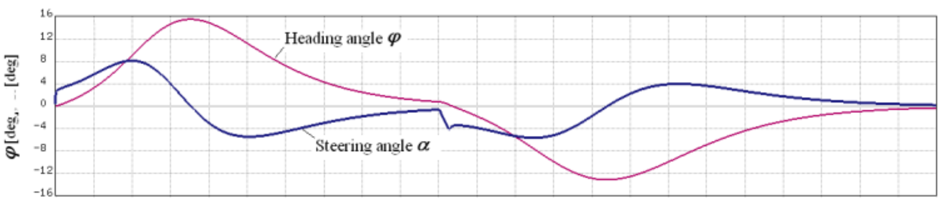

Figure 23: The simulation results of an overtake process following the mathematically postulated sigmoid curves: the relevant heading angle and necessary steering angle values as functions of shuttle front end x-coordinate.

Fig. 24 summarizes the results from simulation and experimental setups to validate the optimized sigmoid-based paths. In both simulations and experiments, the error is below 10%, proving the effectiveness of the method. As a result of the proposed optimized sigmoid-based method, steering angles change more smoothly than with a guided hybrid A-start, since the spikes visible on the blue-line at times 0 and 12 sec have been eliminated shown in Fig. 24(a).

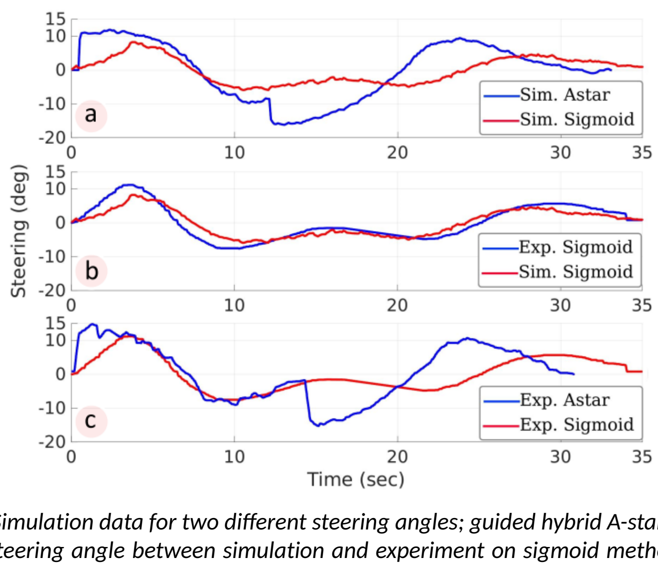

Figure 24: (a) Simulation data for two different steering angles; guided hybrid A-star and optimized Sigmoid. (b) Steering angle between simulation and experiment on sigmoid method. (c): Results from steering experiments using both methods. Figure from article V.

A simulation result is shown in Fig. 24(b), along with the corresponding experimental data. The use of the steering motor is improved due to a smoother trajectory since the range of angles can be reduced by about five degrees. Using this approach will reduce the long-term consumption of the motors, guarantee a long-time high performance, and prevent unexpected failures.

The results prove that the experimental setup is reliable in terms of the simulation environment. The evaluation of the overtaking manoeuvre could be performed via a simulation platform rather than directly on the AV, i.e., it is not essential to test newly developed algorithms directly on the AV. Overtaking the stopped car scenario for this study may not have included all the complex situations, but it is an important step in establishing a verification platform for future studies. Scenarios such as moving objects and vehicles coming from the other direction can easily be created and tested in the simulation without any risk.

A sigmoid-based overtaking manoeuvre generator for an AV was experimentally validated in this study. To overcome the limitations of the current state-of-the-art algorithms, a modified path planning algorithm based on the sigmoid curve is proposed. The overtaking process for TalTech iseAuto was designed with smoothness, safety, and reliability as key values. Developing, formulating, and implementing the proposed overtaking algorithm on a real AV and conducting experiments on real roads were the two objectives of the study. Overtaking multiple vehicles using the proposed method is also possible. The proposed algorithm was validated using high-fidelity simulations. According to the results, the proposed method was effective at reducing steering effort while eliminating unpleasant and unsafe operations. The proposed method outperforms other techniques, such as hybrid A* [52].

3.4.2 Model-based LQR control of an overtaking manoeuvre

To execute desired manoeuvres safely, automatic control is one of the most important sub-tasks. As a result, mathematical kinematic and dynamic vehicle models are required. This implies that motion planning and vehicle control are two different but closely related actions. The first step is to compute a feasible trajectory (from the perspective of the vehicle's dynamics) for the vehicle while accounting for the surrounding obstacles such as other vehicles, pedestrians, and non-moving objects. The second action involves guiding the actuators, such as the acceleration and steering, to follow the trajectory generated by the motion planner, while maintaining the stability of the system and, if possible, a smooth drive. Two variables determine the complexity of a vehicle guidance control problem: the type of control (lateral, longitudinal, or mixed), and the type of model to be controlled (kinematic, linear dynamic, nonlinear simplified dynamic, or non-linear dynamic). The research addresses one of the most complex configurations - a non-linear dynamic problem. A linearized model with a linear quadratic regulator (LQR) that effectively considers the original nonlinear dynamical model of an AV is proposed and simulated. As a result of the simulation results, the lateral y coordinate stabilization task performed well. In addition, this LQR approach, which utilized Simulink/MATLAB standard tools, demonstrated low computational costs, which allows for real-time applications.

AV was modelled using the dynamic bicycle model to balance accuracy with model complexity and computational costs, see Fig. 25.

In this model, the front and rear wheels of a four-wheel vehicle have been replaced with a single (mass-free) front and rear wheel located at the longitudinal axis of the vehicle, thereby simplifying the design of the vehicle. The kinematic model of the AV can be

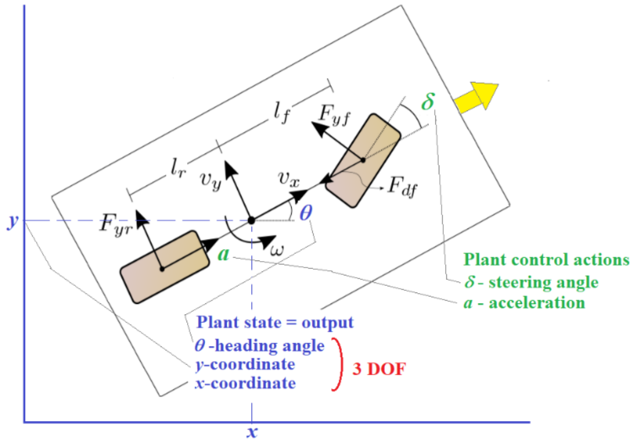

Figure 25: Bicycle model of the autonomous vehicle. Figure from article VI.

formulated as following Equations:

$$\dot{x} = v_x \cos\theta - v_y \sin\theta \tag{3}$$

$$\dot{y} = v_x \sin\theta - v_y \cos\theta \tag{4}$$

$$\dot{\theta} = \omega \tag{5}$$

The coordinates (x, y) denote the location of the centre of mass of the vehicle in the earth-fixed frame, which is the yaw angle of the vehicle (or the heading angle), the velocities v_x, v_y denote the longitudinal and lateral speeds of the body frame, respectively, and is the yaw rate. Here are the equations that describe the dynamics of the analysed AV:

$$m\dot{v}_x = F_x + m v_y \omega \tag{6}$$

$$m\dot{v}_y = -m v_x \omega + 2(F_{yf} \cos\delta + F_y r) \tag{7}$$

$$I\dot{\omega} = 2(l_f F_{yf} \cos\delta - l_r F_{yr}) \tag{8}$$

The mass of the vehicle is denoted by m and its yaw inertia by I; the lateral tire forces acting on the front and rear wheels, respectively, are given by F_{yf} and F_{yr}(in coordinate frames aligned with the wheels), the force of the vehicle is given by F_x, the direction of the front wheel axis is determined by l_f, and the distance between the centre of gravity and the wheel axes is determined by l_r. Because one wheel is used instead of two, the forces acting on the wheels are multiplied by two. The lateral tire forces F_{yf} and F_{yr} acting on front and rear wheels.

$$F_{yf} = C_f \left(\delta - \tan^{-1} \left(\frac{v_y + l_f \dot{\theta}}{v_x} \right) \right) \tag{9}$$

$$F_{yr} = -C_r \tan^{-1} \left(\frac{v_y - l_r \dot{\theta}}{v_x} \right) \tag{10}$$

For the calculation of the constants C_f and C_r it is necessary to consider the difference between real turning trajectories and idealized trajectories of free rolling.

The parameters of TalTech iseAuto are as following: $m = 1160\ kg$, $l_f = 1,275\ m$, $l_r = 1,275\ m$. Based on TalTech iseAuto length, $l_x = 3,6\ m$ and width $l_y = 1,5\ m$ then $I = 1470.3\ kg/m^2$. For calculating C_f and C_r, a straight track with a small yaw angle at steady state, we can estimate v_x const, v_y = 0. with a small steering angle at steady-state with small-angle approximation ($\tan^{-1}(\tan x) = x$) and For a straight track with a small yaw angle and small steering angle at steady state $\cos\delta \approx 1$, $\dot{\omega} \approx 0$. When $v_{xmax} = 50\ km/h$, we can obtain the TalTech iseAuto tire stiffness coefficient estimations $C_f = 43875\ N/rad$, $C_r = 43875\ N/rad$.

To implement the LQR-controller, first, it is necessary to formulate a linearized model with a conventional system and input matrices A and B, as well as state, output and input matrices X, Y, and U. To begin, let us take a look at how velocity V_x is defined as a following Equation:

$$\dot{v}_x = a + v_y\omega \tag{11}$$

V_c is a given speed. Now we can linearize the Eq. 3 and Eq. 4 with the nonlinear functions in the Taylor series we have :

$$\dot{x} = V_c - v_y\theta \tag{12}$$

$$\dot{y} = V_c\theta + v_y \tag{13}$$

with $\cos\delta \approx 1$:

$$\dot{v}_y = \frac{2C_f}{m}\delta - \frac{2(C_f + C_r)}{mv_c}v_y + \left(\frac{2(C_r l_r - C_f l_f)}{mV_c} - V_c\right)\omega \tag{14}$$

$$\dot{\omega} = \frac{2C_f l_f}{I}\delta + \frac{2(C_r l_r - C_f l_f)}{IV_c}v_y - \left(\frac{2(C_f l_f^2 + C_r l_r^2)}{IV_c}\right)\omega \tag{15}$$

The conventional matrix of State-space can be presented as follow:

$$\dot{X} = AX + BU \tag{16}$$

$$Y = CX + DU \tag{17}$$

The control vectors as follow:

$$X = \begin{bmatrix} y \\ v_y \\ \omega \\ \theta \end{bmatrix}, U = \delta \tag{18}$$

And the system matrix A is :

$$A = \begin{bmatrix} 0 & 1 & 0 & V_c \\ 0 & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{19}$$

The matrix elements from Eq. 14 and Eq. 15 are:

$$a_{22} = -\frac{2(C_f + C_r)}{mV_c}, a_{23} = \frac{2(C_r l_r - C_f l_f)}{mV_c} - V_c \tag{20}$$

$$a_{32} = \frac{2(C_r l_r - C_f l_f)}{IV_c}, a_{33} = -\frac{2(C_f l_f^2 + C_r l_r^2)}{IV_c} - V_c \tag{21}$$

The control matrix B in Eq. 16 is :

$$B = \begin{bmatrix} 0 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \tag{22}$$

with the following elements:

$$b_2 = \frac{2C_f}{m}, b_3 = \frac{2C_f l_f)}{I} \tag{23}$$

To complete the state-space matrix we need matrix C and D in Eq. 17 as follow:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{24}$$

with minimizing the cost function:

$$J = \int \left(X^T QX + U^T RU \right) dt \tag{25}$$

Optimal control refers to finding a controller that provides the best performance about some given cost function. In this case, we have a continuous-time LQR optimal control problem where the mathematical model of the controlled object is linear and the functions in the cost function are quadratic. For minimizing the cost functions in Eq. 25, the following rules are applied to matrix Q,R in Eq. 25 :

$$Q_{ii} = \frac{1}{x_{imax}^2} \tag{26}$$

$$R_{ii} = \frac{1}{u_{imax}^2} \tag{27}$$

where $x_i max$ and $u_i max$ are the maximum value for output and input signals. Finally by combination of Eq. 17, Eq. 18, Eq. 28, Eq. 26, Eq. 27 we have:

$$Q = \begin{bmatrix} \frac{1}{y_{max}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{v_{ymax}^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\omega_{max}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\theta_{max}^2} \end{bmatrix}, R = \begin{bmatrix} \frac{1}{\delta_{max}^2} \end{bmatrix} \tag{28}$$

In the proposed method $y_{max} = 5m$, $v_{ymax} = 1km/h, \omega_{max} = \frac{V_c}{(l_r + l_f)} = 1,0893 rad/s$, $\theta_{max} = \pi/2 rad$ and $\delta_{max} = \pi/4 rad$.

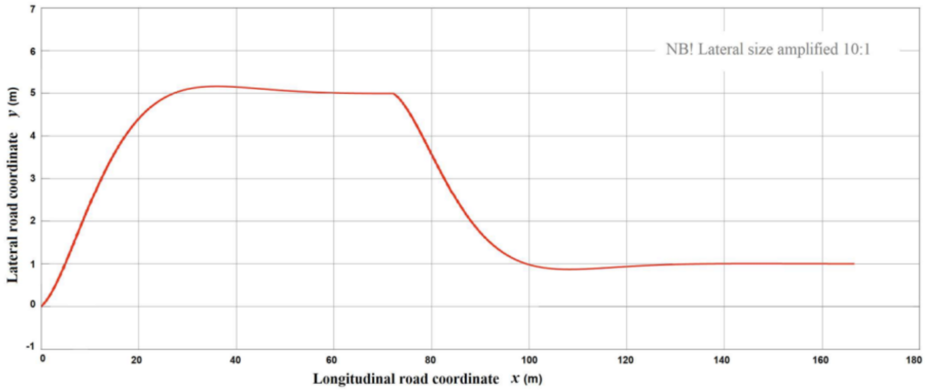

Figure 26: A calculated trajectory consists of two steps: a left shift of 5 m from the initial lane y = 0, followed by a right shift of 1 m back to its original lane y = 0. Figure from article VI.

Fig. 26 illustrates the resulting trajectory of AV. As the output signal approaches the desired line y_{ref1}, y_{ref2} there should be a smooth transition with a small amount of overshoot of 2-3%.

This method described a simulation and modelling technique for a self-driving car of $1160\ kg$ weight. A 2-stage (overtake and return) obstacle avoidance manoeuvre was performed at a constant speed while the steering angle was controlled. Using the Simulink/MATLAB environment, to demonstrate the use of an LQR controller for this manoeuvre. Using LQR controllers is a promising approach, since they are optimal by default, and consider the costs of different characteristics of the vehicle movement.

4 Discussion

Theoretical considerations are mostly related to the methodology and equipment used to conduct further research in the field and laboratories. It is described in a way that enables other researchers to extend it in the field and replicate the primary research findings. In Article I, the methodology for creating a modular smart control system for autonomous UGV as well as the hardware and software architecture of this system were discussed. Researchers can use this study for developing autonomous vehicle software and hardware architectures. A major benefit of Article I is the modular architecture, which is flexible and fast to implement for off-road unmanned ground vehicles. An architecture is proposed with hierarchical hardware and software layers, including three-level hardware and two-level software architecture to be adapted to any AV platform.

Article II is a site-specific HAVI experiment that enables researchers to get an overview of how people react to the concept of creating a common language for the interaction between humans and AV. The objective of this paper was to develop a novel approach to vehicle lights that can inform humans of real-time AV decisions for ensuring safety. Researchers can use this research and extend the idea to find the best method for interaction between AV and pedestrians on the roads. In practice, the method in article II can be implemented for different autonomous vehicle platforms used in HAVI experiments.

Articles III and IV focus on the safety validation of AV developments. Article III primarily focus on creating a digital twin as a virtual environment for safety simulations. For simulations to be reliable, a digital twin of the AV solution set-up is needed. Local governments and other stakeholders interested in deploying AVs on their streets or dedicated areas can use this guide to describe the different tools and processes outlined in the paper to build a safety toolbox. Customers or research institutions can use this toolbox to first simulate their solution and then test it in real-time.

Article IV presents simulation as a validation approach that is practical and effective for assessing safety at different levels. As an example, it is shown how the virtual environment, vehicle model and Autoware are used to simulate different scenarios using TalTech iseAuto with SiL testing. To demonstrate the reliability of the control algorithm, two overtaking scenarios were studied, and its safe performance was evaluated. This testing scheme will enable a considerable improvement in safety and the performance of AVs owing to its development and utilization. Articles III and IV have the following practical outcomes:

- Creating a virtual environment based on the real experiments around the AV testbed.
- Modelling AV into the simulator with the exact sensor positions.
- Constructing different scenarios and performing software-in-the-loop simulations through Autoware.
- Simulating experiments using a digital twin as a model of the real environment.

Article V and VI proposed an overtaking manoeuvre on AVs based on Linear Quadratic Regulator (LQR) and sigmoid function. An optimized sigmoid-based overtaking algorithm was presented in article V, while considering smoothness, reliability, and safety. It is possible to extend the overtaking methodology to multiple vehicles. The proposed algorithm was validated with high-fidelity simulations and its behaviour was predicted. Results showed that the suggested method reduced steering effort and eliminated abrupt changes that led to unsafe and uncomfortable operations. In article VI, numerical simulations using Simulink/MATLAB were used to demonstrate the practical application of LQR

(Linear Quadratic Regulator). It is a promising idea to use an LQR controller because it is optimal by default and takes into consideration the different characteristics of the movement of the vehicle. Using the LQR controller in AVs may lead to a reduction of power used by the steering engines during manoeuvring. The Articles V and VI enabled smooth, safe, and reliable overtaking on real AV (TalTech iseAuto). The overtaking method is based on AV dynamics. Therefore, researchers and companies that design AVs can employ this method in developing their products.

5 Conclusion and future research

5.1 Conclusion

AD will revolutionize automobiles by improving passenger comfort, safety, and convenience but an AD system needs to overcome several challenges before becoming a reality. The primary goal of this dissertation was to propose useful methods for meeting these challenges. To define the main contributions of the dissertation separately at theoretical and practical levels - as discussed and summarized in chapters 3 and 4, the following achievements can be outlined:

- A flexible open-source-based modular control system architecture that can be applied to various kinds of autonomous platforms like AVs, mobile robots, and unmanned ground vehicles. The hierarchical architecture includes three-level hardware and two-level software architecture. The experimental results on two different UGV platforms, and TalTech iseAuto show that the proposed architecture meets main driving requirements and covers safety and reliability aspects.
- A new method for HAVI experiment for safety improvement has been proposed. This method creates a common language for interaction between the AV and pedestrians.
- Creating a virtual environment based on the AV road area and performing SiL by connecting Autoware with LGSVL enables finding a better sensor configuration and safety verification. By using simulation as a validation approach, the research provides a practical and effective technique to evaluate safety at various levels. By utilizing this testing scheme, AVs will be able to improve their performance and safety.
- Creating a digital twin as a base virtual environment for safety evaluation.
- Implementing a simple and fast overtaking method on the real AV to guarantee a smooth, safe, and reliable overtaking manoeuvre. High-fidelity simulations were used to validate the efficiency of this method. The experimental results prove that this method enables an AV to overtake multiple vehicles, reduces steering effort, and has a reliable performance over other state-of-the-art methods like Hybrid A*.
- Simulating and offering a control system for a linearized model using a Linear Quadratic Regulator (LQR) that is designed to take effectively into account the original nonlinear dynamical model of an AV (TalTech iseAuto).

The results of this dissertation were used to improve the safety and reliability of the TalTech iseAuto. However, the modular architecture proposed in Article I will be improved over time, and the optimized modular control system architecture is developed for the TalTech iseAuto. This hardware and software architecture is still under improvement. To evaluate the safety of planning algorithms on the TalTech iseAuto, there is a digital twin of the TalTech campus and an TalTech iseAuto 3D model. Overtaking manoeuvres will be included in the operations of TalTech iseAuto.

5.2 Future work

Future investigations are recommended to focus on advanced methods drawn from this dissertation:

- Modern modular architecture, including high-level control algorithms incorporating AI in planning and navigation layers.
- Improving visual signalling by a combination of audio and lights or different channels for HAVI in complex scenarios.
- The proposed overtaking methods showed that the current behaviour must be significantly expanded to handle more advance and complex scenarios, e.g., overtaking from moving vehicles, and aborting the overtaking manoeuvre in unsafe situations. One needs to identify all possible cases for optimal decision-making and develop heuristic-based rules based on deep domain knowledge. However, it may fail in highly unpredictable but possible scenarios. Possible replacements for heuristic rule-based are MDP or RL based methods. These methods are better suited to capturing the probabilistic nature of the decision-making process and enable learning from data. The probabilistic nature of decision-making can be better captured through these approaches and data can be used to learn from it.
- The AV speed is considered constant during the experiments. It should be possible in the future to change this parameter and study the effect. In addition, the test can be performed by comparing relative speeds between the AV and NPC (between 15 and 40 km/h).
- A high-level decision-making block, developed in collaboration with Aalto University in Finland and Nagoya University in Japan will be added to an open-source planning algorithm for automated vehicles. In future work and research, to prepare a reliable driving for AV, not only the AV speed, several parameters such as the speed of NPC, the distance between vehicles, and safe and smooth path for trajectory, etc. can be considered.
- Simulators are the best environment to test AVs. These environments can be improved by using different high-fidelity simulators like CARLA that use the Unreal engine.
- Moreover, an interesting topic to be explored in future research is the creation of different scenarios similar to those typically encountered in real-life situations for evaluating AV performance.

References

[1] Dorina Pojani and Dominic Stead. Sustainable urban transport in the developing world: Beyond megacities. *Sustainability*, 7(6):7784–7805, 2015.

[2] Hongbo Gao, Xinyu Zhang, Yuchao Liu, and Deyi Li. Cloud model approach for lateral control of intelligent vehicle systems. *Scientific Programming*, 2016:1–12, 2016.

[3] Xiaoxiang Na and David J Cole. Linear quadratic game and non-cooperative predictive methods for potential application to modelling driver–AFS interactive steering control. *Vehicle System Dynamics*, 51(2):165–198, 2013.

[4] Travis J Crayton and Benjamin Mason Meier. Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy. *Journal of Transport & Health*, 6:245–252, 2017.

[5] Shanshan Zhao and Aemal Khattak. Motor vehicle drivers' injuries in train–motor vehicle crashes. *Accident Analysis & Prevention*, 74:162–168, 2015.

[6] Ana M Novoa, Katherine Pérez, Elena Santamariña-Rubio, and Carme Borrell. Effect on road traffic injuries of criminalizing road traffic offences: a time–series study. *Bulletin of the World Health Organization*, 89(6):422–431, 2011.

[7] Catherine Ross and Subhrajit Guhathakurta. Autonomous vehicles and energy impacts: A scenario analysis. *Energy Procedia*, 143:47–52, December 2017.

[8] Kevin Spieser, Kyle Treleaven, Rick Zhang, Emilio Frazzoli, Daniel Morton, and Marco Pavone. Toward a systematic approach to the design and evaluation of automated Mobility-on-Demand systems: A case study in singapore. In Gereon Meyer and Sven Beiker, editors, *Road Vehicle Automation*. Springer International Publishing: 229–245, Cham, 2014.

[9] Nikolaus Lang, Michael Rüßmann, Antonella Mei-Pochtler, Thomas Dauner, Satoshi Komiya, Xavier Mosquet, and Xanthi Doubara. Self-driving vehicles, robo-taxis, and the urban mobility revolution. *The Boston Consulting Group*, 7:2016, 2016.

[10] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE Int. J. Transp. Saf.*, 4(1):15–24, April 2016.

[11] Philip Koopman and Michael Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intell. Transp. Syst. Mag.*, 9(1):90–96, 2017.

[12] Alexandre Moreira Nascimento, Lucio Flavio Vismari, Caroline Bianca Santos Tancredi Molina, Paulo Sergio Cugnasca, João Batista Camargo, Jorge Rady de Almeida, Rafia Inam, Elena Fersman, Maria Valeria Marquezini, and Alberto Yukinobu Hata. A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety. *IEEE Trans. Intell. Transp. Syst.*, 21(12):4928–4946, December 2020.

[13] George Dimitrakopoulos and Panagiotis Demestichas. Intelligent transportation systems. *IEEE Veh. Technol. Mag.*, 5(1):77–84, March 2010.

[14] Yu Yamada, Abu Saleh Md Bakibillah, Kotaro Hashikura, Md Abdus Samad Kamal, and Kou Yamada. Autonomous vehicle overtaking: Modeling and an optimal trajectory generation scheme. *Sustainability*, 14(3):1807, 2022.

[15] Shouyang Wei, Yuan Zou, Xudong Zhang, Tao Zhang, and Xiaoliang Li. An integrated longitudinal and lateral vehicle following control system with radar and Vehicle-to-Vehicle communication. *IEEE Trans. Veh. Technol.*, 68(2):1116–1127, February 2019.

[16] Feng You, Ronghui Zhang, Guo Lie, Haiwei Wang, Huiyin Wen, and Jianmin Xu. Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system. *Expert Syst. Appl.*, 42(14):5932–5946, August 2015.

[17] Ansi/ul-4600 standard for evaluation of autonomous products. *Underwriters Laboratories*, April 2020.

[18] Uma D Ferrell and Alfred H Andy Anderegg. Applicability of ul 4600 to unmanned aircraft systems (uas) and urban air mobility (uam). In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC): 1–7*. IEEE, 2020.

[19] Rob Palin, David Ward, Ibrahim Habli, and Roger Rivett. Iso 26262 safety cases: Compliance and assurance. 2011.

[20] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. An analysis of iso 26262: Using machine learning safely in automotive software. *arXiv preprint arXiv:1709.02435*, 2017.

[21] ISO Iso. Pas 21448-road vehicles-safety of the intended functionality. *International Organization for Standardization*, 2019.

[22] SAE On road Automated Vehicles Standards Committee et al. Sae j3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems, 2018.

[23] Georg Macher, Christoph Schmittner, Omar Veledar, and Eugen Brenner. Iso/sae dis 21434 automotive cybersecurity standard-in a nutshell. In *International Conference on Computer Safety, Reliability, and Security: 123–135*. Springer, 2020.

[24] Christoph Schmittner, Gerhard Griessnig, and Zhendong Ma. Status of the development of iso/sae 21434. In *European Conference on Software Process Improvement: 504–513*. Springer, 2018.

[25] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *J. field robot.*, 37(3):362–386, April 2020.

[26] Pa Warrendale. SAE international releases updated visual chart for its "levels of driving automation" standard for self-driving vehicles.

[27] Andrew Best, Sahil Narang, Lucas Pasqualin, Daniel Barber, and Dinesh Manocha. Autonovi-sim: Autonomous vehicle simulation platform with weather, sensing, and traffic control. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[28] Chi Zhang, Yuehu Liu, Danchen Zhao, and Yuanqi Su. Roadview: A traffic scene simulator for autonomous vehicle simulation testing. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC): 1160-1165*, 2014.

[29] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, Eugene Agafonov, Tae Hyung Kim, Eric Sterner, Keunhae Ushiroda, Michael Reyes, Dmitry Zelenkovsky, and Seonman Kim. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC):1–6*, 2020.

[30] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*. PMLR, 2017.

[31] Ahmad E L Sallab, Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 29(19):70–76, 2017.

[32] Ami Woo, Baris Fidan, and William W Melek. Localization for autonomous driving. In *Handbook of Position Location:1051–1087*. Wiley, January 2019.

[33] Qiang Ren, Xian-Gao Quan, Yun-Long Wang, and Hui-Yun Wang. Isolation and identification of phloroglucinol derivatives from dryopteris crassirhizoma by HPLC-LTQ-Orbitrap mass spectrometry. *Chem. Nat. Compo.*, 52(6):1137–1140, November 2016.

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, Lecture notes in computer science:21–37. Springer International Publishing, Cham, 2016.

[35] Sajjad Mozaffari, Omar Y Al-Jarrah, Mehrdad Dianati, Paul Jennings, and Alexandros Mouzakitis. Deep Learning-Based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Trans. Intell. Transp. Syst.*, 23(1):33–47, January 2022.

[36] D Helbing and P Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics*, 51(5):4282–4286, May 1995.

[37] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR):961–971*, June 2016.

[38] Paulo Vinicius Koerich Borges, Nicola Conci, and Andrea Cavallaro. Video-Based human behavior understanding: A survey. *IEEE Trans. Circuits Syst. Video Technol.*, 23(11):1993–2008, November 2013.

[39] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition:2255–2264*, June 2018.

[40] Tsubasa Hirakawa, Takayoshi Yamashita, Toru Tamaki, and Hironobu Fujiyoshi. Survey on vision-based path prediction. In *Distributed, Ambient and Pervasive Interactions: Technologies and Contexts*, Lecture notes in computer science:48–64. Springer International Publishing, Cham, 2018.

[41] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: a survey. *Int. J. Rob. Res.*, 39(8):895–935, July 2020.

[42] Umit Ozguner, Christoph Stiller, and Keith Redmill. Systems for safety and autonomous behavior in cars: The DARPA grand challenge experience. *Proc. IEEE*, 95(2):397–412, February 2007.

[43] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the DARPA grand challenge. *J. field robot.*, 23(9):661–692, September 2006.

[44] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M N Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William “red” Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *J. field robot.*, 25(8):425–466, August 2008.

[45] Reinis Cimurs, Jaepyung Hwang, and Il Hong Suh. Bezier Curve-Based smoothing for path planner with curvature constraint. In *2017 First IEEE International Conference on Robotic Computing (IRC):241–248*, April 2017.

[46] M Garcia, A Viguria, and A Ollero. Dynamic Graph-Search algorithm for global path planning in presence of hazardous weather. *J. Intell. Rob. Syst.*, 69(1):285–295, January 2013.

[47] Ji-Wung Choi, Renwick Curry, and Gabriel Elkaim. Path planning based on bézier curve for autonomous ground vehicles. In *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008:158–166*, October 2008.

[48] Taehyun Shim, Ganesh Adireddy, and Hongliang Yuan. Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control. *Proc. Inst. Mech. Eng. Pt. D: J. Automobile Eng.*, 226(6):767–778, June 2012.

[49] Xiaohui Li, Zhenping Sun, Dongpu Cao, Daxue Liu, and Hangen He. Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mech. Syst. Signal Process.*, 87:118–137, March 2017.

[50] Xuemin Hu, Long Chen, Bo Tang, Dongpu Cao, and Haibo He. Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. *Mech. Syst. Signal Process.*, 100:482–500, February 2018.

[51] Unghui Lee, Sangyol Yoon, Hyunchul Shim, Pascal Vasseur, and Cédric Demonceaux. Local path planning in a complex environment for self-driving car. In *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent:445–450*, June 2014.

[52] Maxim Likhachev and Dave Ferguson. Planning long dynamically feasible maneuvers for autonomous vehicles. *Int. J. Rob. Res.*, 28(8):933–945, August 2009.

[53] Liang Ma, Jianru Xue, Kuniaki Kawabata, Jihua Zhu, Chao Ma, and Nanning Zheng. Efficient Sampling-Based motion planning for On-Road autonomous driving. *IEEE Trans. Intell. Transp. Syst.*, 16(4):1961–1976, August 2015.

[54] Tianyu Gu, Jarrod Snider, John M Dolan, and Jin-Woo Lee. Focused trajectory planning for autonomous on-road driving. In *2013 IEEE Intelligent Vehicles Symposium (IV):547–552*, June 2013.

[55] Matthew McNaughton, Chris Urmson, John M Dolan, and Jin-Woo Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *2011 IEEE International Conference on Robotics and Automation:4889–4895*, May 2011.

[56] Julius Ziegler and Christoph Stiller. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems:1879–1884*, October 2009.

[57] Rajat Kumar Panda and B B Choudhury. An effective path planning of mobile robot using genetic algorithm. In *2015 IEEE International Conference on Computational Intelligence & Communication Technology:287–291*, February 2015.

[58] He Bing, Liu Gang, Gao Jiang, Wang Hong, Nan Nan, and Li Yan. A route planning method based on improved artificial potential field algorithm. In *2011 IEEE 3rd International Conference on Communication Software and Networks:550–554*, May 2011.

[59] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for Self-Driving cars. April 2016.

[60] Changliu Liu, Wei Zhan, and Masayoshi Tomizuka. Speed profile planning in dynamic environments via temporal optimization. In *2017 IEEE Intelligent Vehicles Symposium (IV):154–159*, June 2017.

[61] Wenda Xu, Junqing Wei, John M Dolan, Huijing Zhao, and Hongbin Zha. A real-time motion planner with trajectory optimization for autonomous vehicles. In *2012 IEEE International Conference on Robotics and Automation:2061–2067*, May 2012.

[62] Yadollah Rasekhipour, Amir Khajepour, Shih-Ken Chen, and Bakhtiar Litkouhi. A potential Field-Based model predictive Path-Planning controller for autonomous road vehicles. *IEEE Trans. Intell. Transp. Syst.*, 18(5):1255–1267, May 2017.

[63] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for Self-Driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, March 2016.

[64] Andrew Bacha, Cheryl Bauman, Ruel Faruque, Michael Fleming, Chris Terwelp, Charles Reinholtz, Dennis Hong, Al Wicks, Thomas Alberi, David Anderson, Stephen Cacciola, Patrick Currier, Aaron Dalton, Jesse Farmer, Jesse Hurdus, Shawn Kimmel, Peter King, Andrew Taylor, David Van Covern, and Mike Webster. Odin: Team Victor-Tango's entry in the DARPA urban challenge. *J. field robot.*, 25(8):467–492, August 2008.

[65] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, Doug Johnston, Stefan Klumpp, Dirk Langer, Anthony Levandowski, Jesse Levinson, Julien Marcil, David Orenstein, Johannes Paefgen, Isaac Penny, Anna Petrovskaya, Mike Pflueger, Ganymed Stanek, David Stavens, Antone Vogt, and Sebastian Thrun. Junior: The stanford entry in the urban challenge. *J. field robot.*, 25(9):569–597, September 2008.

[66] Florian Damerow and Julian Eggert. Risk-Aversive behavior planning under multiple situations with uncertainty. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems:656–663*, September 2015.

[67] Peter Nilsson, Leo Laine, Niels van Duijkeren, and Bengt Jacobson. Automated highway lane changes of long vehicle combinations: A specific comparison between driver model based control and non-linear model predictive control. In *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA):1–8*, September 2015.

[68] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenét frame. In *2010 IEEE International Conference on Robotics and Automation:987–993*, May 2010.

[69] Shilp Dixit, Saber Fallah, Umberto Montanaro, Mehrdad Dianati, Alan Stevens, Francis Mccullough, and Alexandros Mouzakitis. Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects. *Annual Reviews in Control*, 45:76–86, 2018.

[70] Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. *IEEE Trans. Veh. Technol.*, 66(2):952–964, February 2017.

[71] James S Albus. The NIST real-time control system (RCS): an approach to intelligent systems research. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):157–174, 1997.

[72] Nicola Muscettola Gregory, Gregory A Dorais, Chuck Fry, Richard Levinson, and Christian Plaunt. IDEA: Planning at the core of autonomous reactive agents. In *in Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space*, 2002.

[73] Kurt Konolige, Karen Myers, Enrique Ruspini, and Alessandro Saffiotti. The saphira architecture: a design for autonomy. *J. Exp. Theor. Artif. Intell.*, 9(2-3):215–235, April 1997.

[74] R Volpe, I Nesnas, T Estlin, D Mutz, R Petras, and H Das. The CLARAty architecture for robotic autonomy. In *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542):1/121–1/132 vol.1*, volume 1, March 2001.

[75] Alexandru Constantin Serban, Erik Poll, and Joost Visser. A standard driven software architecture for fully autonomous vehicles. In *2018 IEEE International Conference on Software Architecture Companion (ICSA-C):120–127*, April 2018.

[76] Matthieu Carré, Ernesto Exposito, and Javier Ibanez-Guzman. Challenges for the Self-Safety in autonomous vehicles. In *2018 13th Annual Conference on System of Systems Engineering (SoSE):181–188*, June 2018.

[77] D Harrison McKnight and Norman L Chervany. Trust and distrust definitions: One bite at a time. In *Trust in Cyber-societies*, Lecture notes in computer science:27–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[78] John D Lee and Katrina A See. Trust in automation: designing for appropriate reliance. *Hum. Factors*, 46(1):50–80, 2004.

[79] Milecia Matthews, Girish Chowdhary, and Emily Kieson. Intent communication between autonomous vehicles and pedestrians. August 2017.

[80] Marc-Philipp Böckle, Anna Pernestål Brenden, Maria Klingegård, Azra Habibovic, and Martijn Bout. SAV2P: Exploring the impact of an interface for shared automated vehicles on pedestrians' experience. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct*, AutomotiveUI '17:136–140, New York, NY, USA, September 2017. Association for Computing Machinery.

[81] Baohua Sun, Hao Sha, Manouchehr Rafie, and Lin Yang. CDVA/VCM: Language for intelligent and autonomous vehicles. In *2020 IEEE International Conference on Image Processing (ICIP):3104–3108*, October 2020.

[82] Victor Malmsten Lundgren, Azra Habibovic, Jonas Andersson, Tobias Lagström, Maria Nilsson, Anna Sirkka, Johan Fagerlönn, Rikard Fredriksson, Claes Edgren, Stas Krupenia, and Dennis Saluäär. Will there be new communication needs when introducing automated vehicles to the urban context? In *Advances in Intelligent Systems and Computing*, Advances in intelligent systems and computing:485–497. Springer International Publishing, Cham, 2017.

[83] Franz Keferböck and Andreas Riener. Strategies for negotiation between autonomous vehicles and pedestrians. In *Mensch und Computer 2015 – Workshopband:525–532*. De Gruyter, September 2015.

[84] Jesse Fox, Dylan Arena, and Jeremy N Bailenson. Virtual reality. *Journal of Media Psychology*, 21(3):95–113, January 2009.

[85] Chia-Ming Chang, Koki Toda, Daisuke Sakamoto, and Takeo Igarashi. Eyes on a car: an interface design for communication between an autonomous car and a pedestrian. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '17:65–73, New York, NY, USA, September 2017. Association for Computing Machinery.

[86] Koen de Clercq, Andre Dietrich, Juan Pablo Núñez Velasco, Joost de Winter, and Riender Happee. External Human-Machine interfaces on automated vehicles: Effects on pedestrian crossing decisions. *Hum. Factors*, 61(8):1353–1370, December 2019.

[87] Karthik Mahadevan, Sowmya Somanath, and Ehud Sharlin. Communicating awareness and intent in autonomous vehicle-pedestrian interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems:1–12*, 2018.

[88] Michael Clamann, Miles Aubert, and Mary L Cummings. Evaluation of vehicle-to-pedestrian communication displays for autonomous vehicles. Technical Report 17-02119, 2017.

[89] Yeti Li, Murat Dikmen, Thana G Hussein, Yahui Wang, and Catherine Burns. To cross or not to cross: Urgency-based external warning displays on autonomous vehicles to improve pedestrian crossing safety. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications:188–197*, 2018.

[90] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Agreeing to cross: How drivers and pedestrians communicate. In *2017 IEEE Intelligent Vehicles Symposium (IV):264–269*, June 2017.

[91] Friederike Schneemann and Irene Gohl. Analyzing driver-pedestrian interaction at crosswalks: A contribution to autonomous driving in urban environments. In *2016 IEEE Intelligent Vehicles Symposium (IV):38–43*, June 2016.

[92] Matus Sucha, Daniel Dostal, and Ralf Risser. Pedestrian-driver communication and decision strategies at marked crossings. *Accid. Anal. Prev.*, 102:41–50, May 2017.

[93] M Kyriakidis, R Happee, and J C F de Winter. Public opinion on automated driving: Results of an international questionnaire among 5000 respondents. *Transp. Res. Part F Traffic Psychol. Behav.*, 32:127–140, July 2015.

[94] Amin Rezaei and Brian Caulfield. Examining public acceptance of autonomous mobility. *Travel Behaviour and Society*, 21:235–246, October 2020.

[95] Rahul Razdan and System Centering Solutions. Unsettled topics concerning autonomous public transportation systems. SAE International, October 2020.

[96] Rahul Razdan and Florida Polytechnic Univ. Unsettled technology areas in autonomous vehicle test and validation. SAE International, 2019.

[97] Joachim Taiber and International Transportation Innovation Center. Unsettled impacts of integrating automated electric vehicles into a mobility-as-a-service ecosystem and effects on traditional transportation and ownership. Technical report, December 2019.

[98] Mohsen Malayjerdi, Vladimir Kuts, Raivo Sell, Tauno Otto, and Barış Cem Baykara. Virtual simulations environment development for autonomous vehicles interaction. *Volume 2B: Advanced Manufacturing*, 2020.

[99] Vladimir Kuts, Natalia Cherezova, Martins Sarkans, and Tauno Otto. Digital twin: industrial robot kinematic model integration to the virtual reality environment. *Journal of Machine Engineering*, 20(2):53–64, 2020.

[100] Jiyo Palatti. Autonomous overtaking maneuver under complex driving conditions. Master's thesis, Aalto University. School of Electrical Engineering, 2021.

[101] Azim Eskandarian. *Handbook of intelligent vehicles*, volume (2):1–13. Springer, 2012.

[102] T J Gordon and M Lidberg. Automated driving and autonomous functions on road vehicles. *Vehicle System Dynamics*, 53(7):958–994, 2015.

[103] Eleni I Vlahogianni. Modeling duration of overtaking in two lane highways. *Transportation Research Part F: Traffic Psychology and Behaviour*, 20:135–146, 2013.

[104] Sara Moll, Griselda López, Alexander Rasch, Marco Dozza, and Alfredo García. Modelling duration of car-bicycles overtaking manoeuvres on two-lane rural roads using naturalistic data. *Accid. Anal. Prev.*, 160:106317, September 2021.

[105] Shilp Dixit. *Autonomous high-speed overtaking in structured environments*. PhD thesis, University of Surrey, 2020.

[106] Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer, and Lydie Nouvelière. Maneuver-Based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Trans. Intell. Transp. Syst.*, 11(3):589–606, September 2010.

[107] Shohei Kitazawa and Tetsuya Kaneko. Control target algorithm for direction control of autonomous vehicles in consideration of mutual accordance in mixed traffic conditions. In *Advanced Vehicle Control AVEC'16:151–156*. Crc Press, CRC Press/Balkema, P.O. Box 11320, 2301 EH Leiden, The Netherlands, e-mail: Pub.NL@taylorandfrancis.com, www.crcpress.com - www.taylorandfrancis.com, December 2016.

[108] Liang Ma, Jianru Xue, Kuniaki Kawabata, Jihua Zhu, Chao Ma, and Nanning Zheng. A fast RRT algorithm for motion planning of autonomous road vehicles. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC):1033–1038*, October 2014.

[109] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transp. Res. Part C: Emerg. Technol.*, 60:416–442, November 2015.

[110] Usman Ghumman, Faraz Kunwar, and Beno Benhabib. Guidance-Based On-Line motion planning for autonomous highway overtaking. *International Journal on Smart Sensing and Intelligent Systems*, 1(2):549–571, January 2008.

[111] Plamen Petrov and Fawzi Nashashibi. Modeling and nonlinear adaptive control for autonomous vehicle overtaking. *IEEE Trans. Intell. Transp. Syst.*, 15(4):1643–1656, August 2014.

[112] Andrew Gray, Yiqi Gao, Theresa Lin, J Karl Hedrick, H Eric Tseng, and Francesco Borrelli. Predictive control for agile semi-autonomous ground vehicles using motion primitives. In *2012 American Control Conference (ACC):4239–4244*, June 2012.

[113] Ghumman Usman and Faraz Kunwar. Autonomous vehicle overtaking- an online solution. In *2009 IEEE International Conference on Automation and Logistics:596–601*, August 2009.

[114] Tzila Shamir. How should an autonomous vehicle overtake a slower moving vehicle: Design and analysis of an optimal trajectory. *IEEE Transactions on Automatic Control:607–610*, 49(4):607–610, 2004.

[115] Keonyup Chu, Minchae Lee, and Myoungho Sunwoo. Local path planning for Off-Road autonomous driving with avoidance of static obstacles. *IEEE Trans. Intell. Transp. Syst.*, 13(4):1599–1616, December 2012.

[116] Yonglin Gao, Wanglin Jiang, Chaohua Dong, Chunmei Li, Xuejun Fu, Li Min, Jingwei Tian, Haizhu Jin, and Jingyu Shen. Anti-inflammatory effects of sophocarpine in LPS-induced RAW 264.7 cells via NF-κB and MAPKs signaling pathways. *Toxicol. In Vitro*, 26(1):1–6, February 2012.

[117] Ashwin Carvalho, Yiqi Gao, Andrew Gray, H Eric Tseng, and Francesco Borrelli. Predictive control of an autonomous ground vehicle using an iterative linearization approach. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013):2335–2340*, October 2013.

[118] Beomjun Kim, Dongwook Kim, Sungyoul Park, Yonghwan Jung, and Kyongsu Yi. Automated complex urban driving based on enhanced environment representation with GPS/map, radar, lidar and vision. *IFAC-PapersOnLine*, 49(11):190–195, January 2016.

[119] Nikolce Murgovski and Jonas Sjöberg. Predictive cruise control with autonomous overtaking. In *2015 54th IEEE Conference on Decision and Control (CDC):644–649*, December 2015.

[120] Yiqi Gao, Andrew Gray, H Eric Tseng, and Francesco Borrelli. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Veh. Syst. Dyn.*, 52(6):802–823, June 2014.

[121] Georg Schildbach and Francesco Borrelli. Scenario model predictive control for lane change assistance on highways. In *2015 IEEE Intelligent Vehicles Symposium (IV):611–616*, June 2015.

[122] Michael T Wolf and Joel W Burdick. Artificial potential functions for highway driving with collision avoidance. In *2008 IEEE International Conference on Robotics and Automation:3731–3736*, May 2008.

[123] Tim Brüdigam, Michael Olbrich, Marion Leibold, and Dirk Wollherr. Combining stochastic and scenario model predictive control to handle target vehicle uncertainty in an autonomous driving highway scenario. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC):1317–1324*, November 2018.

[124] Gianluca Cesari, Georg Schildbach, Ashwin Carvalho, and Francesco Borrelli. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intell. Transp. Syst. Mag:23–35*, 9(3), 2017.

[125] Georg Schildbach, Lorenzo Fagiano, Christoph Frei, and Manfred Morari. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12):3009–3018, December 2014.

[126] Noor Hafizah Amer, Hairi Zamzuri, Khisbullah Hudha, and Zulkiffli Abdul Kadir. Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges. *J. Intell. Rob. Syst.*, 86(2):225–254, May 2017.

[127] Astrid Rupp and Michael Stolz. Survey on control schemes for automated driving on highways. In Daniel Watzenig and Martin Horn, editors, *Automated Driving: Safer and More Efficient Future Driving:43–69*. Springer International Publishing, Cham, 2017.

[128] Aldo Sorniotti, Phil Barber, and Stefano De Pinto. Path tracking for automated driving: A tutorial on control system formulations and ongoing research. In Daniel Watzenig and Martin Horn, editors, *Automated Driving: Safer and More Efficient Future Driving*. Springer International Publishing:71–140, Cham, 2017.

[129] Gilles Tagne, Reine Talj, and Ali Charara. Design and comparison of robust nonlinear controllers for the lateral dynamics of intelligent vehicles. *IEEE Trans. Intell. Transp. Syst.*, 17(3):796–809, March 2016.

[130] Yonghwan Jeong and Seongjin Yim. Model predictive Control-Based integrated path tracking and velocity control for autonomous vehicle with Four-Wheel independent steering and driving. *Electronics*, 10(22):2812, November 2021.

[131] Thomas Besselmann and Manfred Morari. Autonomous vehicle steering using explicit LPV-MPC. *2009 European Control Conference (ECC)*, 2009.

[132] S Chaib, M S Netto, and S Mammar. H/sub /spl infin//, adaptive, PID and fuzzy control: a comparison of controllers for vehicle lane keeping. In *IEEE Intelligent Vehicles Symposium:139–144*, June 2004.

[133] Daniel Watzenig and Bernhard Brandstätter. *Comprehensive Energy Management – Eco Routing & Velocity Profiles*. Springer International Publishing, June 2017.

[134] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS):287–296*, April 2018.

[135] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[136] Anton Rassõlkin, Levon Gevorkov, Toomas Vaimann, Ants Kallaste, and Raivo Sell. Calculation of the traction effort of ISEAUTO self-driving vehicle. In *2018 25th International Workshop on Electric Drives: Optimization in Control of Electric Drives (IWED):1–5*, January 2018.

[137] Raivo Sell, Mairo Leier, Anton Rassolkin, and Juhan-Peep Ernits. Self-driving car ISEAUTO for research and education. *2018 19th International Conference on Research and Education in Mechatronics (REM)*, 2018.

[138] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Miss. Q.*, 13(3):319–340, 1989.

[139] Raivo Sell and Tauno Otto. Remotely controlled multi robot environment. In *2008 19th EAEEIE Annual Conference:20–25*, June 2008.

[140] Daniel J Fremont, Edward Kim, Yash Vardhan Pant, Sanjit A Seshia, Atul Acharya, Xantha Bruso, Paul Wells, Steve Lemke, Qiang Lu, and Shalin Mehta. Formal Scenario-Based testing of autonomous vehicles: From simulation to the real world. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC):1–8*, September 2020.

[141] Prabhjot Kaur, Samira Taghavi, Zhaofeng Tian, and Weisong Shi. A survey on simulators for testing Self-Driving cars. In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD):62–70*, April 2021.

[142] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, Eugene Agafonov, Tae Hyung Kim, Eric Sterner, Keunhae Ushiroda, Michael Reyes, Dmitry Zelenkovsky, and Seonman Kim. LGSVL simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC): 1–6*, September 2020.

Acknowledgements

This research was supported by the European Union's Horizon 2020 Research and Innovation Programme, under grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No 2014-2020.4.01.20-0289.

I would like to express my sincere thanks to my friends, colleagues, and especially my family (my wife, and children), for their support in helping me to achieve this major milestone in my life. My sincere gratitude goes out to Professor Raivo Sell, who always kept me motivated and enthusiastic about my research in the field of autonomous vehicles. Having them as my inspiration has kept me going, and I would like to thank them for inspiring me to keep going and one day achieve the level I would like to reach.

Abstract
Advanced autonomous vehicle's functions for safety improvements in urban mobility context

Autonomous Vehicles (AV) have the potential to significantly enhance mobility through better access to services and increase safety. Furthermore, the traffic efficiency can be considerably improved. For an AV to become a reality, numerous challenges need to be solved. A system for Autonomous Driving (AD) must be capable of operating a car without human intervention. To drive the vehicle safely and efficiently, it must observe its surroundings, make the right decisions, and manoeuvre as directed by traffic rules. The highest level of automation, i.e., Level 4 and Level 5, presumes that the AV can drive as well as an experienced human driver, including handling complex manoeuvres like overtaking and managing communication with pedestrians and passengers.

The dissertation presents a summary of the six articles published by the author in 2018–2022 on modelling, simulation of control algorithms of AVs focusing on human-vehicle interactions, simulations, and advanced path planning algorithms for complex manoeuvres. The main outcomes of the dissertation are:

- Research on the development of a modular smart control system architecture for unmanned ground vehicles (UGV) using hierarchical hardware and software structures, such as three-level hardware and two-level software architecture that can be adapted to the target platform and tested on actual UGVs and AV.
- The development of intelligent functions on an autonomous vehicle for interacting with pedestrians and ensuring their safety by light patterns designed for experimentation.
- The development of advanced techniques in the safety validation area by using end-to-end simulation technologies for the safety evaluation of an AV through a virtual environment by using geospatial data and a 3D model of the AV.
- The validation and evaluation of safety through the development of digital twins. Demonstration of the potential of advanced control methods for AVs such as sigmoid functions, optimization, and model-based LQR control for overtaking manoeuvres.

The primary objective of the current research was to address the challenges of bringing an AV to the road without compromising safety. The functions and algorithms developed under this research were all validated, first in simulation environments and then on the field by using actual vehicles in real traffic situations. Valuable data and experimental results were collected and based on the analysis a new solution to improve the safety of AVs was proposed. All initial goals were achieved, and research questions were addressed. An autonomous vehicle control system was developed using a modular approach. A novel human-vehicle interaction using autonomous vehicle lights was developed and experimented with different target groups. A novel digital twin design methodology was presented, and most of all, new overtaking methods for AVs were developed and implemented on the TalTech iseAuto.

Kokkuvõte
Täiustatud autonoomsete sõidukite funktsioonid ohutuse parandamiseks linnaliikluse kontekstis

Autonoomsed ehk isejuhtivad sõidukid võimaldavad liikuvusteenustele oluliselt paremat juurdepääsu erinevatele sihtrühmadele. Selle eelduseks on aga töökindlad ja turvalised autonoomse juhtimise funktsioonid. Selleks, et isejuhtivad sõidukid saaks reaalsuseks, tuleb lahendada arvukalt erinevaid väljakutseid. Autonoomse juhtimise süsteem peab suutma autot ohutult ja efektiivselt juhtida ilma inimese sekkumiseta. Sõiduki ohutuks ja tõhusaks juhtimiseks peab see olema suuteline korrektselt tajuma enda ümbritsevat keskkonda, tegema õigeid juhtimisotsuseid ja sooritama manöövreid vastavalt liikluseeskirjadele ja liiklussituatsioonile. Automatiseerimise kõrgeimal tasemel, st 4. ja 5. tasemel, eeldatakse, et isejuhtiv sõiduk on võimeline juhtima sama hästi kui kogenud inimjuht, sealhulgas sooritama keerulisi manöövreid, nagu möödasõidud ja manöövrite vajadusel katkestamine. Lisaks on oluline masin-inimene suhtlusliides ehk isejuhtiv sõiduk peab olema vajadusel võimeline vahetama infot jalakäijate ja reisijatega.

Doktoritöös esitatakse kokkuvõte autori kuuest peamisest publikatsioonist perioodil 2018–2022 modelleerimise, autonoomsete sõidukite juhtimisalgoritmide simulatsiooni, inimene-sõiduki omavahelisele suhtluse, simulatsioonide ja keerukate manöövrite jaoks täiustatud teeplaneerimise algoritmidest. Peamised tulemused on järgmised:

- Teadusuuringud mehitamata maismaasõidukite (UGV) modulaarse juhtimis-süsteemi arhitektuuri arendamiseks, kasutades hierarhilisi riist- ja tarkvarastruktuure. Pakutakse välja kolmetasandiline riist- ja kahetasandiline tarkvaraarhitektuur, mida on võimalik kohandada erinevatele robotsõidukitele.
- Autonoomse sõiduki intelligentsete funktsioonide arendamine jalakäijatega suhtlemiseks valgusmustrite abil.
- Täiustatud tehnikate väljatöötamine ohutuse valideerimiseks, kasutades AV minibussi ohutuse hindamiseks virtuaalses keskkonnas.
- Turvalisuse tagamine ja hindamine digitaalsete kaksikute arendamise kaudu.
- Autonoomsete sõidukite täiustatud juhtimismeetodite (nt sigmoidfunktsioonid, optimeerimine ja mudelipõhine LQR-juhtimine möödasõidumanöövrite jaoks).

Käesoleva uuringu esmane eesmärk oli tegeleda probleemidega, mis on seotud autonoomse sõiduki teedele toomisega fookusega ohutusele. Kõik selle uurimistöö raames välja töötatud funktsioonid ja algoritmid valideeriti katsete käigus, esmalt simulatsioonikeskkondades ja seejärel tegelike sõidukitega reaalsetes liiklusolukordades. Koguti väärtuslikke andmeid ja katsetulemusi ning analüüsi põhjal pakuti välja uudne lahendus ohutuse parandamiseks. Kõik esialgsed eesmärgid saavutati vastavalt plaanile. Sõiduki autonoomne juhtimissüsteem töötati välja kasutades modulaarset lähenemist. Töötati välja uudne inimese ja sõiduki suhtluskeel visuaalsete sümbolite abil tulede kaudu ja seda katsetati erinevate sihtrühmadega. Esitleti uudset digitaalse kaksiku loomise metoodikat ning enneкõike töötati välja ja rakendati TalTech iseAuto minibussil uued autonoomsete sõidukite möödasõidumeetodid.

Appendix 1

I

Raivo Sell, Eero Väljaots, Tengiz Pataraia, and Ehsan Malayjerdi. Modular smart control system architecture for the mobile robot platform. Proceedings of the Estonian Academy of Sciences, 68(4):395–400, 2019

Proceedings of the Estonian Academy of Sciences,
2019, **68**, 4, 395–400
https://doi.org/10.3176/proc.2019.4.08
Available online at www.eap.ee/proceedings

MANUFACTURING
ENGINEERING

Modular smart control system architecture for the mobile robot platform

Raivo Sell*, Eero Väljaots, Tengiz Pataraia, and Ehsan Malayjerdi

Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia

Received 12 April 2019, accepted 16 May 2019, available online 19 November 2019

Abstract. The paper is focusing on the research of current technologies of multi-purpose mobile robots, including implementation of modular layout, open software frameworks, self-organization with plug and play (PnP) capability and adaptive sensor fusion. The hardware layout is studied in detail and modular architecture is proposed as an optimal solution of module interconnection. Software modular architecture is developed in the similar principle, taking plug and play connectivity into account. The framework of the software consists of implementing middleware for the software modularization and three-level hierarchical structure. Practical implementation platforms are introduced and future developments discussed.

Key words: mobile robot, unmanned ground vehicle, system architecture.

1. INTRODUCTION

Unmanned ground vehicles (UGVs) used to be operated by the remote control in performing hazardous tasks over the distance. Nowadays, UGVs, are developed for various purposes, e.g., for research and industrial use, and these are sophisticated vehicles having semi-autonomous or full autonomous modes. For autonomous navigation and safe driving more sophisticated software and hardware is required, including high performance sensors like light detection and ranging (LiDAR); and cameras. Autonomous driving requires different sub-tasks to be solved, e.g. localization, base-navigation, local and global planning, obstacle avoidance, etc. [1]. Although these tasks are similar for most of the mobile robots, they still are very dependent on the selected sensors and robot hardware specifics. To overcome this issue, modular structure of the system is the most obvious choice, especially if the similar functionality has to be applied for different mobile robots. The modular concept of the mobile robots has been studied in many researches and specific solutions have been offered, e.g. in [2,3]. This research is based on the knowledge and expertise acquired from the first Estonian self-driving car – ISEAUTO [4,5], which was put on the road after less than a year-long development. The success of the project motivates to apply similar software and hardware concept also to smaller off-road vehicles. The conceptual solution is an open source modular smart control system for mid-size off-road mobile robotic platforms. Two different robots were used for the experiments and implementation of the concept. The manuscript is organized so that three following chapters introduce two mobile robots as a base platform for the concept implementation. Chapters 5 to 7 describe the concept of the modular architecture and the implementation.

2. IMPLEMENTATION PLATFORMS

The unmanned robotic platforms where the concept of modular architecture is applied, are universal mobile

* Corresponding author, raivo.sell@taltech.ee

robotic platform UKU – developed by the students of Tallinn University of Technology (TalTech); and Agronaut – a robot for the agriculture, developed by the Estonian company Hecada. Both vehicles are in the same size but use different concepts of steering and power system.

3. UNIVERSAL MOBILE ROBOT UKU

The all-terrain mobile robot UKU, shown in Fig. 1, is powered by the electric motor and Li-Io batteries. The robot is with a rear-wheel drive without differential and mechanical transmission. It has off road tires suitable for climbing over the obstacles or for example, plowing snow autonomously on the parking lot in winter. The robot has a special self-contained measurement system to measure the efficiency, similar to [6] and dynamics of energy consumption. Robot weights 260 kg and has nominal speed of 4 m/s. The main electric motor is a permanent magnet DC motor producing 4 kW. Basic navigation sensors are SICK 2D laser scanners (measuring range up to 80 m), ultrasonic rangers on both sides and long-range 360-degree LiDAR on the top of the vehicle. The more detailed architecture of the robot is described in [7].

4. AGRICULTURAL MOBILE ROBOT AGRONAUT

The Agronaut [8], shown in Fig. 2, is a universal mobile robotic platform, which purpose is practical testing of unmanned technologies and navigation in agricultural conditions. Its physical layout is symmetric and modular, consisting of identical modules that are connected to each other through hydraulic steering linkage. One module accommodates the 15-kW power plant that powers the hydraulic system. The other module is free for transporting necessary task-specific equipment. It is also possible to connect the third and the fourth module and actuators using the same physical interface on both

Fig. 1. Universal mobile robot UKU.

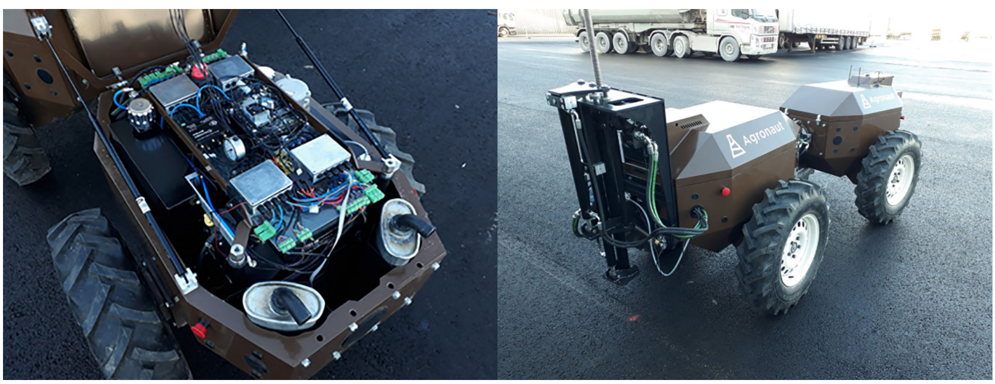

Fig. 2. Agronaut UGV platform, modular electronic units on top of the hydraulic system.

ends of the body control module. Agronaut is all-wheel drive robot and has robust design to suit for the agricultural field of use. Therefore, unlike robot UKU, all actuators outside the body are hydraulically powered and hydraulic lines are routed from one unit to another. As the vehicle can be assembled with task-specific modules, the control system also has to be modular and easily configurable for the required task. Each hydraulic actuator has its own electronic control, implementing PID-regulators for controlling hardware, connected with central computer through controller area network (CAN). As the platform weighs 470 kg, it suits perfectly for automating repetitive simple tasks usually carried out by humans, e.g., automated soil sampling of the cultivated land.

5. CONCEPT OF MODULAR ARCHITECTURE

Hardware and software architecture were developed based on technical requirements which were set initially according to the level of functionality and robustness of the system. Key priority of software and hardware architecture was to create a safe, easily reconfigurable and scalable system where each module/microcontroller has a task to control each locomotion unit (steering, driving motors, brakes) and different sensors across the platform. The whole control system is based on two communication protocols – Controller Area Network (CAN) bus and Universal Datagram Protocol (UDP). These protocols were chosen based on several key priorities: speed, reliability, and robustness. Intermodule communication was developed based on the CAN bus. Choice of CAN bus was made due to its high-transmission reliability, real-time capabilities, and robustness. Communication between the master controller and the computer was done through UDP messages due to its speed, reliability, and efficiency. Figure 3 shows SysML block definition diagram (bdd) and internal block diagrams (ibd) for general modular architecture. These diagrams describe how messages are delivered from the main computing unit to the lower level controllers and main parameter values of the data flow.

6. IMPLEMENTATIONS OF MODULAR ARCHITECTURE

The concept of modular architecture is applied to the base platforms described earlier. The hardware architecture implemented in mobile robot UKU has two-level software control, and three-level hardware control architecture. Figure 4 shows implementation of the hardware architecture in specific UGV platform. It has high-level sensors dealing with navigation and obstacle avoidance. High-level sensors are directly connected to the main computing unit, which runs the Robotic Operating System (ROS) middleware and open source components for its main tasks. ROS-based system control algorithms produce driving commands to low-level controllers through the master controller. Master controllers prioritize and translate messages between high-level and low-level controllers. Low-level controllers are dealing with direct control of motor drivers, consisting of proportional-integral-derivative (PID) and ground safety algorithms. There is a separate safety controller and it is independent from both high-level and low-level controllers. It monitors output signals as well as CAN network messages and can stop the vehicle in case anomalies occur. The system has also remote link over the 4G/5G mobile network to provide online data stream between server and operator.

Control software for each control unit of the mobile robot was designed in a similar manner to make it simple, easily understood and configurable in the future for different systems, e.g. front and rear locomotion units of the robot have the same PID regulators with different data inputs and controller coefficients.

Let’s take an example of a steering controller where desired wheel angle (setpoint) is sent from the ROS

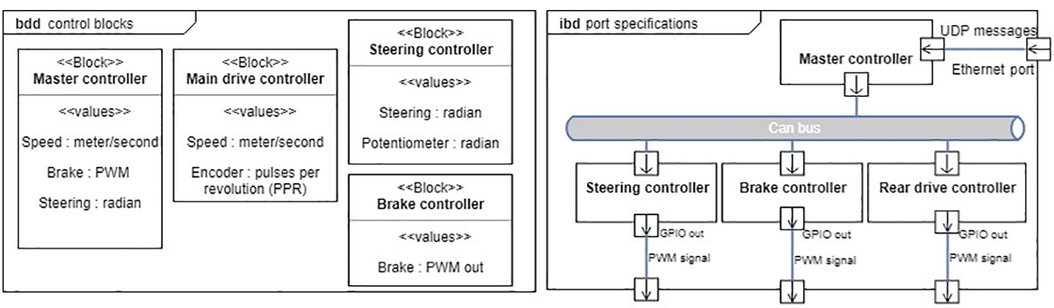

Fig. 3. General modular architecture of UGV.

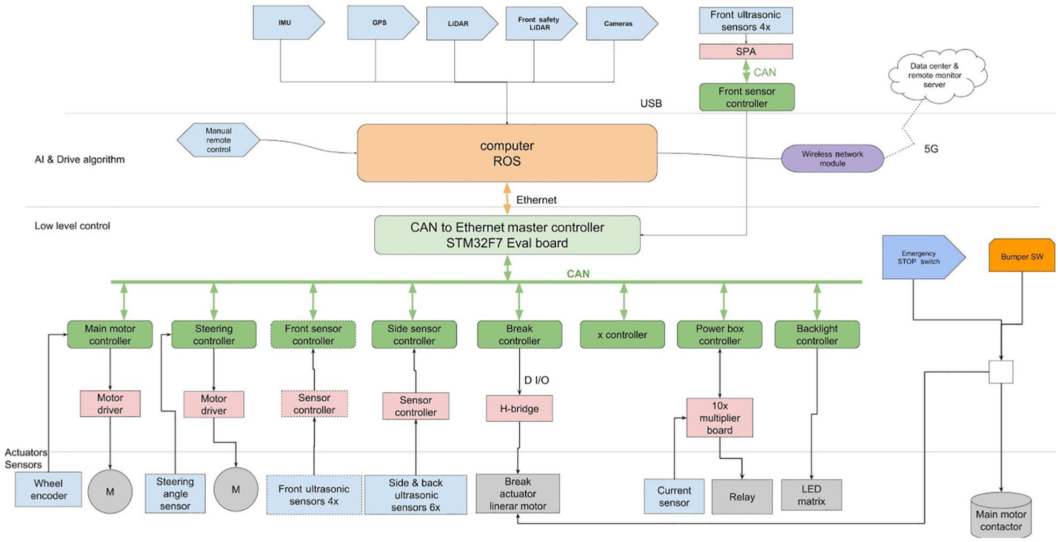

Fig. 4. Hardware architecture implemented on the UGV UKU.

computer with the CAN bus protocol and the controller's task is to calculate the most optimal motor speed to reach the desired steering angle.

Control object, in this case, is H-bridge type motor driver, which is changing the polarity of the voltage based on the pulse width modulation (PWM) signals, which are generated from the controller. Feedback device for the steering motor is a simple analog sensor (potentiometer) due to its simplicity of integration and accuracy. Control process of the PID is shown in Fig. 5.

PID controller of both actuators was tuned and validated separately by step tests. This process clearly shows the dynamical characteristic of each actuator. Testing and fine-tuning of PID controller defines the overall performance of the mobile robot in the future and is therefore crucial. Figure 6 shows the most optimal test result for front steering. Setpoint of the experiment was steering angle input from the computer varying from –1.0 to 1.0 radians, and potentiometer value as a controller feedback. This test case where the practical method of [9] was followed shows step response for controller with an angle input from 0.9 rad to –0.9 rad, which is one full rotation of the steering axle of the UGV. Proportional derivative (PD) action of the PID controller was enough to reach the optimal results for both steering and rear-wheel drive. The advantage of using only PD characteristics is rapid output and short time required to return process value to setpoint. PD formula can be seen in Eq. (1):

$$u = k_p + e + k_d \frac{d_e}{d_t}, \quad (1)$$

where u – static characteristics, k_p – proportional gain, k_d – derivative gain, e – error, d_e – change in error; and d_t – change in time. After tuning the PD parameters for steering motor we got the most optimal result using coefficient values: $k_p = 0.1$ and $k_d = 0.9$. The graph shows that the steady state is reached after 8 seconds. This means that the robot will do one full turn of the

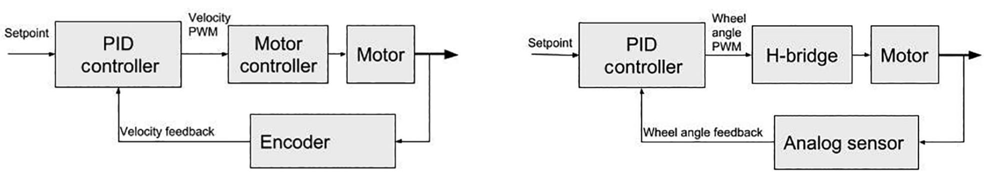

Fig. 5. Steering and main motor control diagram.

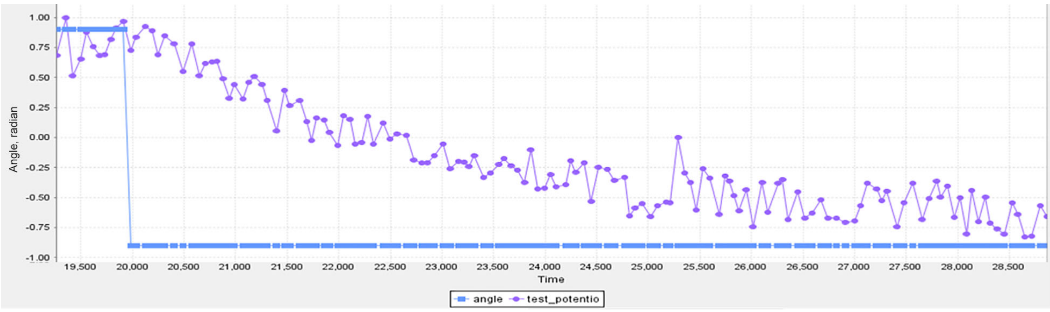

Fig. 6. PD tuning test results.

steering axle in 8.5 seconds, which is a satisfactory result for the research if we take into account the fact that the experiment was done on a standing vehicle.

7. SOFTWARE ARCHITECTURE

The high-level software architecture of the system, shown in Fig. 7, is based on the Robot Operating System (ROS). The reasons behind this decision were open-source drivers for multiple sensors and simplicity of integration of third-party software like Autoware, Yolo and multiple device drivers. According to the current architecture, the mobile robot takes several inputs from sensors. The global positioning system (GPS) is used for localization and path following, which is defined by a human. 2D (LiDAR), and camera inputs are used for obstacles' detection and safety. Output commands are steering angle, brake and linear velocity, which are sent to the low-level controllers over the UDP messages. Main software for computation of current architecture is Autoware [10], which is an open source library for self-driving cars and thus has many advanced software capabilities like lane following, obstacle avoidance, traffic light detection, lane detection etc.

The ROS platform itself is based on high modularity and scalability due to its master/slave architecture. ROS communication protocol is based on the publish-subscribe method and therefore it allows us to use external libraries and run separate individual nodes that will easily interact with each other even on multiple platforms. The ROS is a middleware and operates well on multiple cross platforms however software architecture described in this section does not use ROS on lower level controllers due to its lack of real-time capabilities. To merge ROS and lower-level controllers, software bridge was built, which converts custom ROS messages to the UDP messages. Modularity of this particular high-level software architecture is mainly the result of key principles of ROS and its approach of implementing multiple software libraries as building blocks of the primary product.

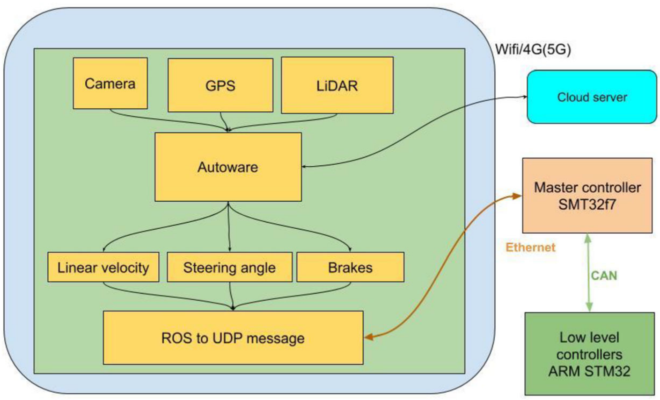

Fig. 7. Software architecture and message flow.

8. CONCLUSIONS

A modular architecture was proposed in this study to achieve flexibility and fast implementation process for off-road unmanned ground vehicles (UGVs). The architecture proposes hierarchical hardware and software structure and in particular, three-level hardware and two-level software architecture, which can be adapted according to the target platform specifics. System design and modular concept was implemented by taking into account an early stage mechatronic methodology proposed by [11]. The implementation example relies on the off-road universal robot platform UKU described in the first part of the paper. In addition, another similar robot platform is considered to reach comparable results of implementation. Both UGVs have the same size and similar purpose but they are different in their power and locomotion concept. The proposed concept for the hardware controller was evaluated in more detail using a test drive of UKU. Results showed that the proposed architecture guarantees the main driving requirements achieving stable navigation in dynamic and unknown environments, having fast and flexible implementation at the same time. The experiment of the implementation of proposed modular control system for UKU confirms that the proposed modular architecture can be easily implemented for similar unmanned robots, e.g. Agronaut. The future works include advancements in modular architecture, in particular, the high-level control algorithms including AI-based mission planning and navigation.

ACKNOWLEDGEMENTS

The research was supported by the European Commission through the H2020 project Finest Twins (grant No. 856602). The publication costs of this article were covered by the Estonian Academy of Sciences.

REFERENCES

1. Saska, M., et al. Navigation, localization and stabilization of formations of unmanned aerial and ground vehicles. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, May 28–31, 2013*. IEEE, 2013. https://ieeexplore.ieee.org/document/6564767
2. Ahmadzadeh, H., Masehian, E., and Asadpour, M. Modular Robotic Systems: Characteristics and Applications. *J. Intell. Rob. Syst.*, 2016, **81**(3–4), 317–357.
3. Wang, Y., et al. Building Unmanned Plant Factory with Modular Robotic Manipulation and Logistics Systems. In *Recent Developments in Intelligent Computing, Communication and Devices. Advances in Intelligent Systems and Computing, Vol. 752* (Patnaik, S. and Jain, V., eds). Springer, Singapore, 2019, 11–19.
4. Rassõlkin, A., Sell, R., and Leier, M. Development Case Study of the First Estonian Self-Driving Car, ISEAUTO. *Electr. Control Commun. Eng.*, 2018, **14**(1), 81–88.
5. Sell, R., Leier, M., Rassõlkin, A., and Ernits, J.-P. Self-driving car ISEAUTO for research and education. In *Proceedings of the 19th International Conference on Research and Education in Mechatronics (REM), Delft, Netherlands, June 7–8, 2018*. IEEE, 2018. https://ieeexplore.ieee.org/document/8421793
6. Rassõlkin, A., Hõimoja, H., and Teemets, R. Energy saving possibilities in the industrial robot IRB 1600 control. In *Proceedings of the 7th International Conference-Workshop Compatibility and Power Electronics (CPE), Tallinn, Estonia, June 1–3, 2011*. IEEE, 2011. https://ieeexplore.ieee.org/document/5942236
7. Väljaots, E., Sell, R., and Rimasauskas, M. Unmanned Ground Vehicle Energy Efficiency Validation in Territory Surveillance Mission. *Solid State Phenom.*, 2016, **251**, 164–170.
8. Väljaots, E., Lehiste, H., Kiik, M., and Leemet, T. Soil sampling automation using mobile robotic platform. *Agron. Res.*, 2018, **16**(3), 917–922.
9. Rassõlkin, A. and Vodovozov, V. A test bench to study propulsion drives of electric vehicles. In *Proceedings of the 8th International Conference-Workshop Compatibility and Power Electronics (CPE), Ljubljana, Slovenia, June 5–7, 2013*. IEEE, 2013. https://ieeexplore.ieee.org/ document/6601169
10. Kato, S., et al. Autoware on board: enabling autonomous vehicles with embedded systems. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, April 11–13, 2018*. IEEE, 2018. https://ieeexplore.ieee.org/document/8443742
11. Sell, R., Coatanéa, E., and Christophe, F. Important aspects of early design in mechatronic. In *Proceedings of the 6th International Conference of DAAAM Baltic, Industrial Engineering, Tallinn, Estonia, April 24–26, 2008*, 177−182. http://innomet.ttu.ee/daaam08/Online/Design%20Engineering/Coatanea-Sell.pdf

Mobiilsete robotite modulaarne arhitektuur

Raivo Sell, Eero Väljaots, Tengiz Pataraia ja Ehsan Malayjerdi

Modulaarne arhitektuur on võtmetähtsusega mobiilsete robotite arendusel, kui tegemist on multiotstarbelise robotsõidukiga. Artiklis on tutvustatud modulaarsuse kontseptsiooni keskklassi mobiilsele robotile, kus peamine fookus on tarkvaraline modulaarsus, iseorganiseeruvus koos lihtsalt ühendatavate lisaseadmetega, andurite väljundite kombineerimine ja riistvaraline modulaarsus. Väljatöötatud kolmetasandiline tarkvara raamistik sisaldab vahevara ja seda on rakendatud kahe erineva mobiilse roboti juhtsüsteemides.

Appendix 2

II

Ruxin Wang, Raivo Sell, Anton Rassolkin, Tauno Otto, and Ehsan Malayjerdi. Intelligent functions development on autonomous electric vehicle platform. Journal of Machine Engineering, 20, 2020

Journal of Machine Engineering, 2020, Vol. 20, No. 2, 114–125
ISSN 1895-7595 (Print) ISSN 2391-8071 (Online)

Received: *11 December 2019* / Accepted: *22 January 2020* / Published online: *24 June 2020*

autonomous vehicles, safety engineering,
robots, human machine interaction

Ruxin WANG[1]
Raivo SELL[1]
Anton RASSOLKIN[1]
Tauno OTTO[1*]
Ehsan MALAYJERDI[1]

INTELLIGENT FUNCTIONS DEVELOPMENT ON AUTONOMOUS ELECTRIC VEHICLE PLATFORM

Autonomous driving is no longer just an idea of technology vision instead a real technical trend all over the world. The continuing development to a further level of autonomy requires more on mobile robots safety while bringing more challenges to human-vehicle interaction. A robot autonomous vehicle (AV) as a research platform operates an experimental study on human-AV-interaction (HAVI) and performs a novel method for mobile robot safety assurance. Not only autonomous driving technology itself but human cognition also performs an essential role in how to ensure better autonomous mobile robot safety. A Wizard-of-Oz experiment in the university combing a survey-based study indicates public attitudes towards driverless robot vehicles. HAVI experiment have been carried through light patterns designed for experiment. This paper presents an attempt to investigate humans' acceptance and emotions as well as a validation to bring the mobile robot vehicle to a high-level autonomy.

1. INTRODUCTION

As more improvement coming from every aspects of technology, the greatest barrier standing in the way of the advent of fully autonomous robot vehicles lies in building people's trust in the machine and enhancing their sense of safety. Companies and researchers are trying hard to cut down the cost which makes the question no longer is if it's possible to enable autonomous vehicles, it's down to, will we allow this foreseeable future to happen. A team from Stanford found that, for AV, most humans managed to make crossing decisions based on vehicle cues alone instead of communicating via driver cues [1]. Master students from Chalmers University [2] raised the opinion that vehicle movement alone is not enough to compensate for the loss of driver cues in AVs and the creation of specialized interfaces for communicating with humans is needed. Scholars in US [3] investigated intent communication cues for AVs by comparing the effectiveness of various methods of presenting human-vehicle

[1] Tallinn University of Technology, Tallinn, Estonia
[*] E-mail: tauno.otto@taltech.ee
https://doi.org/10.36897/jme/117787

street crossing information. All the investigations above have consolidated the necessity of Human-Autonomous Vehicle-Interaction research. The main aim of the paper is to give an overview on current state of the art on the topic of robot human-autonomous vehicle interaction (HAVI). A Wizard of Oz [4] experiment is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous has been done during performed studies. The results of experiment are analysed and discussed.

According to ERTRAC (European Road Transport Research Advisory Council) [5], the following chapters (Fig. 1) list the main challenges and objectives on the path to higher levels of automation. To this, policy and societal aspects must be firstly addressed to ensure proper user information and acceptance, then it will trigger the necessary regulatory adaptations.

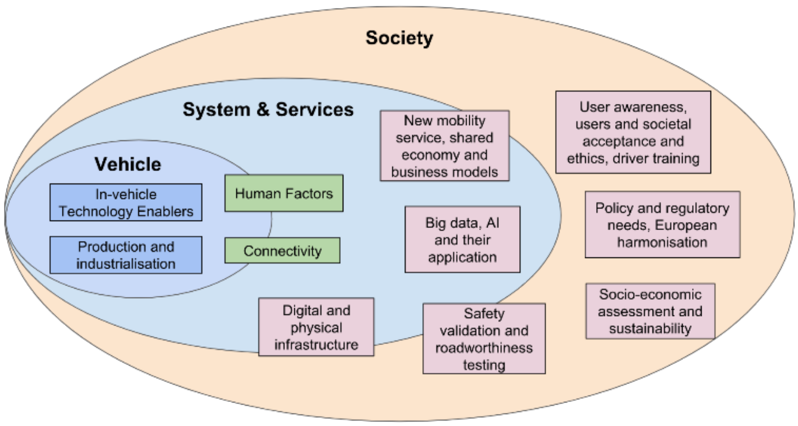

Fig. 1. Main challenges and objectives on the path to higher levels of automation

At current research was validated safety during roadworthiness testing, while using campus closed area as a physical infrastructure. Before that the robot, road and surrounding landscape was digitalised and possible scenarios were simulated. Also, user awareness and acceptance of self-driving robots were topics of interest. In-vehicle technology enablers are described in more detail at next chapters.

Delft University of Technology [6] has implemented a 63-question online survey among 5000 respondents in 109 countries assessing correlation coefficients with the countries' objective road safety statistics and countries' developmental status in terms of education and gross domestic product (GDP) per capita. However, since the highly and fully automated vehicles are not available yet, the results of the survey comes more out of participants' imagination of the future automation and should be only taken as a reference.

2. SELF-DRIVING PLATFORM DEVELOPMENT

Platform design is shown in Fig. 2. All moulds and panel frames are specifically designed for this project. Body design of the platform takes also into account a location of sensors required for the autonomous cruising. Sidebar lights are implemented by using

RGB LED matrix where all individual LEDs are independently driven. It means that the vehicle can easily change lights from red to white on both ends as well as signal custom figures separately in any light panel. The LED element is the main HAVI device communicating with humans. The vehicle has the following technical parameters as shown in the Table 1.

Fig. 2. Electric autonomous robot platform design

One of the intentions of the project was to keep the development open and make the project accessible to new developers and students. An open source software platform is critical in that respect. The main software framework is Autoware which is an autonomous driving stack running on top of the Robot Operating System (ROS).

Table 1. Last-mile autonomous robot platform

Type	Cargo or passenger
Cruising speed, km/h	20
Turning radius, m	9
Main motor, kW	47
Battery, kWh	16
Dimensions	
Height, m	2.3
Length, m	3.5
Width, m	1.3
Sensors	Pieces
LiDAR Velodyne VLP-16	2
Safety LiDAR	1
Ultrasonic sensors front and back	8
Short distance radar	1
Cameras	8
RTK-GNSS	1
IMU	1

As AVs can pose a danger to human life, special attention was targeted to the system and integration testing of the software [7]. The platform software architecture and message flow is shown in Fig. 3.

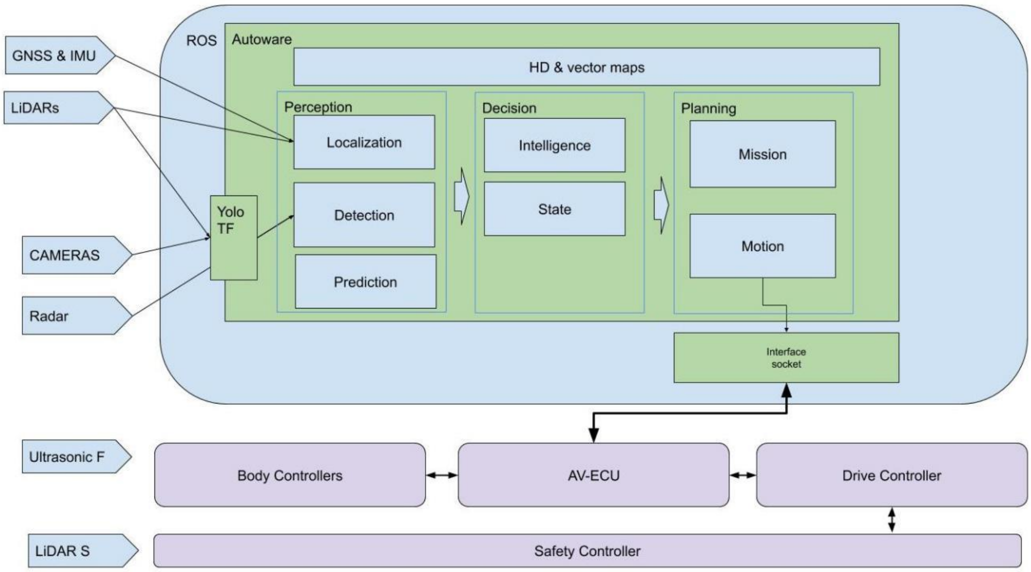

Fig. 3. Autonomous platform software architecture and message flow

The platform takes inputs from following types of sensors:

1) LiDAR, radar and camera inputs are used for localization, obstacles detection, object classification and safety;
2) Global Navigation Satellite System (GNSS) is used for localization correction;
3) Ultrasonic sensors are used for manoeuvring and second level obstacle detection
4) Output commands are steering angle and linear velocity which are sent to the low -level controllers over UDP messages.

The robot platform has four Basler Pylon cameras for object detection tasks. One in the front, one on the top and two on both sides. A ROS package for real-time object detection based on YOLO is applied for object detection.

The pre-trained model of the convolutional neural network can detect pre-trained classes including the dataset from Pascal Visual Object Classes (VOC) [8] and Common Objects in Context (COCO) [9]. This package publishes number of detected objects and their position. Based on detected and classified objects the platform changes the lights according to identified humans.

Lights on the platform are designed with the idea of being able to display different symbols for the humans in addition to the regular lights. This gives to the humans a better idea of what the vehicle does or plans to do. To achieve this, a custom light is designed, consist of 512 red, green, and blue (RGB) light-emitting diode (LED) as a LED matrix module.

The LED matrix module is covered with a diffuser and the assembly is located behind the windscreen back and front. The LED matrix module is based on WS2812 intelligent control LEDs. Each LED can be independently addressed as an RGB pixel that can achieve 256 levels of brightness and 16 777 216 colours in total with a scanning frequency of 400 Hz.

All RGB LED matrix modules are controlled by dedicated electronic control unit (ECU). The module ECU supply a high current 5 V power to the LEDs and drives all LEDs individually. All four modules ECUs and a light control ECU are connected on the vehicle CAN bus where messages of requested light behaviour are received. A PC sends status messages through a master controller to light control ECUs which generates specific messages to each LED modules (Fig. 4).

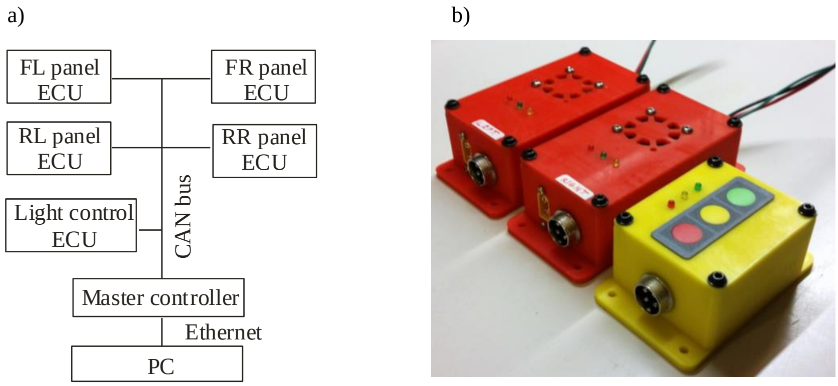

Fig. 4. Hardware architecture (a) of two panel electronic control unit (ECUs) with light control ECU and light control ECU module (b)

3. HUMAN-AUTONOMOUS VEHICLE-INTERACTION (HAVI) EXPERIMENT

3.1. EXPERIMENT DESIGN

With the existing robot platform and the current researches going on [7, 10, 11], it is necessary to go further into the communication between human beings and AVS [12]. The aim of this experiment is to gather the knowledge about:

1) What's the attitudes of people towards the general safety of autonomous driving technology?
2) How do people feel about the interaction with driverless vehicles especially the campus shuttle minibus?
3) How to bring up a universal human-machine language for more harmonious interactions?

Since a single LED panel is illuminated by 16×8 px, three patterns are chosen to be finally transplanted onto the robot platform as shown in Table 2. Performance of designed line patterns is shown in Fig. 5. Arrows pattern (Fig. 5a) address a message for human to start crossing the crosswalk and indicates Zebra Line pattern (Fig. 5b) while human crossing crosswalk.

A questionnaire was created to collect feedbacks and personal data from humans interacted with robot platform. The factors mentioned in the questionnaires are mostly based on the previous studies on AV. In [13] gender difference was found to strongly affect the trust of whether AVs can make a correct decision and stop for humans via their technology. Authors in [6] claim that typically, younger people express a more positive attitude towards automated vehicles however, as age increases, risk acceptance decreases [14, 15]. People with higher educational backgrounds tend to be more favour of AV than those less educated as shown in [16]. In terms of Industry 4.0 state of the art research has been successful in development of digital twins of industrial robots [17]. The AV have enormous potential to be tested and developed as digital twins regarding human-machine interface as well.

Table 2. Illustration of light design pattern for the HAVI experiment

Pattern	Zebra Line	Arrows	Cross
Action	Pass	Pass	Stop
Visualization			

a) b)

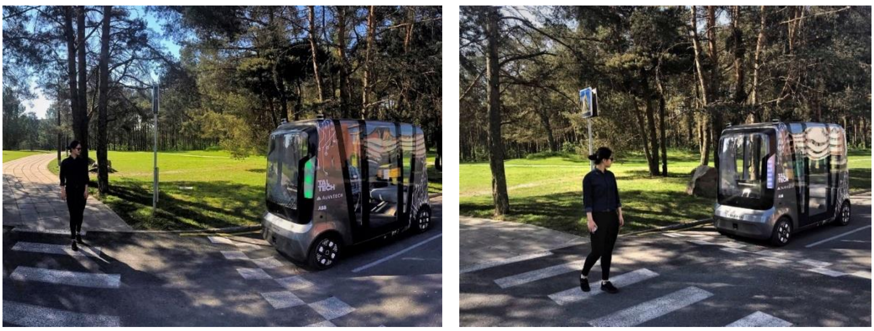

Fig. 5. Performing experiment with pattern Arrows (a) and Zebra Line (b)

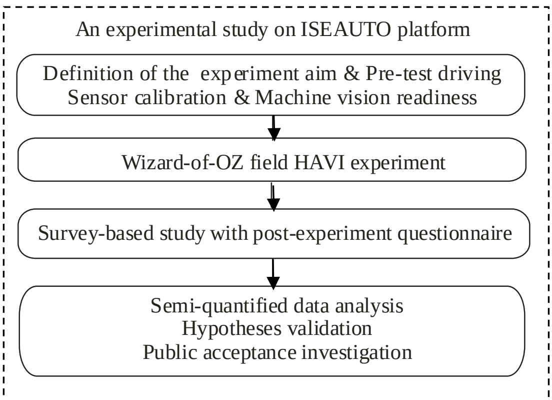

Fig. 6. The experiment plan on robot platform

Experimental HAVI study plan on robot platform is shown in Fig. 6. All test driving was mostly operated during daytime on the designated roads in the university. There were clear signs at the crossroads where interactions happen between humans and the vehicle near the zebra line. Participants in the experiments were randomly chosen when humans are crossing the road and a following questionnaire will be filled by the human.

3.2. THEORETICAL ANALYZING MODEL

To better reveal the mechanism of this HAVI experiment, the initial TAM concept [18] (Fig. 7a) as a base and some other related models applicated on AVs [19, 20] have been studied to help build up our technology model (Fig. 7b) thus analyse the necessity of the whole experiment. Questions covered in the questionnaire are also designed out of the intention to collect data for validating the model.

The initial trust (IT) has been studied in human-automation interaction [21] as a key element [22] and there has also been empirical support in AV filed [23] about the drivers' trust in the technology. However, the trust from the humans hasn't been widely introduced to the assessment of the HMI design on AV. A German version questionnaire named Trust in Automation (TiA) was adopted for exploring the trust in autonomous driving where five subscales: Reliability/Competence, Familiarity, Trust, Understanding, Intention of Developers built up the criteria. However, most of the current researches are discussing IT in the autonomous technology itself instead of viewing from the efficiency of interactions between human and AV.

Prior beliefs and experiences are based on empirical evidence acquired by means of the senses, particularly by observation and documentation of patterns and behaviour through experimentation [24]. As mentioned in the research targeting human behaviour, humans often make risky decisions in assessing the danger that vehicles pose [25]. Those decisions are generally made from their previous experience and empirical knowledge (EK).

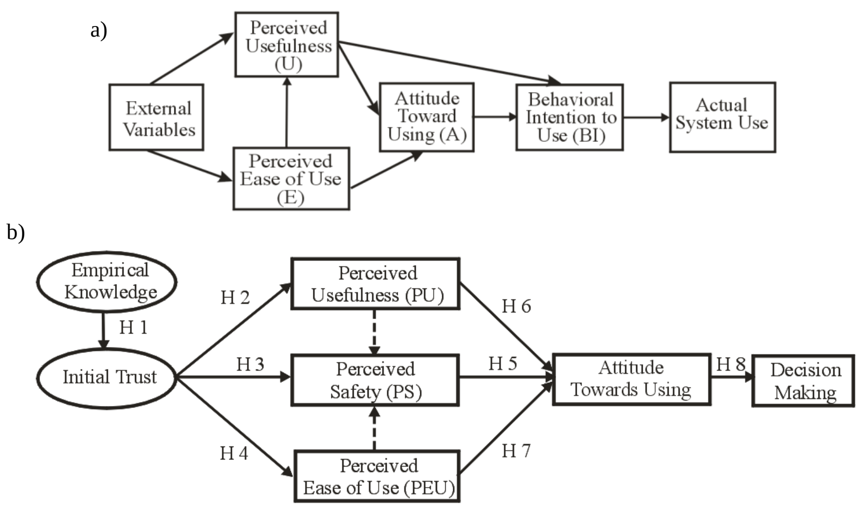

Fig. 7. Original TAM contents (a) [18] and model proposal of HAVI experiment (b)

Attitude toward Using (ATT) refers to an individual's positive or negative feelings towards using a technology. People with a positive attitude towards a technology tend to have a higher intention to use it [26].

Perceived Usefulness (PU) was defined as degree to which a person believes that using a particular system would enhance his or her job performance [27]. Under the current circumstance, it's hard to promise the decision accuracy of AV when it encounters other road users especially humans who are more vulnerable. Although the autonomous technology has been developing for decades but accidents are still inevitable when it's hard to predict human behaviour and relate it a similar context for AV to understand.

Perceived Ease of Use (PEU) was defined as extent to which a person believes that using particular system would be free of effort. The hardest part of HMI design on the vehicle is to make sure the message is well delivered and understood by receivers in both ways. For one thing, the vehicle can present its understanding of the road condition via the LED lights on. For another, the humans can notice the lights and make right decisions to ensure their own safety.

The attitude towards the HMI design on the AVs can directly reflect humans' acceptance of the designing concept and it is also the basis of validation on this road safety approach. To create a common language between human and AV, it's easier to make the vehicle more human-like instead of changing people's mindset for understanding the machine [28, 29]. The result of Decision Making (DM) helps to assess the whole experiment whether the interaction improves mutual understanding and lower the risk of fatal collision and misjudging.

The main conclusions as hypotheses in this model are empirical knowledge is strongly related to Initial Trust, that has positive effect on Perceived Safety, Usefulness and Ease of Use; which in turn has a positive effect on Attitude toward Using that is strongly related to Decision Making.

1) H1: EK is strongly related to IT.
2) H2-H4: IT influences PS, PU and PEU and PU, PEU both link to PS.
3) H5-H7: PS, PU and PEU influences ATT.
4) H8: ATT is strongly related to DM.

4. ANALYSIS

To better ensure the safety during the operation, although the robot platform is able of fully autonomous functioning, when carrying out the experiment, a human driver still sat in the car using controller to manipulate the driving. However, the operator himself pretended to be a passenger and hid his hands without being spot to manually drive the self-driving car. Thus, the experiment was using a Wizard-of-Oz method for HAVI and according to the survey afterwards, all the subject participants reckoned the car was in autonomous mode. Thus, for an experimental study, this approach doesn't affect the actual results since safety is the priority during the experiments.

Respondents divided equally between male and female, giving a good example for non-bias analysis and to avoid the gender influence. However, when comes to the feeling towards the interaction with AV, the results surprisingly almost present a tie between being absolutely fine and cautious. Colour signals are preferred as the message delivered during HAVI by most of the participants. However, due to its new trial on HAVI, the lights design of the platform was only clearly understood by half of the respondents while the rest were confused or had no idea of the meaning. The full results are presented in Fig. 8.

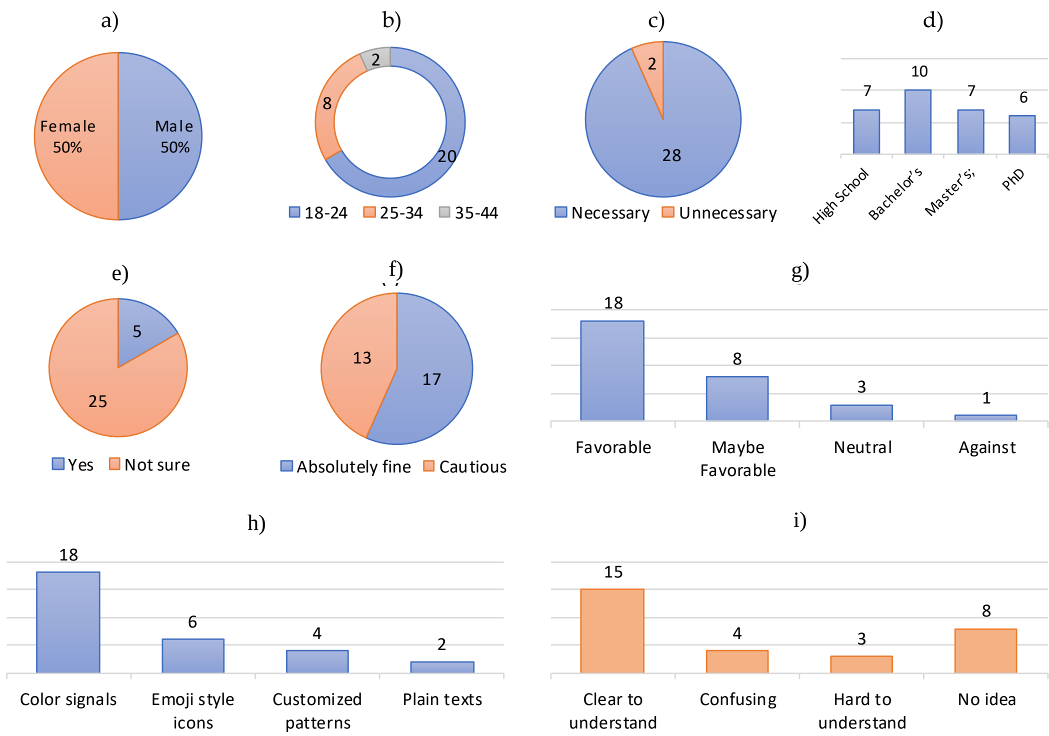

Fig. 8. Results of the survey: Gender (a), Age (b), Opinion towards the necessity of HAVI experiment (c), Level of education (d), Willingness of sharing roads with AV (e), Feeling towards the interaction with self-driving vehicles (f), Attitude towards driverless vehicles on the roads (g), Preference for the message type during an interaction (h), Feedback of platform lights Design (i)

Combined with the model proposed before, the results can be further analysed for each hypothesis in the previous chapter.

1) The education level and the age can both reflect an individual's knowledge basis and affect their perception on the surroundings. Telling from the statistics, younger people with less education experience tend be more favourable towards AV and open to share roads with them while older generations with higher education tend to have more concern and appear to be reluctant to the trendy technology. They prefer to wait and observe until the readiness of mature products in the market.
2) Those who trust more in AV also tend to feel it more necessary to carry out HAVI experiment and positive attitudes towards AV can bring them more confidence during the HAVI. Also, the acceptance of the technology affects people's choice on the product/ technology design, here was the preference for the message type.

Besides these, better interpreting of the pattern and the activeness of engaging in HAVI can help the individual feel less stressed during the experiment.

3) Participants who also are the end users of the product/technology are achieving a more successful interaction. It has meanings from both sides where on one hand people should feel ease to understand thus make next-step decisions when seeing the lights, while on the other the car should make the most of the lights to harmonize HAVI process.
4) A commonly accepted machine language, car lights in our case, can increase the chance of people to accept the technology. HAVI has two main bodies, namely the human and the autonomous vehicle, so either side fails to pass through their understanding would lead to a bad decision making. The algorithm can optimize the decision made by vehicles via machine vision while this information should be perceived by human beings.

5. CONCLUSIONS

The HAVI experiment described above on self-driving electric vehicle was carried out in campus area thus the result of the experiment is somewhat site specific. However, by taking usages of the educational research vehicle, this HAVI experiment provides an approach to improve road safety for AV and help researchers to get an overview of how people react to the concept of creating a common language between humans and AV. In practice remotely controlled multi robot environment as tested in [30] can be implemented also for larger mobile autonomous robots used in HAVI experiment.

With the increasing number of robotised AVs, more effort from all sectors is needed and emphasized to ensure safety. One of the goals of the paper was to develop a novel on-vehicle light design which can inform humans of the real-time decision made by robotised AV. Sensors on an AV should correctly identify humans and deliver a clear information in time indicating its movement.

This research consists two parts: a series of field experiments and a questionnaire-based survey right after each independent experiment. It’s notable that the importance of safety always comes first when people encounter AV unexpectedly and have to instantly make a subconscious decision during the interaction. It is also clearly seen out of the experiment that the communication between AVs and humans needs to be taken much more seriously by vehicle manufacturers as well as research institutions. AV without a driver is much more challenging than expected. Defining a universal and simple driving HAVI is clearly not sufficient. Visual signalling in combination audio and other possible channels need to be experimented and designed for future autonomous vehicles.

REFERENCES

[1] ROTHENBUCHER D., LI J., SIRKIN D., MOK B., JU W., 2016, *Ghost Driver: A Field Study Investigating the Interaction Between Humans tnd Driverless Vehicles*, 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 795–802.

[2] LAGSTRÖM T., LUNDGREN V.M., 2015, *AVIP – Autonomous Vehicles Interaction with Humans*, Chalmers University, Retrieved from http://publications.lib.chalmers.se/records/fulltext/238401/238401.pdf
[3] CLAMANN M., AUBERT M., CUMMINGS M.L., 2017, *Evaluation of Vehicle-to-Human Communication Displays for Autonomous Vehicles*, 96th Annual Transportation Research Board Meeting, Washington DC.
[4] DAHLBÄCK N., JÖNSSON A., AHRENBERG L., 1993, *Wizard of Oz Studies – Why and How*, Knowledge-Based Syst., 6/4, 258–266.
[5] *Automated Driving Roadmap*, 2015, Brussels, Belgium.
[6] KYRIAKIDIS M., HAPPEE R., DE WINTER J.C.F., 2015, *Public Opinion on Automated Driving: Results of an International Questionnaire Among 5000 Respondents*, Transp. Res. Part F Traffic Psychol. Behav., 32, 127–140.
[7] SELL R., LEIER M., RASSÕLKIN A., ERNITS J.-P., 2018, *Self-Driving Car ISEAUTO for Research and Education*, 19th International Conference on Research and Education in Mechatronics (REM 2018).
[8] EVERINGHAM M., 2015, *The Pascal Visual Object Classes Challenge: A Retrospective*, Int. J. Comput. Vis., 111/1, 98–136.
[9] LIN T.Y., 2014, *Microsoft COCO: Common Objects in Context*, Springer, Cham, 740–755.
[10] RASSÕLKIN A., GEVORKOV L., VAIMANN T., KALLASTE A., SELL R., 2018, *Calculation of the Traction Effort of ISEAUTO Self-Driving Vehicle*, 25th International Workshop on Electric Drives: Optimization in Control of Electric Drives (IWED), 1–5.
[11] RASSÕLKIN A., VAIMANN T., KALLASTE A., 2018, *Propulsion Motor Drive Topology Selection for Further Development of ISEAUTO Self-Driving Car*, 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON).
[12] SELL R., RASSÕLKIN A., RUXIN W., OTTO T., 2019, *Integration of Autonomous Vehicles and Industry 4.0*, Proceedings of the Estonian Academy of Sciences, 68/4, 389–394.
[13] DEB S., WARNER B., POUDEL S.R., BHANDARI S., 2016, *Identification of External Design Preferences in Autonomous Vehicles*, Proceedings of the Industrial and Systems Engineering Research Conference, Anaheim, California.
[14] TURNER C., MCCLURE R., 2003, *Age and Gender Differences in Risk-Taking Behaviour as an Explanation for High Incidence of Motor Vehicle Crashes as a Driver in Young Males*, Inj. Control Saf. Promot., 10/3, 123–130.
[15] LIU P., ZHANG Y., HE Z., 2019, *The Effect of Population Age on The Acceptable Safety of Self-Driving Vehicles*, Reliab. Eng. Syst. Saf., 185, 341–347.
[16] ALESSANDRINI A., ALFONSI R., SITE P.D., STAM D., 2014, *Users' Preferences Towards Automated Road Public Transport: Results from European Surveys*, Transp. Res. Procedia, 3, 139–144.
[17] KUTS V., OTTO T., TÄHEMAA T., BONDARENKO Y., 2019. *Digital Twin Based Synchronised Control and Simulation of The Industrial Robotic Cell Using Virtual Reality*, Journal of Machine Engineering, 19/1, 128−145.
[18] DAVIS F.D., 1989, *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*, MIS Q., 13/3, 319–340.
[19] XU Z., 2018, *What Drives People to Accept Automated Vehicles? Findings From a Field Experiment*, Transp. Res. Part C Emerg. Technol., 95, 320–334.
[20] ZHANG T., 2019, *The Roles of Initial Trust and Perceived Risk in Public's Acceptance of Automated Vehicles*, Transp. Res. Part C Emerg. Technol., 98, 207–220.
[21] PARASURAMAN R., SHERIDAN T.B., WICKENS C.D., 2000, *A Model for Types and Levels of Human Interaction with Automation*, IEEE Trans. Syst. Man, Cybern. – Part A Syst. Humans, 30/3, 286–297.
[22] LEE J.D., SEE K.A., 2004, *Trust in Automation: Designing for Appropriate Reliance*, Hum. Factors J. Hum. Factors Ergon. Soc., 46/1, 50–80.
[23] KÖRBER M., BASELER E., BENGLER K., 2018, *Introduction Matters: Manipulating Trust in Automation and Reliance in Automated Driving*, Appl. Ergon., 66, 18–31.
[24] WILKINSON R.G., PICKETT K. E., 2006, *Income Inequality and Population Health: A Review and Explanation of the Evidence*, Soc. Sci. Med., 62/7, 1768–1784.
[25] DEB S., RAHMAN M.M., STRAWDERMAN L.J., GARRISON T.M., 2018, *Pedestrians' Receptivity Toward Fully Automated Vehicles: Research Review and Roadmap for Future Research*, IEEE Trans. Human–Machine Syst., 48/3, 279–290.
[26] MARANGUNIĆ N., GRANIĆ A., 2015, *Technology Acceptance Model: A Literature Review from 1986 to 2013*, Univers. Access Inf. Soc., 14/1, 81–95.
[27] DAVIS F.D., BAGOZZI R.P., WARSHAW P.R., 1989, *User Acceptance of Computer Technology: A Comparison of Two Theoretical Models*, Management Science, 35/8, 982–1003.
[28] MOKHTARZADEH A.A., YANGQING Z.J., 2018, *Human-Robot Interaction and Self-Driving Cars Safety Integration of Dispositif Networks*, IEEE Inter. Conference on Intelligence and Safety for Robotics (ISR), 494–499.

[29] BROOKS R., 2017, *The Big Problem With Self-Driving Cars Is People*, IEEE Spectrum, Retrieved from https://spectrum.ieee.org/transportation/self-driving/the-big-problem-with-selfdriving-cars-is-people
[30] SELL R., OTTO T., 2008, *Remotely Controlled Multi Robot Environment*, Proceedings of 19th EAEEIE Annual Conference, Tallinn, Estonia, 20−25.

Appendix 3

III

Mohsen Malayjerdi, Bariş Cem Baykara, Raivo Sell, and Ehsan Malayjerdi. Autonomous vehicle safety evaluation through a high-fidelity simulation approach. Proceedings of the Estonian Academy of Sciences, 70(4):413–421, 2021

Proceedings of the Estonian Academy of Sciences,
2021, **70**, 4, 413–421
https://doi.org/10.3176/proc.2021.4.07
Available online at www.eap.ee/proceedings

MANUFACTURING ENGINEERING

Autonomous vehicle safety evaluation through a high-fidelity simulation approach

Mohsen Malayjerdi[a*], Barış Cem Baykara[a], Raivo Sell[b,a] and Ehsan Malayjerdi[a]

[a] Mechanical and Industrial Engineering Department, Tallinn University of Technology, Tallinn, Estonia
[b] FinEst Twins Smart City Center of Excellence, Tallinn University of Technology, Tallinn, Estonia

Received 15 June 2021, accepted 13 July 2021, available online 2 November 2021

Abstract. The autonomous vehicle (AV) industry aims to design strategic plans to ensure the safety of the developed systems before their mass deployment. Real-road testing is shown to be impractical for validating these systems as it requires many years if not decades of testing in different environmental conditions. For solving this issue, the method should be complemented with simulation. The primary goal of this research was to develop advanced techniques in the safety validation area by using end-to-end simulation technologies. In this study, we present a simulation approach for safety evaluation of an AV shuttle, iseAuto, currently operating at the Tallinn University of Technology campus. We created a virtual environment by using geospatial data from the specified path on the university campus that includes all relevant features. Then, we converted the map to a 3D format applicable for the SVL simulator. Also, we provided the AV 3D model to use in the simulation and equipped it with the SVL virtual sensors to provide data for the Autoware perception algorithms, which is the control software of the shuttle. To show the efficiency of the proposed method, we designed two overtaking scenarios and observed the AV behaviour under the test. Finally, we demonstrate how the system enables us to evaluate AV's decision-making performance and safety in different situations.

Key words: autonomous vehicle, simulation, safety validation, high-fidelity simulator.

1. INTRODUCTION

Development of autonomous vehicles is one of the top trends in the automotive industry and the technology has been evolved to make them safer. Thus, engineers are facing new challenges, especially in moving toward Levels 4 and 5 of the Society of Automotive Engineers (SAE). To place autonomous vehicles (AVs) on roads and evaluate the reliability of their technologies, they have to be driven billions of miles [1]. It would take a long time to achieve this, unless with the help of simulation. Furthermore, due to the past real crash cases of AVs, a high-fidelity simulator has become an efficient and alternative approach to provide different testing scenarios for controlling these vehicles, also enabling safety validation before real-road driving [2–5]. Different high-resolution virtual environments can be developed for simulators by using cameras or lidars to simulate the scenarios as close to the real world as possible [6]. Also, virtual environment development enables us to customize and create various urban backgrounds for testing the vehicle. Creating a virtual copy of an existing intelligent system is a common approach nowadays, called a digital twin [7,8]. Extensive research and development, such as in [9,10] or [11], has been performed on AVs in recent years involving simulation. However, most of that has employed a low-fidelity simulator that cannot be a reliable reference for safety validations.

In this paper, we focus on the utilization of a high-fidelity simulator for an AV shuttle at Tallinn University

* Corresponding author, momala@taltech.ee

of Technology (TalTech), Estonia. The TalTech AV research group is well known for its AV shuttle, iseAuto [12], which is operational on the campus for experimental research purposes (Fig. 1). The vehicle was designed and developed from scratch by implementing the previously proposed mechatronic design methodology [13–15] with a special focus on early design stages. The first prototype development was a joint venture with TalTech and the local industry Silberauto [16]. This shuttle is controlled by Autoware [17], a Robotic Operating System (ROS) based platform for self-driving vehicles.

The overall research project was planned to be executed in two stages. First, the virtual environment was built based on the campus AV road area, where most of our real experiments take place, to create the simulation framework. We used geospatial images to generate the environment as a Unity terrain. Among different modern AV simulators such as CARLA [18], LGSVL (in 2021 the name was changed to SVL) [19] and Gazebo, we opted for SVL to be our simulator due to its compatibility with our control software (Autoware) and our terrain generation platform Unity. Another reason was to create different scenarios and perform software-in-the-loop (SIL) simulation by connecting Autoware with SVL. This enables us to find a better sensor configuration and settings in addition to the verification of the decision-making system that leads to safety assessment.

2. SIMULATOR

Simulation has been widely used in vehicle manufacturing, particularly for mechanical behaviour and dynamical analysis. However, AVs demand more due to their specific nature. Simulation in various complex environments and scenarios involving other road users with different sensor combinations and configurations enables us to verify their decision-making algorithms. One of the most popular robotic simulator platforms is Gazebo. It is based on ROS and utilizes physics engines and various sensor modules suitable for autonomous systems. Nevertheless, Gazebo lacks modern game engine features such as Unreal and Unity, which give the power to create a complex virtual environment and realistic rendering.

CARLA and SVL, on the other hand, are modern open-source simulators based on these game engines, Unreal and Unity respectively, which also have good compatibility with our AV stack Autoware. However, comparing these two is beyond the scope of our discussion, but we selected SVL as our simulator mainly because of its compatibility with our terrain generator Unity.

Figure 2 shows a full map of the simulation workflow and different layers in the simulator as well as the control software (Autoware). Vehicle 3D model and the virtual environment, which were built inside Unity, were imported to the simulator. The simulator allows cus-

Fig. 1. TalTech iseAuto – an AV shuttle.

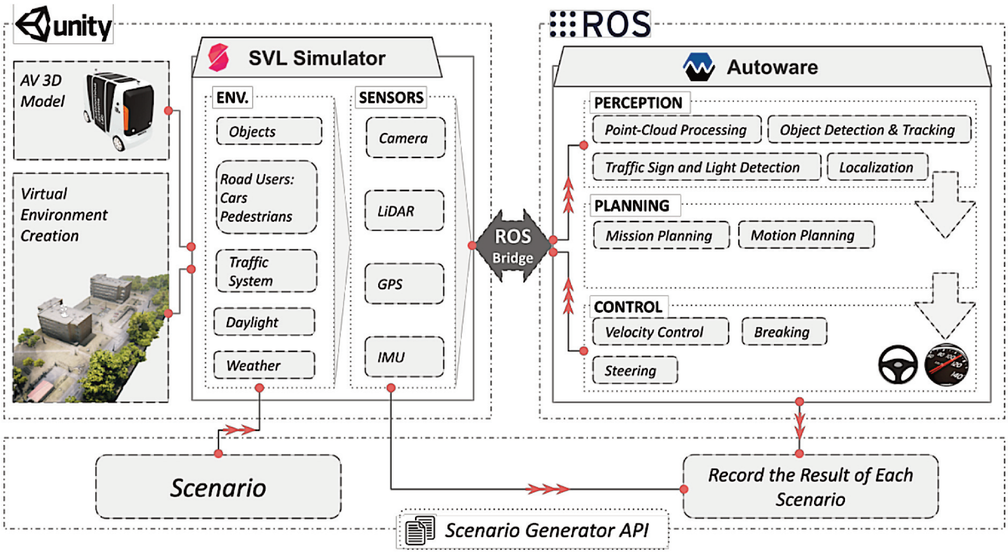

Fig. 2. High-level architecture of the simulation and the AV system.

tomizing the environment to create different scenarios such as adding/removing other road users, inserting traffic systems, adjusting the time of day and the weather of the scene. There is a scenario generator API that connects to the simulator and creates various scenarios according to the user definition. Then, the virtual sensors used in the AV provide information for the perception of the environment. This information is transferred via a ROS bridge to our control software platform to use in the perception algorithms for the localization and detection. Perception results are used in the Autoware planning section which makes the control commands for the AV. These control commands are sent back to the simulator via the ROS bridge to navigate the vehicle inside the simulator. Furthermore, in the case of any failure in any scenario, some sensor data and vehicle navigation commands are recorded for further study.

The iseAuto 3D model and its lidar sensors are illustrated in Fig. 3. A Velodyne VLP-32 was installed at the top front of the shuttle and a VLP-16 at the top back. Two Robosense Bpearl were installed at the left and right sides of the vehicle. Furthermore, to cover the blind zone in front of the vehicle, a RS-LiDAR-16 was installed in the front bumper. This lidar configuration creates a good point-cloud coverage around the vehicle for perception purposes.

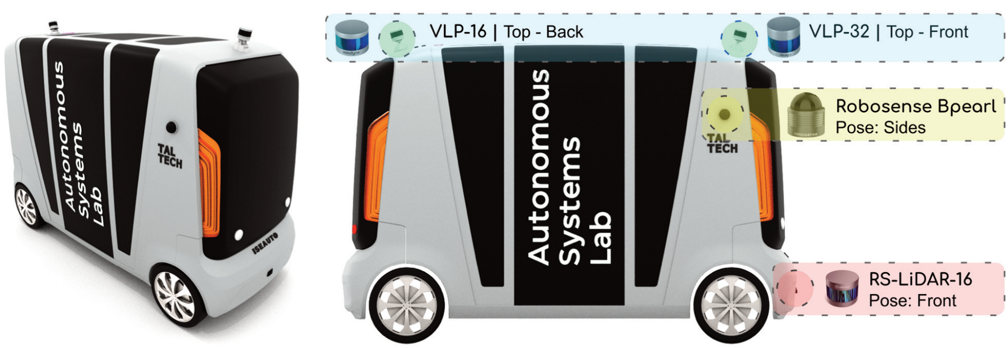

Fig. 3. iseAuto simulated model with different lidars installed.

3. VIRTUAL ENVIRONMENT CREATION

The fierce competition in the gaming industry nowadays has generated many features for game engines. These engines can simulate physics and thus be exploited as simulators aside from game development. SVL and others have already taken advantage of the aforesaid engines and created a framework for testing autonomous vehicles within such physics simulators. Even though these simulators provide some basic tools and assets to get started, it is still not sufficient. To make it more realistic, we need to have real-world terrains simulated.

3.1. Workflow

In order to create a terrain for simulation, the area to be simulated has to be mapped. There are certain steps to follow:

- Data Collection and Processing;
- Terrain Generation.

Data is collected by aerial photography and processed further to obtain a dense point-cloud of the area to be mapped. The point-cloud is then processed through a process called segmentation. Lastly, it is fed into Unity as an input for terrain generation.

3.2. Data collection and processing

Aerial imagery of the area to be mapped has to be captured with a camera drone. The images are captured at a grid flight path, which ensures that the captured images cover different sides of a subject. In order to make sure that the images have maximum coverage, the flight path is followed three times from different camera angles but at a constant altitude. Taking aerial photos is one of the most important steps in the mapping process as it will significantly affect the outcome of the process and the amount of work to be done to process those images. There are also external factors that may affect the quality of the pictures taken off the ground. Weather conditions and scene lighting may create artifacts on the pictures, which may disturb the photogrammetric process. The images taken are georeferenced by the drone and if necessary, a stationary Real Time Kinematic (RTK) device can be utilized to mitigate errors and shift the positioning data stamped on the pictures. The onboard IMU provides the pictures with orientation, so that later they can be stitched together and used for photogrammetric processing. Third party software aligns and creates the dense point-cloud from the pictures that were captured. Once the dense point-cloud is created, the segmentation and classification of the points is needed in order to separate unwanted objects and vegetation from the point-cloud data. However, removing is not to be performed in the point-cloud as the positional information they provide for their respective objects will aid terrain generation to spawn details. Figure 4 shows the three main steps to generate the Unity train from geospatial data.

3.3. Terrain generation

Digitalization of a real-life environment can be used for simulating AVs in countless different scenarios without taking the vehicle out for once. Terrain generation from point-cloud is performed right in Unity. In-house developed plugin reads a pre-classified point-cloud file, and based on chosen parameters it creates a normal map, a heightmap and a colour map to utilize in conjunction with the Unity's terrain engine to create realistic environments.

4. SIMULATION AND SAFETY ASSESSMENTS

Based on the simulation architecture illustrated in Fig. 2, the AV can be run inside the virtual environment. In collaboration with Florida Polytechnic University and Embry-Riddle Aeronautical University, we developed a regime for creating edge-case scenarios for safety validation of the shuttle working on our campus pilot road

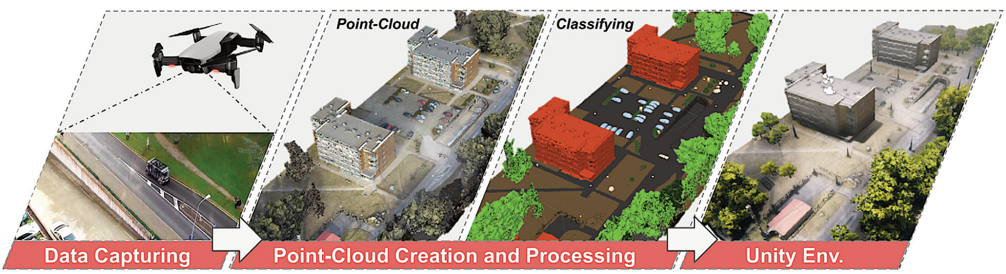

Fig. 4. Steps for virtual environment generation.

[20]. Now, by using a high-fidelity simulator we can simulate different scenarios close to real life in order to evaluate the control algorithm performance and safety. In terms of defining these scenarios, SVL provides a Python API for spawning different objects such as cars and pedestrians inside the virtual environment with different motion plans.

Figure 5 shows iseAuto facing a stopped Non-Player Character (NPC) vehicle that is spawned in front of the AV. Picture (a) is inside the SVL environment while picture (b) illustrates the lidar perception of the environment in RViz visualization tool. There is no filtering applied on this point-cloud; therefore, everything is mixed together and it is hard to distinguish objects for later processing. One of the challenging topics of self-driving development is overtaking. The way that the AV should decide for this mission and the risks that it faces are under study. Our experience with the vehicle trying to pass a stopped NPC or an object has led us to focus on this topic more. In this way, simulations can help first to improve our perception and detection system, and then to improve the mission and motion planning for a safe overtake. The first steps for detection are filtering and clustering the point-cloud. Autoware has some predefined features for them. One common point-cloud filtering is ground removal, in which some part of the point-cloud defined as ground will be separated. Each lidar point-cloud can be filtered separately or once after concatenation with other lidars. Filtering parameters have an intensive effect on the detection result. Sometimes losing 10 to 20 points due to the improper filtering will result in the object not to be detected.

Filtering and clustering are illustrated in Fig. 6. Filtering was applied to Fig. 5b. As a result, the ground, which can be seen in the figure, is almost removed from the point-cloud (see Fig. 6a). However, the NPC points remained and they were clustered as an object in Fig. 6b. Filtering accuracy results in high-performance object detection and safe decision making [21]. Figure 7 illustrates how different ground filtering parameters can change maximum distance for detecting a stopped NPC in front of the AV shuttle, although both cases have similar clustering parameters. Figure 7b shows that the NPC is

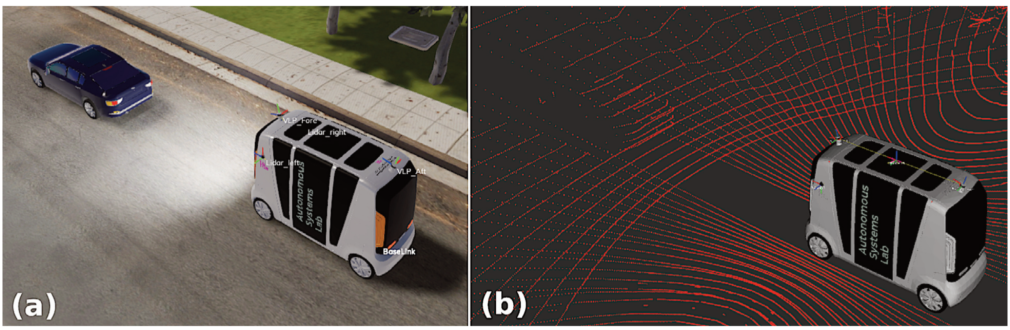

Fig. 5. (a) SVL environment versus (b) Rviz point-cloud visualization.

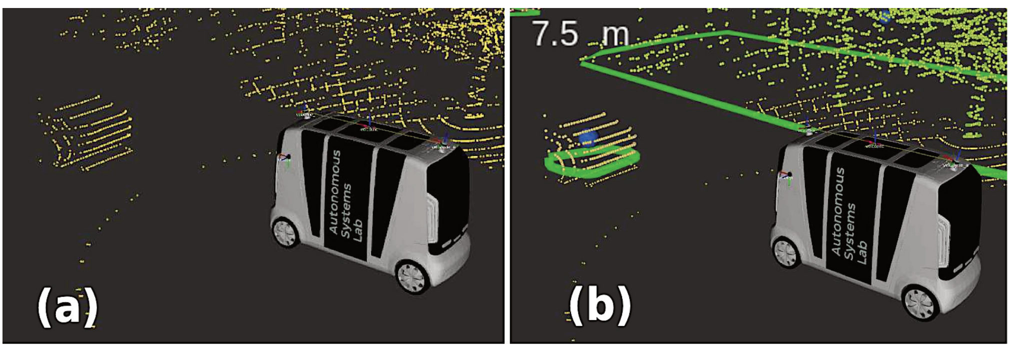

Fig. 6. (a) The ground filtering of the point-cloud and (b) applying of Euclidean clustering.

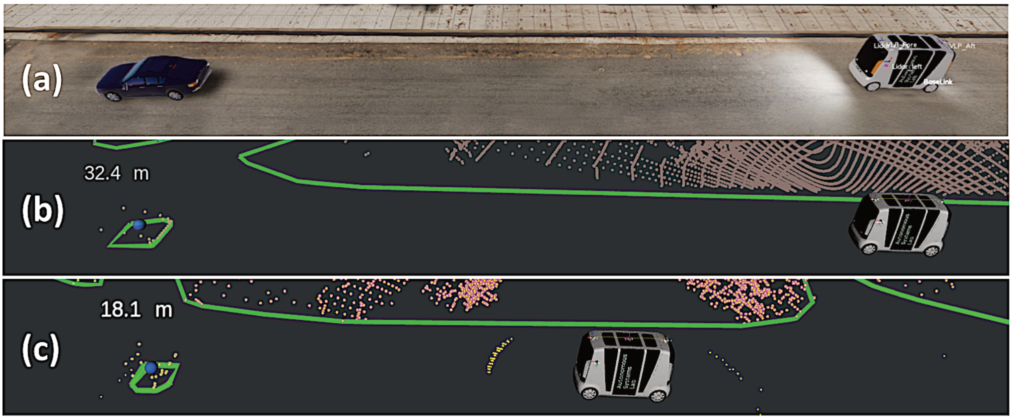

Fig. 7. Maximum distance for detecting a stopped NPC after filtration with different filtering parameters.

detected by the AV shuttle from the distance of 32 metres but picture (c) demonstrates that the maximum distance enabling to detect an object has decreased to 18 metres. The more distance we have for detection, the more time we have for making a smooth control decision. In AVs with multiple lidars, filtering accuracy can be improved by performing it before point-cloud concatenation.

4.1. Scenario definition

Scenarios are plans for studying simulations effectively. A good scenario generator can help to validate the whole control system faster in a more reliable way, guaranteeing to cover all the corner cases that might cause failure in the system. There are several methods for generating the scenarios such as human designed, grid search and optimized searching. For example, in [22], the authors implemented a learning method to find safety-critical scenarios for specific tasks. In this paper, for showing the simulation workflow, two main and simple overtaking scenarios were studied. Figure 8 demonstrates two different situations in overtaking: scenario A shows a stopped car that is overtaken by our shuttle while scenario B shows the same mission with an additional car, already starting to overtake the two others.

4.2. Running simulation

In this section, the two described scenarios are simulated inside the simulator and shuttle behaviour is monitored.

- Scenario A

In this scenario, the shuttle is passing a stopped vehicle by generating an alternative local waypoint. The overtaking algorithm is enabled after the shuttle has detected an obstacle in its path. Five different frames of this scenario simulation are shown in Fig. 9. First, the AV follows the way and detects the obstacle (step 1), then stops 15 metres before the object (step 2) and generates a new waypoint (step 3). Then, it starts to follow the new waypoint, and finally, after passing the obstacle, it changes the lane back to the initial path (step 4) and continues its former route (step 5).

By simulating scenario A several times in different areas, the overtaking algorithm for passing a static object was initially evaluated and verified. But to investigate more challenging situations, various road users such as other vehicles and pedestrians should be involved in the scenario. For this, another scenario was designed by adding another vehicle driving forward from behind in the opposite lane.

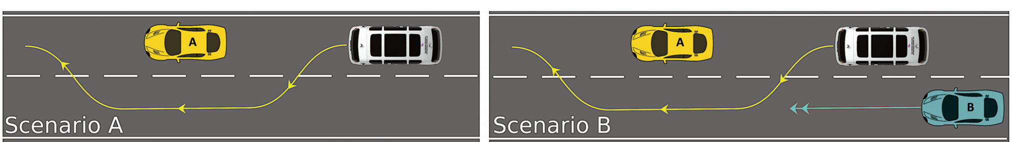

Fig. 8. Two different scenarios for overtaking.

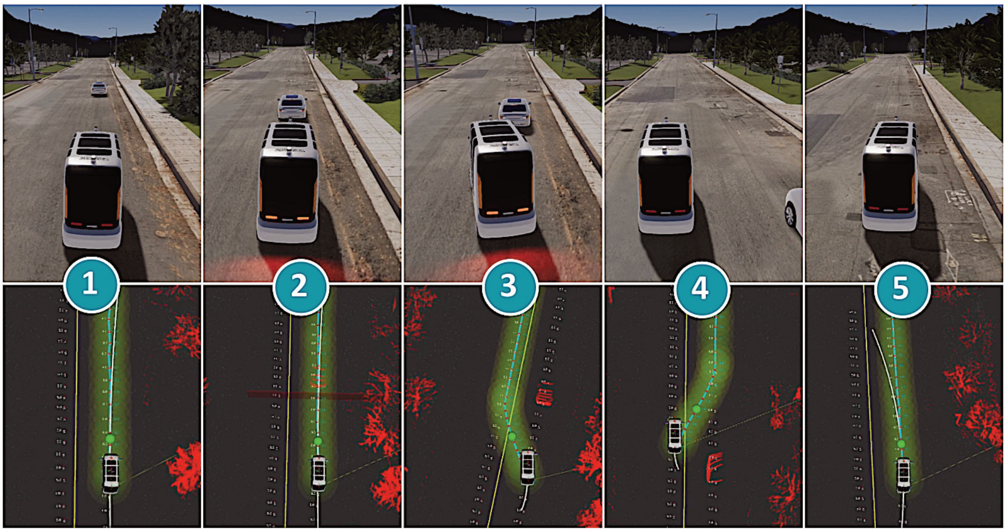

Fig. 9. Five different steps of the scenario A simulation in the SVL simulator (top) and in the Rviz (bottom).

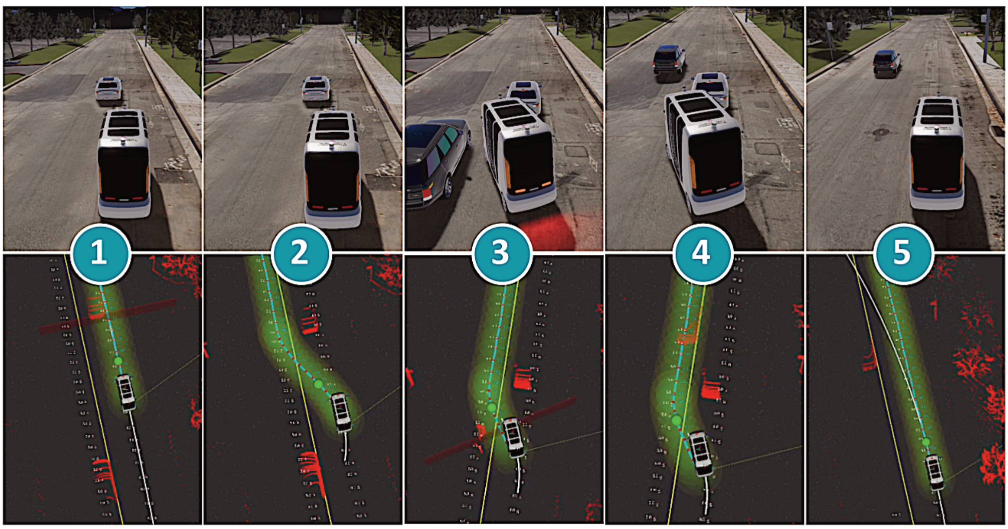

Fig. 10. Different steps of an overtaking process.

- Scenario B

Figure 8 shows the scenario B scheme, that a third vehicle is overtaking the shuttle and the stopped vehicle. It is expected that the shuttle prevents collision and considers the opposite lane traffic. Similar to the former scenario, five steps of scenario B are recorded in Fig. 10. As seen in the simulation, the AV reaches the static object and stops to prepare for overtaking (step 1). The moving vehicle is visible in the Rviz software (frame 1 image below) as a red point-cloud cluster. It is expected that the shuttle prevents collision and considers the opposite lane traffic while overtaking. In step 2 the shuttle starts to overtake and the new path is generated. Before the shuttle changes the lane, it meets the moving vehicle in the green area (collision area, any object inside it is an obstacle), then the shuttle stops before the collision happens. Finally, after the moving vehicle drives more than 15 metres along the green area, the shuttle starts to follow the route and changes the lane back to its initial path.

This scenario was simulated with a different value for variables such as the speed of the moving vehicle and the lateral position of each vehicle on the road. The results recorded collision in some cases and investigations showed that due to the limited size of the green area and lack of an efficient motion prediction while shifting lanes, the AV can collide with other road users that are not considered. Therefore, using the current overtaking algorithm without any added prediction feature is rejected and it is not safe to be implemented in the real shuttle.

5. CONCLUSIONS

Safety validation is crucial for most of the AV developments and deployments. The simulation as a validation approach presented in this paper offers a practical and effective way to evaluate the safety in different levels. This paper provides the simulation architecture of iseAuto with SIL testing, which shows how the virtual environment and vehicle model are used in combination with Autoware to simulate different scenarios. As an illustration, two overtaking scenarios were studied and the control algorithm was examined based on its safe performance. In conclusion, the development and utilization of this testing scheme will enable the development of safety improvement and autonomous vehicle performance.

ACKNOWLEDGEMENTS

This research received funding from two grants: the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No. 2014-2020.4.01.20-0289. The publication costs of this article were covered by the Estonian Academy of Sciences and Tallinn University of Technology.

REFERENCES

1. Kalra, N. and Paddock, S. M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract.*, 2016, **94**, 182–193.
2. Chao, Q., Jin, X., Huang, H. W., Foong, S., Yu, L. F. and Yeung, S. K. Force-based heterogeneous traffic simulation for autonomous vehicle testing. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 20–24, 2019*. IEEE, 8298–8304.
3. Aeberhard, M., Rauch, S., Bahram, M., Tanzmeister, G., Thomas, J., Pilat, Y., Homm, F., Huber, W. and Kaempchen, N. Experience, results and lessons learned from automated driving on Germany's highways. *IEEE Intell. Transp. Syst. Mag.*, 2015, **7**(1), 42–57.
4. Anderson, S. J., Peters, S. C., Pilutti, T. E. and Iagnemma, K. Design and development of an optimal-control-based framework for trajectory planning threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In *Robotics Research* (Pradalier, C., Siegwart, R. and Hirzinger, G., eds). Springer, Berlin, Heidelberg, 2011, 39–54.
5. Razdan, R., Lumina, J., Balachandran, A., Cheng, C., Sreenivas, S., Fernando, X., Taiber, J., Kalia, A., Keel, N., Zuby, D., Krishnan, K., Langer, D. and Sell, R. Unsettled technology areas in autonomous vehicle test and validation. SAE Technical Paper Series, 2019.
6. Malayjerdi, M., Kuts, V., Sell, R., Otto, T. and Baykara, B. C. Virtual simulations environment development for autonomous vehicles interaction. In *Proceedings of the ASME 2020 International Mechanical Engineering Congress and Exposition, Portland, OR, USA, November 16–19, 2020.* ASME, **2B**, IMECE2020-23362.
7. Kuts, V., Modoni, G. E., Otto, T., Sacco, M., Tähemaa, T., Bondarenko, Y. and Wang, R. Synchronizing physical factory and its digital twin through an IioT middleware: a case study. *Proc. Est. Acad. Sci.*, 2019, **68**(4), 364–370.
8. Kuts, V., Cherezova, N., Sarkans, M. and Otto, T. Digital Twin: industrial robot kinematic model integration to the virtual reality environment. *J. Mach. Eng.*, 2020, **20**(2), 53–64.
9. Lu, B., He, H., Yu, H., Wang, H., Li, G., Shi, M. and Cao, D. Hybrid path planning combining potential field with sigmoid curve for autonomous driving. *Sensors*, 2020, **20**(24), 7197.
10. Andersen, H., Schwarting, W., Naser, F., Eng, Y. H., Ang, M. H., Rus, D. and Alonso-Mora, J. Trajectory optimization for autonomous overtaking with visibility maximization. In *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, October 16–19, 2017*. IEEE, 1–8.
11. Easa, S. M. and Diachuk, M. Optimal speed plan for the overtaking of autonomous vehicles on two-lane highways. *Infrastructures*, 2020, **5**(5), 44.
12. Sell, R., Leier, M., Rassõlkin, A. and Ernits, J. Self-driving car ISEAUTO for research and education. In *Proceedings of the 2018 19th International Conference on Research and Education in Mechatronics (REM), Delft, Netherlands, June 7–8, 2018*. IEEE, 111–116.
13. Christophe, F., Sell, R., Bernard, A. and Coatanéa, E. Opas: Ontology processing for assisted synthesis of conceptual design solutions. In *Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering* Conference*, San Diego, USA, August 30–September 2, 2009*. ASME, **49026**, 249–260.

14. Sell, R., Coatanéa, E. and Christophe, F. Important aspects of early design in mechatronic. In *Proceedings of the 6th International Conference of DAAAM Baltic Industrial Engineering, Tallinn, Estonia, April 24–26, 2008*, 177–182.
15. Wang, R., Sell, R., Rassõlkin, A., Otto, T. and Malayjerdi, E. Intelligent functions development on autonomous electric vehicle platform. *J. Mach. Eng.*, 2020, **20**(2), 114–125.
16. Rassõlkin. A., Sell, R. and Leier, M. Development case study of the first estonian self-driving car, iseauto. *Electr. Control Commun. Eng.*, 2018, **14**, 81–88.
17. Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y. and Azumi, T. Autoware on board: Enabling autonomous vehicles with embedded systems. In *Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, April 11–13, 2018*. IEEE, 287–296.
18. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. and Koltun, V. Carla: An open urban driving simulator. 2017, *arXiv:1711.03938*.
19. Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S. et al. LGSVL simulator: A high fidelity simulator for autonomous driving. 2020, *arXiv:2005.03778*.
20. Medrano-Berumen, C., Malayjerdi, M., Akbaş, M. I., Sell, R. and Razdan, R. Development of a validation regime for an autonomous campus shuttle. In *Proceedings of the 2020 SoutheastCon, Raleigh, NC, USA, March 28–29, 2020*. IEEE, 1–8.
21. Habermann, D., Hata, A., Wolf, D. and Osório, F. S. 3d point clouds segmentation for autonomous ground vehicle. In *Proceedings of the 2013 III Brazilian Symposium on Computing Systems Engineering, Rio de Janeiro, Brazil, December 4–8, 2013*. IEEE, 143–148.
22. Ding, W., Chen, B., Xu, M. and Zhao, D. Learning to collide: An adaptive safety-critical scenarios generating method. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, October 25–29, 2020.* IEEE, 2243–2250.

Autonoomse sõiduki turvalisuse hindamise suure täpsusega simulatsiooni meetod

Mohsen Malayjerdi, Barış Cem Baykara, Raivo Sell ja Ehsan Malayjerdi

Autonoomsete sõidukite tööstus planeerib strateegilisi lahendusi, et kindlustada turvalisus enne, kui autonoomsed sõidukid viiakse masstootmisse. Turvalisuse saavutamiseks on vajalik läbi viia väga erinevaid teste. Kõikide testide tegemine reaalse sõidukiga reaalses linnaruumis on pigem ebapraktiline ja võtaks aega aastaid. Selle probleemi vältimiseks kasutatakse simulatsioone. Antud artikli eesmärgiks on välja pakkuda metoodika ja tehnoloogia turvalisuse valideerimise simulatsioonideks autonoomsete sõidukite testimisel. Artiklis on välja pakutud turvalisuse hindamise meetod, mis on realiseeritud TalTechi linnakus tegutseva TalTechi iseauto autonoomse sõiduki platvormil. On loodud virtuaalne mudel linnaku testalast, mis sisaldab eri objekte ja mis on konverteeritud 3D-kaardiks Unity keskkonnas. Loodud virtuaalne mudel on omakorda sisendiks SVL-simulaatorile, mis ühendab endas virtuaalsete andurite simulatsiooni ning Autoware algoritmid, mis juhivad TalTechi iseautot. Demonstratsioonlahendusena on kirjeldatud simulatsioonijuhtu, kui isejuhtiv sõiduk peab tegema möödasõidu seisvast autost, mis blokeerib sõidurea. Lõpuks on näidatud, kuidas antud lahendus võimaldab hinnata isejuhtiva sõiduki otsuste tegemise võimekust ja turvalisust eri situatsioonides.

Appendix 4

IV

Raivo Sell, Ehsan Malayjerdi, Mohsen Malayjerdi, and Baris Cem Baykara. Safety toolkit for automated vehicle shuttle-practical implementation of digital twin. In 2022 International Conference on Connected Vehicle and Expo (ICCVE):1–6. IEEE, 2022

This full text paper was peer-reviewed at the direction of IEEE Instrumentation and Measurement Society prior to the acceptance and publication.

Safety Toolkit for Automated Vehicle Shuttle - Practical Implementation of Digital Twin

Raivo Sell
FinEst Centre for Smart Cities, Dept. of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
raivo.sell@taltech.ee

Ehsan Malayjerdi
FinEst Centre for Smart Cities, Dept. of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
ehsan.malayjerdi@taltech.ee

Mohsen Malayjerdi
FinEst Centre for Smart Cities, Dept. of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
momala@taltech.ee

Baris Cem Baykara
FinEst Centre for Smart Cities, Dept. of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
baris.baykara@taltech.ee

2022 International Conference on Connected Vehicle and Expo (ICCVE) | 978-1-6654-1687-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCVE52871.2022.9742881

***Abstract*—Safety of autonomous driving and automated vehicles is one of the most important concerns to bring self-driving vehicles on the streets. This paper deals with important aspects of safety evaluation and validation steps in order to provide a full set of tools and procedures to safety evaluations. For effective evaluation, simulations are extremely important procedures which rely on the digital model of the environment and vehicles. This paper demonstrates the process of creating digital twins for safety evaluations.**

***Index Terms*—autonomous driving, safety validation, digital twin, smart city**

I. Introduction

Autonomous driving is one of the key technologies today to reshape the transport sector to the new paradigm and contributes significantly to carbon emission reductions. All major car manufacturers like GM, Ford, Tesla, etc as well as technology giants like Alphabet, Apple are investing huge amounts of money into automated vehicle developments. The shift from conventional fossil fuel-based human-driven vehicles has already started and will accelerate constantly. Although experiencing some optimistic promises by OEM and other players, it is expected that for example in China alone as having a potential becoming the largest AV market in the world, 66% of the passenger-kilometers traveled in 2040 are by autonomous vehicles [1]. However, there are several challenges and obstacles slowing down the development and deployment. Safety is one of the most critical aspects and it can not be neglected throughout the whole development process. Even technically most important, ordinary users and most of the stakeholders are emphasizing safety as the most important concern of autonomous vehicles deployment. In recent years, different surveys all over the world [2]–[6] represent people's high level of concern for safety and security. On average, more than 70% of respondents were highly concerned about safety issues. Therefore, the AVs technology will be successful when it gains social acceptance by providing safety.

The industry and academia are focusing more and more on safety aspects resulting in developing standards and the procedures for safety evaluation, validation, and verification. MIT Technology Review [7] Autonomous driving: Safety first, conducted in corporations with Intel emphasizes that safety is of paramount importance for operating autonomous vehicles. This includes in addition to passengers and drivers aslo pedestrians, road, micro mobility traffic etc. Thus safety issues must be resolved to the full satisfaction of the public. However there are many unsettled topics concerning automated driving and in particular verification and validation aspects which need to be addressed. SAE Edge research reports are covering some of these in detail [8]–[10].

The research and the safety evaluation toolkit as a tool for evaluation the use cases are in the focus of this paper where the first step of the procedure is to create a digital twin of the real use case environment. The digital twin is a model of the real environment and is used for simulations as well as other testing experiments. The following work is the advancement of simulations case studies [11], [12], autonomous driving experiments [13], and digital twin developments [14] in TalTech Autonomous Vehicles research group and in cooperation with Florida Polytechnic University, Advanced Mobility Institute.

II. Safety Validation Platform

Currently, there are three main ways of testing autonomous cars including simulation, track testing, and real road testing [15]. All studies have shown that the simulation is totally safe, cheaper, much faster, and reproducible [16] compared to the other ways of testing. In this research, we present a safety evaluation platform based on simulation testing. There are several open-source and proprietary simulators specially designed for AVs. These simulators include low-fidelity

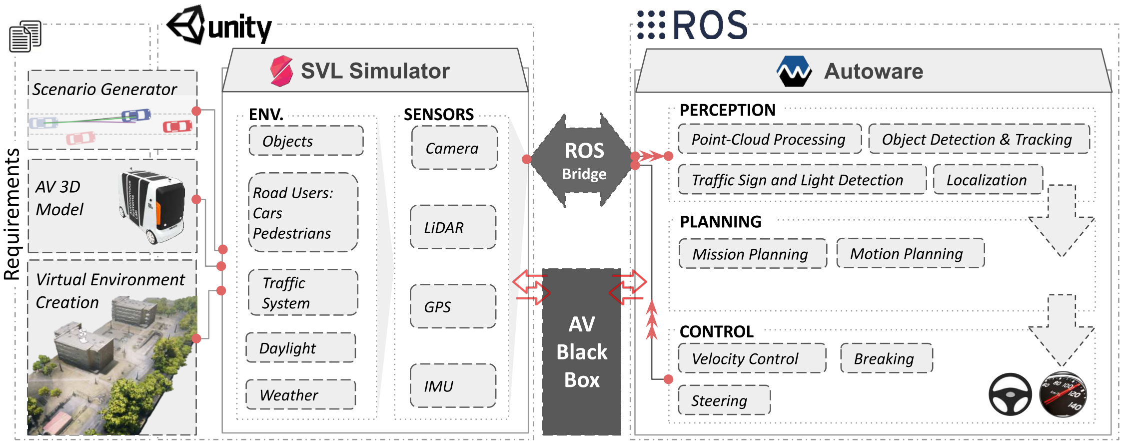

Fig. 1. Safety validation workflow

ones, which run simulations in low details, and high-fidelity ones provided higher details in simulations, that are utilized based on the user's requirements. Game-engine simulators like CARLA [17] and SVL [18], which are run based on Unreal and Unity engines respectively, employ a powerful physics engine and provide end-to-end simulations. To achieve accurate and reliable results, SVL, a high-fidelity Unity-based simulator, was chosen. This simulator enables us to employ the AV software running on the vehicle to control the ego in simulations. Hence, Autoware [19], the ROS-based open-source autonomous driving stack, was connected to the simulator through a ROS bridge. Fig. 1 shows a summary workflow of the validation platform, including requirements, SVL simulator, Autoware stack, and how they are related to each other. There are three main requirements for starting a simulation. First, each simulation needs a plan or scenario for execution. In this step, we build different conditions that jeopardize ego safety. Second, the ego corresponding 3D model is needed to define realistic sensor configuration and provide similar dynamics and dimensions.

Last, to simulate real-world situations, we need a copy of that environment in the virtual. This idea of creating a perfect virtual copy of the ego and its working environment inside simulations is called a digital twin that we will discuss in the next section. After supplying these requirements, the simulator runs predefined scenarios while providing sensor information for perception algorithms, and in return, receiving the control command from the Autoware motion controller. Also, a monitoring block is recording all the issued commands and behavior of the ego during each scenario for later study on that scenario. This block acts as a black box as shown in the figure.

III. Digital Twin

This idea, fittingly, is a digital representation of a physical object or service. Digital twin uses real-world data to create a simulation and predict how an object performs. A digital twin of the ego or the working environment will enable developers to push the limits of AV technology by creating a virtual test platform to evaluate them quicker and safer. We created digital twins of the ego and working environment to minimize the testing consequences. In the following, we will explain each separately.

A. *Virtual Terrain Generation*

For simulations, it is crucial to mimic real events and conditions in order to generate, test and evaluate reliable data sets. Most of these conditions are provided by industry leading game engines such as Unity engine. These engines can be used to create digital twin models and representations of the physical world elements.

The most important part of a graphical simulation is the 3D map of the area where an autonomous vehicle is to drive. Digital twins of geographical places can be created in various ways. However, an effective way to generate such representation is the photogrammetric approach. This approach takes a data set of images obtained by an RGB camera as the input and delivers realistic terrain objects in the Unity engine as the output. To obtain such a dataset, aerial images are needed to be obtained of the area to be mapped in different angles and orientations. Agisoft Metashape, a photogrammetry software, then takes this dataset and generates a colored dense point cloud. The point cloud is then segmented out by another software called, CloudCompare. CloudCompare allows for point cloud manipulations. The segmentation process separates the ground points from the off-ground points. Next, each point is then classified based on LAS 1.4 specifications [20]. Once the dense point cloud is segmented and classified, an in house unity plugin takes it and creates a Unity terrain object with the specified classification tag. The terrain is created only from the ground points. For vegetation and other objects, a grayscale mask image is generated so that pre modeled 3D objects (i.e. trees, grass, rocks and such) can be spawned on the correct spots in the engine. After all the objects are spawned and configured, SVL simulator then can import, build and publish the terrain on the SVL maps section.

B. Virtual Shuttle Model

In order to create the ego virtual model, we followed the simulator instruction that required Unity to build the model and prepare it for the simulations. The STL model of the ego body and other components were imported to Blender, an open-source graphics software toolset, for assembly and preparation for the Unity process. Then, the Blender file was imported to Unity and was built based on the simulator library. Next, we uploaded the output file to the simulator server to use as a new vehicle. We carried out the process of sensor configuration (see Fig. 2) in the SVL server relative to the base coordinates defined in Unity. Sensors position and orientation were determined by the actual dimension in the real vehicle. Fig. 2 demonstrates different lidar sensors employed by the ego. After these steps, the ego was ready to use in the simulation.

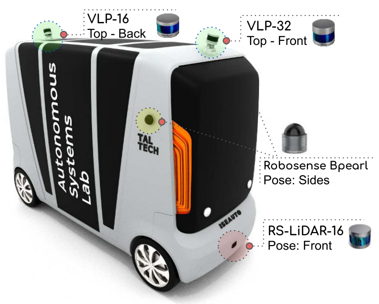

Fig. 2. Ego digital twin; 3D model with the sensor configuration

IV. Validation Scenarios

We introduced that scenarios are one of the main requirements for the validation platform. Due to the extensive possible conditions that we can simulate and run the ego in them, it is required to have a good strategy and plan to target edge-case scenarios and address the issues faster. There are several ways to generate scenarios including random fuzzing techniques [21], probabilistic programming languages like Scenic [22], and human-defined schemes. Although human-defined scenarios do the job for the control algorithm development, generally, it is necessary to utilize a comprehensive method for finding bugs and safety issues.

A simple scenario for overtaking maneuver is illustrated in Fig. 3. The first picture shows the initial starting point for each actor, and the second one shows the moment while the ego decides to overtake. For instance, a fuzzer can generate a variety of cases by changing parameters value such as relative distances and speeds of actors in the scenario. That leads to finding corner cases in which the ego vulnerabilities show up.

After determining the scenario, it should be written in the simulator scenario format, a JSON file, or a Python script.

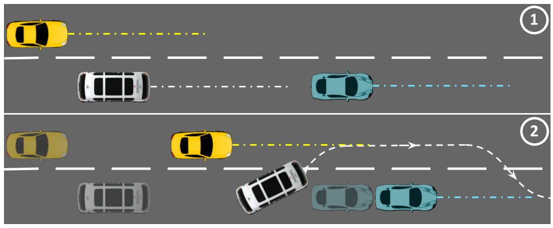

Fig. 3. A scenario defined to evaluate the overtaking safety. (1) indicates the initial start and (2) while overtaking

Afterward, we run it in the simulator to monitor the ego decisions. Fig. 4 depicts two steps of the scenario, and as can be seen, the ego did not start overtaking while the next lane was busy by the car coming from behind.

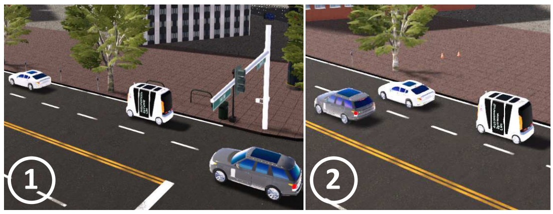

Fig. 4. The scenario simulation in SVL. (1) indicates the initial start and (2) while overtaking

Overall, scenario generation is one of the key factors of having high-efficiency simulation. Employing targeted scenarios enables us to find edge-cases quickly that cause failure. It also helps us to develop our new ideas on different components of the control software without any hesitation and worries. Next, we will present how software-in-the-loop (SiL) simulations were carried out on the platform.

V. Software In The Loop Simulations

Autoware is one of the powerful open-source autonomous software that works on the ROS platform. Many researchers and companies have focused on Autoware because of its open-source nature, ROS-based, and high flexibility. Autoware consists of different parts including perception, planning, and control, each of which has several components (see Fig. 1). The simulator will provide the virtual sensor data for the perception algorithms. Then, the planning algorithms make decisions by interpreting the data processed in the perception, and finally, the control algorithms issue commands.

Localization is one of the perception components that define the position of the ego inside the map. This process which employs ndt-matching algorithms needs a reference map of the working environment. In the following, we will discuss it in detail, then present a running simulation:

A. A Valid Map for localization

1) 3D point cloud map: In autonomous vehicles, point cloud maps are used in a variety of applications such as localization and path planning. The point cloud map can be aligned with sensor information acquired in real time to determine the vehicle's current location and can be combined with information that recognizes other vehicles to generate the ego vehicle's route.

LiDAR, a critical sensor for self-driving cars, has the two characteristics: First of all, it can provide precise distance information of structure with point form. Based on its high resolution, Second the LiDAR sensor can represent the vehicle's surroundings in detail. A point cloud map is a 3D map that reconstructs the environment based on the characteristics of the LiDAR sensor. A point cloud map can be used in a variety of applications (such as localization and path planning) in an autonomous vehicle.

3D maps allow self-driving cars to locate themselves in their surroundings. To localize using a map and Lidar data, the point cloud from the sensor must be associated with the point cloud from the map. In autonomous vehicles, this is referred to as scan matching. Iterative Closest Point, which uses 6 degrees of freedom to find the closest point to the geometric entity from a given 3D point cloud, is a common way to do this.

In real-world scenarios, our points will most likely be a little off the map. Measurement errors will result in slightly misaligned points, and the world may change slightly between when we record the map and when we make our new scan. These minor errors are addressed by Normal Distribution Transform NDT matching. Rather than attempting to match points from our current scan to points on the map, we attempt to match points from our current scan to a grid of probability functions generated from the map. Following is the two task performed:

NDT mapping (Map generation) [23] includes converting the LiDAR point cloud into a piecewise continuous and differentiable probability density (NDT). The probability density is made up of a set of normal distributions, with each point in the point cloud assigned to a voxel. A voxel is a three-dimensional lattice cube to which points are assigned based on their coordinate value. The point cloud is divided into k ND voxel clouds and combined, and the voxel grid filter is used to reduce the computation cost and noise from the 3D map.

For creating the map the velodyne VLP-32 Lidar is used to record the point cloud data. The lidar is connected on top of the vehicle and drives once in the desire with the 5-7 km/h to save the Point Clouds. In the next step, this data is used to create a 3D point cloud map (see Fig. 5).

2) Lane-level map: The majority of self-driving solutions rely on high-definition maps (HD maps), which are specialized lane-level maps with extremely high locational accuracy. Big mapping companies use mobile mapping cars (specially equipped vehicles with sensors for map data collection) to collect data for creating HD maps. Along with the necessary data processing, creating and maintaining HD maps in a changing world is very expensive. The availability of HD maps would significantly lower the bar for widespread adoption of autonomous driving.

Fig. 5. A Point cloud map is created based on Lidar data.

HD maps are lane-level maps with extremely high locational accuracy that provide a lot of information about road geometry, various traffic regulating elements, and road surroundings.

These can be incorporated into the HD map localization model. It means that maps and their features could be used as a reference to help the self-driving vehicle locate itself using its perception [24]. To fully utilize HD maps and navigate complicated urban traffic, autonomous driving requires very precise localization (a few cm level accuracy is required).

Based on the semantic information contained in HD maps, HD maps can be used to extend vision beyond the normal sensory range, and concepts such as electronic horizon [25] are introduced and included in behavior planning.

HD maps can be used if a vehicle can be localized relative to a map, which requires sensors. Sensors are also required because AD usually requires a precise and robust perception of the environment.

According to [26], HD map content is divided into three models, each of which is consistently geo-referenced:

- Road model - used for broad strategic planning, such as navigation.
- Lane model - used for perception and tactical planning (guidance), includes detailed and feature-rich lane level data.
- Localization model - aids in the mapping of self-driving vehicles.

In general, the goal of a road model is to generate a plan from end to end. The topological structure is a priority, and centimeter-level accuracy is not that much of an issue. There are four HD map formats:

- Autoware vector maps
- OpenDrive
- Navigation Data Standard
- Lanelet2

Lanelet2 [27] is a C++ library to create a map for autonomous driving (see Fig. 6). This XML base OSM data format is used for path planning. To describe all the map data, the Lanelet has six different main Items (points, linestrings, polygons, lanelets, areas, and regulatory elements). Separate from map data (format) there is also a software framework and API that enable the use of different traffic rules for different user types. It can be different for cars, cyclists, and emergency vehicles resulting in totally different routing schemes (for example allowing emergency vehicles to drive in the opposite lane). There are free editors available, like JOSM [27], [28].

The periods of OSM only being for mapping enthusiasts are over. Corporations are making an increasing proportion of edits, and their primary focus has been on road networks, whereas non-corporate mappers are more interested in editing buildings and points of interest [29]. A corporate data-team member edited nearly 17 percent of the global road network (measured per kilometer) as of March 2020.

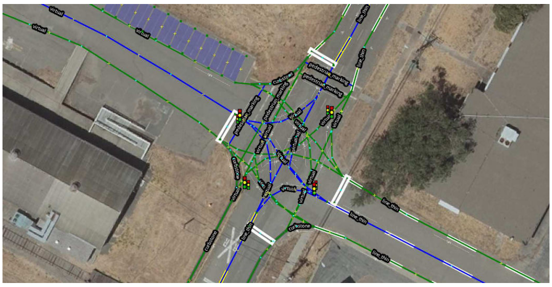

Fig. 6. Lanelet2 Map

When the Lanelet map is created by JOSM, then the map is tested before it is used in the real world. The testing process includes the following steps:

- loading the map on the Pointcloud map to match the Lanelet with the pointcloud map Fig 7.
- testing the routing, traffic lights and other road elements on the map.
- testing with Openplanner simulator [30].

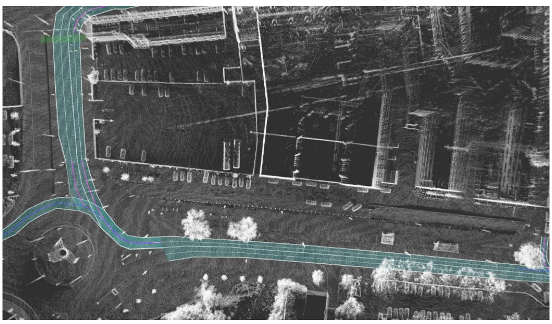

Fig. 7. Align the Lanelet map with point cloud map

3) Running Simulations: After providing all simulation requirements, we run each scenario simulation and evaluate the ego software performance by different criteria such as collision existence, ride comfort level, path deviation, etc. Hence, all necessary data will be recorded for later analysis. For instance, Fig. 8 demonstrates three screenshots captured at different times in a SiL simulation in the same scenario described before. Top images indicate the SVL simulator screens, and the below ones are from RViz visualization software. RViz is a ROS-based visualization software that displays different data such as sensor point cloud data, waypoints, the map, and the ego position.

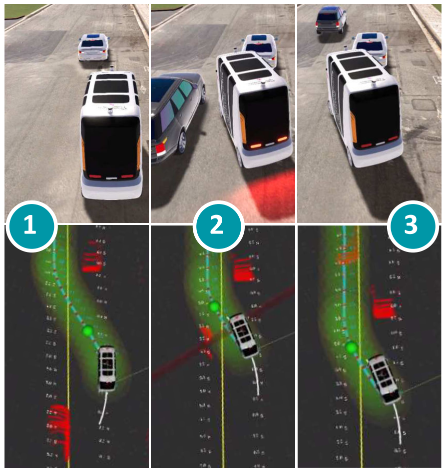

Fig. 8. A scenario was simulated while Autoware was controlling the ego

Fig. 8 (1) is the initial point of the scenario that the ego wants to overtake the front car while a car is reaching from the side lane. The second image shows that the ego stopped and didn't continue the overtaking maneuver because the side lane vehicle almost reached the ego waypoint. Finally, after the car passed the ego, it started moving along the waypoint. We should run the scenario with various relative distances and speeds for the players in the scene in order to cover all the possible cases and validate the results. This can be achieved by taking the advantage of fuzzing techniques to generate a variety of scenarios. Then we can improve the efficiency of finding the safety violations by implementing an optimization technique to discover highly potential safety-hazardous situations.

VI. Physical Validation In Smart City Context

As the digital version of the real world is a good tool for simulation and running a huge number of scenarios and combinations in a cost effective way, it will never fully replace real-life testing and validation. Moreover the simulation process itself must be validated and verified to be as close as possible to its real-world counterpart. Unfortunately, this is often very costly and hard to implement as it requires significant state-of-art technology as well as infrastructure and an environment suitable for experimentation. In this work, all mentioned conditions are fulfilled and the digital twin can be validated in the real smart city environment with the real SEA L4 driverless vehicle. The vehicle used for the validation is based on open-source autonomous driving stack Autoware [19] and is developed in the AV lab as an experimental prototype AV shuttle - TalTech iseAuto [31], [32]. Validation experiments are conducted on the university campus semi-open road as a AV test track equipped with smart bus stops, smart pedestrian crossing and 5G mobile network.

VII. Conclusion

Safety is the key concern in autonomous driving technology and the deployment of vehicles to real traffic. In this paper a short overview was given, how to evaluate the safety, detect edge cases, and create the environment for the safety evaluation process. The main focus was digital twin creation as a base virtual environment for safety simulations. Digital twin of the AV solution set-up is crucial in order to get reliable results out of the simulations. All together different tools and processes described in this paper forms a Safety toolkit for local governments or other stakeholders interested to deploy AV shuttles on their streets or dedicated areas. For example the toolkit can be applied to Florida Suntrax in order to create a full digital twin out of it and in that way enable customers or research institutions to run their solution virtually at first place and then enter real testing.

Acknowledgment

This research has received funding from two grants: the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No 2014-2020.4.01.20-0289.

References

[1] T. Möller, A. Padhi, D. Pinner, and A. Tschiesner, "The future of mobility is at our doorstep," *McKinsey Center for Future Mobility*, 2019.

[2] A. Rezaei and B. Caulfield, "Examining public acceptance of autonomous mobility," *Travel behaviour and society*, vol. 21, pp. 235–246, 2020.

[3] S. Greaves, B. Smith, T. Arnold, D. Olaru, and A. T. Collins, "Autonomous vehicles down under: an empirical investigation of consumer sentiment," 2018.

[4] M. Kyriakidis, R. Happee, and J. C. de Winter, "Public opinion on automated driving: Results of an international questionnaire among 5000 respondents," *Transportation research part F: traffic psychology and behaviour*, vol. 32, pp. 127–140, 2015.

[5] B. Schoettle and M. Sivak, "Motorists' preferences for different levels of vehicle automation," University of Michigan, Ann Arbor, Transportation Research Institute, Tech. Rep., 2015.

[6] B. Schoettle and M. Sivak, "A survey of public opinion about autonomous and self-driving vehicles in the us, the uk, and australia," University of Michigan, Ann Arbor, Transportation Research Institute, Tech. Rep., 2014.

[7] Intel et al., "Safety first for automated driving," Tech. Rep., 2019.

[8] R. Razdan, "Unsettled technology areas in autonomous vehicle test and validation," SAE Technical Paper, Tech. Rep., 2019.

[9] J. Taiber and R. Sell, "Unsettled impacts of integrating automated electric vehicles into a mobility-as-a-service ecosystem and effects on traditional transportation and ownership," SAE Technical Paper, Tech. Rep., 2019.

[10] R. Razdan, "Unsettled topics concerning autonomous public transportation systems," SAE Technical Paper, Tech. Rep., 2020.

[11] M. Malayjerdi, B. C. Baykara, R. Sell, and E. Malayjerdi, "Autonomous vehicle safety evaluation through a high-fidelity simulation approach," *Proceedings Of The Estonian Academy Of Sciences*, p. 413–421, 2021.

[12] M. Malayjerdi, V. Kuts, R. Sell, T. Otto, and B. C. Baykara, "Virtual simulations environment development for autonomous vehicles interaction," in *ASME International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2020.

[13] R. Wang, R. Sell, A. Rassolkin, T. Otto, and E. Malayjerdi, "Intelligent functions development on autonomous electric vehicle platform," *Journal of Machine Engineering*, vol. 20, 2020.

[14] V. Kuts, T. Otto, Y. Bondarenko, and F. Yu, "Digital twin: Collaborative virtual reality environment for multi-purpose industrial applications," in *ASME International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2020.

[15] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[16] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," *arXiv preprint arXiv:2101.05337*, 2021.

[17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[18] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[19] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 287–296.

[20] American Society for Photogrammetry & Remote Sensing, "Las specification 1.4-r15," 2019.

[21] J. C. Han and Z. Q. Zhou, "Metamorphic fuzz testing of autonomous vehicles," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 2020, pp. 380–385.

[22] D. Fremont, X. Yue, T. Dreossi, S. Ghosh, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: Language-based scene generation," *arXiv preprint arXiv:1809.09310*, 2018.

[23] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, 2003, pp. 2743–2748 vol.3.

[24] W. Kühn, M. Müller, and T. Höppner, "Road data as prior knowledge for highly automated driving," *Transportation research procedia*, vol. 27, pp. 222–229, 2017.

[25] C. Ress, A. Etemad, D. Kuck, and J. Requejo, "Electronic horizon-providing digital map data for adas applications," in *Proceedings of the 2nd International Workshop on Intelligent Vehicle Control Systems (IVCS)*, 2008, pp. 40–49.

[26] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," *The Journal of Navigation*, vol. 73, no. 2, pp. 324–341, 2020.

[27] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1672–1679.

[28] "Josm is an extensible editor for openstreetmap," https://josm.openstreetmap.de/, accessed: 2022-01-12.

[29] J. Anderson, D. Sarkar, and L. Palen, "Corporate editors in the evolving landscape of openstreetmap," *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, p. 232, 2019.

[30] H. Darweesh, E. Takeuchi, K. Takeda, Y. Ninomiya, A. Sujiwo, L. Y. Morales, N. Akai, T. Tomizawa, and S. Kato, "Open source integrated planner for autonomous navigation in highly dynamic environments," *Journal of Robotics and Mechatronics*, vol. 29, no. 4, pp. 668–684, 2017.

[31] R. Sell, A. Rassõlkin, R. Wang, and T. Otto, "Integration of autonomous vehicles and industry 4.0." *Proceedings of the Estonian Academy of Sciences*, vol. 68, no. 4, 2019.

[32] R. Sell, M. Leier, A. Rassõlkin, and J.-P. Ernits, "Self-driving car iseauto for research and education," in *2018 19th International Conference on Research and Education in Mechatronics (REM)*. IEEE, 2018, pp. 111–116.

Appendix 5

V

Ehsan Malayjerdi, Raivo Sell, Mohsen Malayjerdi, Andres Udal, and Mauro Bellone. Practical path planning techniques in overtaking for autonomous shuttles. Journal of Field Robotics, 39(4):410–425, 2022

Received: 21 September 2021 | Revised: 23 November 2021 | Accepted: 24 December 2021

DOI: 10.1002/rob.22057

RESEARCH ARTICLE

WILEY

Practical path planning techniques in overtaking for autonomous shuttles

Ehsan Malayjerdi[1] | Raivo Sell[1] | Mohsen Malayjerdi[1] | Andres Udal[2] | Mauro Bellone[3]

[1]Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Tallinn, Estonia

[2]Department of Information Technology, Tallinn University of Technology, Tallinn, Estonia

[3]Smart City Center of Excellence, Tallinn University of Technology, Tallinn, Estonia

Correspondence
Ehsan Malayjerdi, Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Ehitajate tee 5, Tallinn 19086, Estonia.
Email: ehsan.malayjerdi@taltech.ee and ehsan.malayjerdi@ttu.ee

Funding information
European Union's Horizon 2020 Research and Innovation Programme, Grant/Award Number: 856602; The European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, Grant/Award Number: 2014-2020.4.01.20-0289

Abstract

This paper proposes a reliable optimized sigmoid-based path planning algorithm that ensures smooth, fast and safe overtaking maneuver, while maintaining the necessary safety distance. In the proposed method, the desired smoothness of trajectories, the changes in steering angle and the lateral acceleration are controlled in a robust way. This paper describes the simulations, and the confirming real-world experiments, conducted using the autonomous shuttle iseAuto. Our results suggest that the sigmoid A-star algorithm leads to a smoother and more reliable motion when compared to other two standard methods. Specifically, the abruptness of necessary steering angle changes is reduced by factor of 4, and approaching the level of an experienced driver-like maneuver.

KEYWORDS

automated vehicle, optimization, path planning, trajectory evaluation

1 | INTRODUCTION

Self-driving vehicles and especially autonomous minibuses, also referred to as autonomous shuttles, are approaching exploitation in several cities worldwide. However, most of the autonomous shuttle projects are still in the test stage, and they can only demonstrate limited autonomous functionalities. One of the basic maneuver is to avoid collisions, for instance with a parked vehicle, by changing the driving lane. Although it may look a simple task to humans, most of the commercially available automation level 4 autonomous shuttles face difficulties in performing this task. Both research and manufacturers aim to bring the necessary improvements to vehicle safety, traffic accident rates, and vehicles' environmental impact. The first generation of autonomous shuttles could drive only in structured environments with simple path planning algorithms. However, new self-driving vehicles in automation level 4 and 5 are expected to drive close to human drivers' basic skill (Committee et al., 2014), meaning that the self-driving algorithm must support more complex operations like lane-change, overtaking, etc.

In Hegeman et al. (2005) the authors argue that up to 10% of driving accidents are related to lane change events. It is clear that the lane change is not a safe maneuver even for human drivers, and lane changing to perform a safe overtaking is a tremendous challenge for automated shuttles. Overtaking algorithms require comprehensive information about the surrounding environment in all directions, and complex calculations of static and dynamic objects in the scene (Milanés et al., 2012). Furthermore, information about other problematic factors such as different weather conditions, various traffic

[Correction added on 20 January 2022, after first online publication: The original versions of Figures 15, 17, and 21 were from a prior version of the paper, and have now been updated to the final versions.]

situations, interaction with other road users (cars and pedestrians), and different road quality (C. Li, Wang, et al., 2015) are also required for the planning algorithm. As a result, path planning is considered as an essential subtask of the automated shuttles' software (Carroll et al., 1992; Chae & Yi, 2020; Majidi et al., 2017). It is typically divided into global and local path planners according to the planning scope (Lu et al., 2020). By considering the entire environment from the start point to the target point, global planners are primarily concentrated on generating a path that minimizes time and distance to reach the target. Local path planners, unlike the global ones, focus on improving driving safety, using sensory data and vehicle stability information, in the obstacle avoidance process by taking into account different constraints during the navigation.

With the final goal to improve urban maneuvers of autonomous shuttles, this paper describes a new approach for overtaking based on smooth sigmoid curves using a two-phase overtaking maneuver. The strategy stands on creating an optimized sigmoid function according to the shuttle kinematic model for fast, smooth and safe generation of overtaking paths based on perception, optimal low-level steering, and trajectory planning parameters. Figure 1 shows the two-phase of the overtaking maneuver: (I) lane change and obstacle passing, (II) return to the original lane and continuation. Furthermore, this paper introduces a high-fidelity simulation test bed to verify and validate algorithms performance before implementing on the real shuttle. An extensive experimental campaign was carried out using our automated shuttle—iseAuto (Rassõlkin et al., 2018; Sell et al., 2018).

In summary, the main contributions of our work are: (1) A simple fast overtaking method able to take overtaking decisions and path planning within seconds; (2) A safe path planning approach that clearly shows the safe generated path using a verification procedure; (3) An improved human–machine interface by communicating the intention to overtake; (4) A safe overtaking method by generating a traffic-law compliant path that is verified in simulation and validated via implementation on a real automated shuttle demonstrating that the proposed method is safe and reliable.

The remainder of the paper is divided into eight sections: Section 2 presents related works on path planning algorithms that stand on Hybrid A-star, fuzzy logic, sigmoid functions or combination of different methods. Section 3 describes the kinematic model used for automated shuttles. Section 4 introduces our strategy for generating the sigmoid-based paths. Then the paper continues describing the simulation implemented in MatLab and the SVL environment. Our testbed vehicle is described in Section 6, including the sensor setup and the software

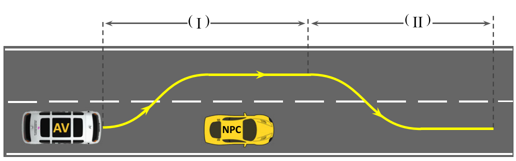

FIGURE 1 The overtaking maneuver by autonomous vehicle (AV) containing of the two lane change phases

architecture. Section 7 proposes the experimental results on our test-site performed using different case study, including a comparison with other existing strategies, discussing their main limitations and our improved strategy. Relevant conclusions are drawn in Section 8.

2 | RELATED WORK

In self-driving vehicles, overtaking trajectories are computed in planning modules by decision-making algorithms. Different types of decision-making algorithms are available in the literature, such as binary decision diagrams (Claussmann et al., 2015), learning-based technologies (Liu et al., 2019, 2020; Mo et al., 2021) model predictive control (MPC), and nonlinear MPC (Palatti et al., 2021; Viana et al., 2019).

Planning modules are divided into path planning and trajectory planning. Path planning algorithms generate safe paths for obstacle avoidance based on vehicle dynamic (Wang et al., 2019), which represents an interesting topic for research in the field of self-driving vehicles. A wide range of algorithms have been used in related research including artificial potential fields method (APF) (Feng et al., 2021; Y. Huang, Ding, et al., 2019; Shufeng & Junxin, 2018; Wahid et al., 2020; Xie et al., 2021).

In Naranjo et al. (2008), a fuzzy logic controller is proposed for the lane change process. In this method, the fuzzy controller reacts as a driver behavior during the overtaking operation. Although the proposed method shows acceptable results in their three experiments, the information for navigation is fully based on GPS (Global Positioning System) data, thus resulting in possible lack of performance in case of signal loss of accuracy and reliability.

On a different line of research, the A-star as a graph search algorithm, and its improved variants, are widely studied and implemented (Dolgov et al., 2010; Montemerlo et al., 2008). However, the classical A-star algorithm has some limitation, such as path planning challenges on intersections (Erke et al., 2020), Shinpei, 2017), and computational time (Duchoň et al., 2014). In this paper, some of these limitations are described and addressed in detail through some experiments. In Dixit et al. (2018), Lattarulo et al. (2017), the Bezier curves were utilized for the three-phase overtaking process through simulation. The proposed method has shown robust and smooth overtaking performance. However, the simulation was not reliable enough, and the method lacks of safety verification.

The method proposed in Chae and Yi (2020), Majidi et al. (2017) is an optimized global path planning algorithm with two-step optimization. Two cost functions were defined; first, minimizing the distance between the automated vehicle and the front vehicle at the starting point. Second, to minimize the sum of the automated vehicle lateral error from the reference path, and the steering velocity during the operation. Although the generated path was optimized, this study was carried out without any experiment and just based on a dynamic model in the CarSim simulator that is not realistic enough as a verification testbed.

In the filed of path planning for autonomous driving, many research studies use mathematical functions such as quintic polynomial curves (Zhu et al., 2018), polynominal function (Chen & Huang, 2018), Clothoid curves (Lambert et al., 2019; Liyang et al., 2020; Silva & Grassi, 2018), and sigmoid curve (Laghmara et al., 2019).

The sigmoid is a mathematical function which has a characteristic S-shape curve. Thanks to its nonlinearity, and the computational simplicity of its derivative, the sigmoid function is the most commonly used function in path planning. Of particular interest are the following three parameters: continuity, ease of discriminability, and simplicity (X. Li, Sun, et al., 2015), which constitute a motivation for using sigmoid functions for lane change maneuvers. A parameterized sigmoid based lane change operation is proposed in Ammour et al. (2020). In this method built for safe and comfortable lane change operation, the vehicle dynamics, and geometrical constraints, are considered in sigmoid functions for path planning. Ammour et al. create a path with a longitudinal distance, a lateral offset, and the trajectory curvature to develop a precise path tracking control strategy that minimizes the longitudinal and lateral error. Their simulation results show that the proposed algorithm has stable performance in lane change operation. A novel approach for obstacle avoidance on highway is proposed in Ammour et al. (2021), in their last method time-varied S-shaped sigmoid functions are used to define a restriction area based on the vehicle speed and distance to obstacles. The sigmoid S-shaped right side is an obstacle area, and the left side is defined as a safety gap for driving. Then the MPC algorithm is used for trajectory planning. The proposed method was only simulated in Matlab under three different scenarios. The simulation results show that this method has satisfactory results.

A Hybrid path planning algorithm is proposed in Lu et al. (2020), which combined Sigmoid curve with repulsive and attractive potential fields to improve performance, safety and feasibility of the generated paths. The overtaking path is created using the combination of obstacle avoidance, vehicle dynamics and sigmoid curves. The simulation results show that the proposed method improves vehicle stability and ride comfort during autonomous driving. An electronic driver assistance and collision avoidance system is proposed in Isermann et al. (2008). The system is a combination of object detection based on Lidar pointclouds fusion with camera images, path planning, and trajectory planning. The authors use the sigmoid curves for the path planning stage. The distance from front obstacles, safety area for lane change, and speed are the main parameters of their path planning algorithm. In the proposed system, the authors create the shortest evasive manoeuvre using sigmoid curves in consideration of the limitations of maximum lateral acceleration, maximum jerk, and dynamics of the steering actuator. Their experimental results and a comparison with klothoide functions show that the proposed method is able to provide a robust accident avoidance in Autonomous vehicles.

The method proposed in Ben-Messaoud et al. (2018) is a combination of parameterized sigmoid function and rolling horizon for generating a smooth path. The rolling horizon method is used for splitting the trajectory in convex areas. The authors argue that this method is effective in creating smooth and short overtaking paths. The rolling horizon method has been validated for lane changing process by simulation. The simulation results show that the algorithm effectively performs collision avoidance maneuvers for static and dynamic obstacles.

Supporting the use of sigmoid functions in generating paths, in X. Huang, Zhang, et al. (2019), it is used for creating overtaking maneuver paths. The proposed method has shown successful overtaking in two simulated scenarios. In this method, the decision making is based on reference paths, distance between two vehicles, relative speed and safety factors. Experimental results show that the proposed method can properly handle the overtaking operation with a fixed velocity of the preceding vehicle.

Inspiring from previous literature and preserving our existing platform for autonomous driving, it became natural to implement a sigmoid A-star decision making algorithm as a combination of the Hybrid A-star and sigmoid functions to build fast, smooth, safe and reliable overtaking maneuvers. Differently with respect to previous literature describing the path planning with sigmoid function only in simulation, here an experimental evaluation using an autonomous shuttle in a urban environment is proposed.

3 | KINEMATIC MODEL OF AUTOMATED SHUTTLES

Figure 2 describes the simplified kinematic model of the automated shuttle with 3 degrees of freedom (x, y, φ). As most of the maneuvers considered in this study are taking place at a low speed below 15 km/h, it is estimated that the lateral accelerations of wheels remain below 0.2 g. As a result, the hypothesis of negligible sideslip of wheels can be applied (S. Li et al., 2019). That, in turn, makes it possible to perform trajectory calculations on the basis of kinematic model without considering more detailed effects present in sophisticated dynamical models. Thus the control task of the overtake maneuver can be simplified to the task of controlling changes in heading angle at a given speed according to the described kinematic model. Additionally, as the considered speed limit is relatively low in the present study (for safety reasons), only 1–2 s (1–4 m of distance) are used for the initial acceleration at the beginning of any maneuver, and most of the movements of automated shuttle will occur at a constant speed.

The steering model in Figure 2 describes the change of heading angle during time step dt at given velocity v, and current steering

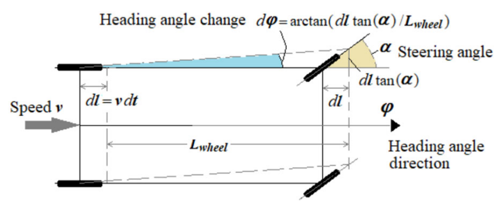

FIGURE 2 Kinematic model for the automated shuttle ("bicycle" model assumption). α denotes the steering angle of front wheels, φ is the heading angle of the vehicle, dl is the incremental forward displacement of rear wheels, $d\varphi$ is the resulting change of the heading angle, and finally L_{wheel} is the wheelbase

angle α. Possible sideslip corrections of the front wheels are omitted (S. Li et al., 2019), thus in a small time-step dt the rear wheels forward distance $dl = v \times dt$, while the movement of front wheels follows the steering angle α. As a result, incremental changes in the heading angle φ can be described by the following equation:

$$d\varphi = \arctan(dl \times \tan(\alpha))/L_{wheel}. \quad (1)$$

Equation (1) can be simplified for small time-steps (small longitudinal movement steps dl) to:

$$d\varphi = dl \times \tan(\alpha)/L_{wheel} \quad (2)$$

and for modest steering angles ($|\alpha| \le 0.1\ldots0.2$ rad) further to:

$$d\varphi = dl \times \alpha/L_{wheel}. \quad (3)$$

Using Equation (3) it is possible to conclude that to achieve reasonable changes of heading angle (e.g., 0.3 rad, approximately 17°), the required driving distance should be of the order of the wheelbase L_{wheel}. As a result, the required minimum distance to perform full lane change maneuver should be of order of 2–3 wheelbases. The information about the wheelbase measure can be used to specify reasonable safety distances to avoid abrupt maneuvers.

4 | GENERATION OF SIGMOID PATHS FOR OVERTAKING

In this section, the motivation for moving from other standard methods, for instance the hybrid A-start algoritm, to the improved sigmoid-based path generation are explained. Then the sigmoid curve-based overtaking path generation algorithm is described in details.

4.1 | Limitations of the hybrid A-star algorithm

One of the challenges in the path planning for autonomous shuttle is to use a time varying input, for example, occupancy grid maps, required to create a path in dynamic environments, change as the vehicle moves. Hybrid A-star (Dolgov et al., 2010) is a modified A-star algorithm designed for autonomous vehicles path planning in dynamic areas, and implemented in our prototype (Rassõlkin et al., 2018; Sell et al., 2018). First of all, the cost map, a fundamental concept in mobile robot navigation, is created from Lidar's data. Then, this map is used to find efficient and safe routes across the point cloud map. A 2D cost map (gray rectangle) is shown in Figure 3. Black areas in the gray cost map represent objects in the map.

When the autonomous shuttle stops because of an obstacle on the waypoint, the Hybrid A-star algorithm starts iterating to find an alternative waypoint to avoid the obstacle. After generating a new path, the vehicle starts driving along the path. The Hybrid A-star, as a path planning algorithm for a mobile robot, was successfully implemented and tested. However, the results were not satisfactory and reliable. There are three important limitations using the Hybrid A-star for path planning on autonomous shuttles: (I) Computing time: the iterations to find an acceptable suboptimal trajectory is time-consuming. (II) Reliability: similarly to a random selection algorithm, there is no way to know that the generated path is the correct and safe for the operation; This is risky for an autonomous driving application with passengers (see Figure 4). (III) Lack of reactivity: the vehicle cannot react fast enough to a dynamically changing environment.

Therefore, further investigation was conducted to explore the possibilities of using the proposed sigmoid-based method to replace the Hybrid A-star.

4.2 | Overtaking using the sigmoid-based path

The decision making steps that lead to an overtaking operation are shown in Figure 5. The algorithm first checks for a probable detected object, if a static object is found in the detection range, then an overtaking maneuver should take place, consisting of a 3–5 s pause

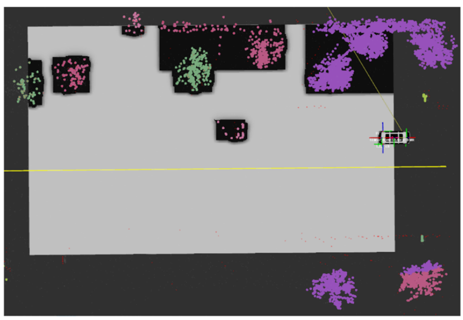

FIGURE 3 Cost map generated from the filtered 3D point cloud, the gray area corresponds to the drivable surface

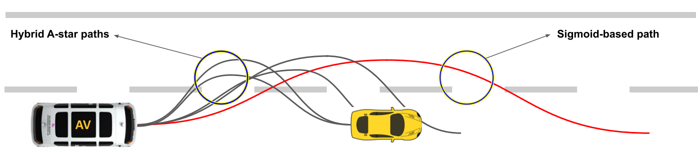

FIGURE 4 Comparison between the sigmoid-based and the Hybrid A-star generated path for overtaking

to prepare a new mission plan, then overtaking. In the case of the object restarting to move, the automated shuttle just decreases its speed and continues with the preset path generated using the A-star algorithm. In principle, this function pauses the trajectory following algorithm, that works on a pre-defined waypoint, and overlaps an optimized sigmoid-based generated path to overtake the obstacle, then resumes the original plan.

The first step in the overtaking maneuver is the detection of stationary objects that are blocking the initial desired waypoint path. Sometimes a vehicle might be blocking the entire road lane, but in many cases it could be blocking only part of it being parked on the roadside. If the object blocks the desired path, and it is in the detection range, the autonomous shuttle must stop. This is a safety requirement according to the current regulation, but in principle it is possible to overtake without a full stop. The detection range is approximately a rectangular area along the path with a predefined width, typically a little bit wider than the vehicle width (shown as green area in Figure 5).

In the implementation of the current overtaking algorithm, one of the main practical limitations is ensuring the smoothness of the steering angle changes maintaining the kinematic feasibility of the generated paths. This is necessary to ensure comfortable riding experience for passengers and to avoid a possible steering motor overload. In the case of overtaking maneuvers consisting of two lane changes (see Figure 6), one of the possible approaches to ensure a smooth turning and steering, is the application of two mathematically defined sigmoid functions.

In the field of automated shuttle control, only a few studies describe the application of sigmoid curves for the path planning, for example, Shao et al. (2018), S. Li et al. (2019), Lu et al. (2020). Our contribution along this line of research is the experimental report of a practical application of this methodology proposed in literature only using simulations.

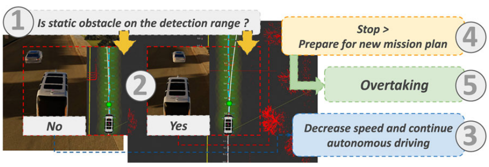

FIGURE 5 Five decision-making steps of the overtaking algorithm in the case an obstacle was detected by the shuttle

The first phase of the sigmoid path generation, referred to as STAGE I in Figure 6, can be described by the following Equation (4) including the exponential sigmoid term:

$$y_{sig1}(x) = y_0 - a(y_2 - y_0) + b\frac{(y_2 - y_0)}{1 + e^{\frac{x_{01} - x}{D_{01}}}}, \tag{4}$$

where y_0 is the initial y-coordinate of STAGE I starting point P_0, y_2 is the desired final y-coordinate of STAGE I, the sigmoid center point x-coordinate x_{01} is defined as the midpoint between P_1 and P_0 via $x_{01} = \frac{x_1 + x_0}{2}$, and D_{01} is the abruptness parameter that is calculated as $1/k$ of the distance between the x-coordinates of the points P_1 and P_0 as $D_{01} = \frac{x_1 - x_0}{k}$. Offered factor $1/k$ defines what percentage of the lateral shift occurs in the "tail stabilization" area between the points $P_1 - P_2$, and the rest occurs between the points P_0 and P_1. The fitting parameter values a and b are specified from the initial condition that the starting point of movement must coincide with P_0, and the boundery value of y-coordinate must correlate with y-coordinate of the point P_2.

Similarly, for sigmoid path in STAGE II, i.e. returning to the initial lane, a re-adaptation of Equation (4) can be used as follows:

$$y_{sig2}(x) = y_2 - a(y_4 - y_2) + b\frac{y_4 - y_2}{1 + e^{\frac{x_{23} - x}{D_{23}}}}, \tag{5}$$

where y_2 is the actual y-coordinate of STAGE II starting point (end-point of previous STAGE I), y_4 is the desired final y-coordinate of STAGE II, the sigmoid center point x-coordinate $x_{23} = (x_2 + x_3)/2$ is defined as midpoint between P_2 and P_3, and sigmoid abruptness parameter $D_{23} = (x_3 - x_2)/k$ is calculated by the difference of the x-coordinates of points P_3 and P_2.

5 | OVERTAKING SIMULATIONS

This section contains the simulation setup that evaluates the sigmoid-based overtaking algorithm. First, the kinematic model and the sigmoid curve were implemented in MATLAB to tune the model by finding an optimized set of parameters to be validated in the SVL simulation environment, and finally used in the experimental setting.

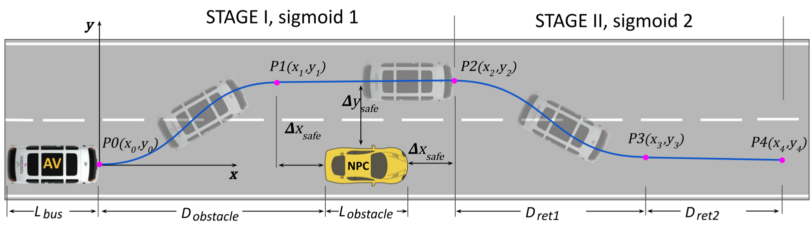

FIGURE 6 Overtaking maneuver description for a static obstacle. The two sigmoid paths are defined by the reference points $P_0 - P_1 - P_2$ and $P_2 - P_3 - P_4$, respectively

5.1 | Trajectory optimization

To evaluate the sigmoid curve utilization, and to achieve a comfortable ride with an accurate trajectory tracking (smooth steering), it is required to find the best parameter values for the sigmoid curve. Hence, the overtaking trajectory was simulated in a virtual model provided by MATLAB, and then a GA optimization algorithm (Wahde, 2008) was employed to find the best fit values. In the simulation, the "Vehicle Body 3DOF Dual Track" block as the automated vehicle kinematic model and the "Pure Pursuit" block to simulate the corresponding navigation controller were used.

The five main parameters were picked from the curve definition as follows: abruptness factor $k_{abruptness}$, distance to the obstacle D_{obs}, safety distance to the obstacle in the longitudinal direction Δx_{safe}, and the two longitudinal travel distances while returning to the driving lane D_{ret1}, D_{ret2} (see Figure 6). Moreover, the lookahead distance parameter $Lo_{distance}$ of the pure-pursuit controller was included to the optimization process to increase the trajectory following accuracy. The minimization of the following error leads to precise movement, which is required for maneuvers such as overtaking.

All the curve parameter values, and their descriptions are reported in Table 1 including their range of validity. After 500 simulation runs, GA found the optimum values reported in Table 2. For a better understanding of the optimization outcome, initial values were also suggested for the parameters listed in Table 2. Then, two simulations were performed by employing the initial and optimal data, while steering changes and trajectory tracking were recorded during the mission. Figure 7 shows the performance in tracking and steering by using suboptimal parameters and the optimized one. Figure 7a shows the not-optimized trajectory (red dots) and how the vehicle follows them (black circles), while Figure 7b shows the optimized trajectory. It is clear that the AV followed the optimized trajectory more accurately and also the steering changes are smother (see Figure 8).

5.2 | SVL simulation

Several realistic car simulators are powered by a physics engine compatible with our automated shuttle control software, such as SVL and CARLA. They use modern game engine features like Unreal and Unity, giving them the power to create complex virtual environments as well as realistic rendering.

In this paper, the SVL simulator (Rong et al., 2020) is used to assess the sigmoid-based overtaking algorithm. This simulator is based on the Unity game engine that provides various environments and car models in the simulation. For testing the maneuver in the simulator, a detailed iseAuto 3D model with the equipped Lidar was implemented inside Unity, and assigned the engine.

The evaluation process was built by creating a simple overtaking scenario inside the SVL simulator (see Figure 9). In this scenario, an NPC (Non-player character) car is placed in the middle of the waypoint, then the shuttle decision making capabilities are observed. Figure 9b reports the ROS visualization software screen that shows the point cloud of the simulated environment, the shuttle, and the vehicle's straight desired waypoint. In this figure, the shuttle detects an object in its waypoint's detection range (green area) and stops to take a decision on whether to perform an overtaking maneuver or not. The red line is visualized before the NPC as a stop indicator. Figure 9a shows the SVL simulation environment, including the iseAuto 3D model, and a stopped NPC. In the next step, the shuttle starts to generate a new smooth waypoint based on the sigmoid curve.

Figure 10 shows four time instants (frames) from start to end of the overtaking operation inside the simulator and the corresponding ROS visualization. Frame 1 shows that the new path, presented by Equation (1), is generated for STAGE I of overtaking Figure 10. Next, the shuttle starts to move toward the new waypoint (frame 2). After passing the object, the shuttle generates the new path starting the STAGE II of the mission using Equation (4). Finally, in the last frame, it returns to its original waypoint.

TABLE 2 Parameter values in a initial and optimized case

Param.	Init. val.	Optim. val.	Param.	Init. val.	Optim. val.
$Lo_{distance}$	5 m	3.88 m	Δx_{safe}	3 m	3.49 m
$k_{abruptness}$	10	3.62	$D_{ret.1}$	13 m	13.04 m
$D_{obstacle}$	13 m	14.28 m	$D_{ret.2}$	13 m	10.93 m

TABLE 1 Parameters for the kinematic model used in the optimization and experimental setup

Param.	Val.	Description	Param.	Val.	Description
$D_{obstacle}$	8–15 m	Distance to obstacle	$L_{obstacle}$	4 m	Obstacle length in x-dir
W_{road}	6 m	Road width	$W_{obstacle}$	2 m	Obstacle width in y-dir
Δx_{safe}	3–5 m	Safety distance in x-dir.	$D_{ret.1\&2}$	5–15 m	Travel distance to the initial lane
L_{bus}	3.4 m	Shuttle length	$Lo_{distance}$	3–8 m	Pure Pursuit Look-ahead distance
L_{wheel}	2.55 m	Shuttle wheel base	Δy_{safe}	2 m	Safety distance in y-dir
v	10 km/h	Shuttle constant speed	$k_{abruptness}$	3–8	abruptness factor

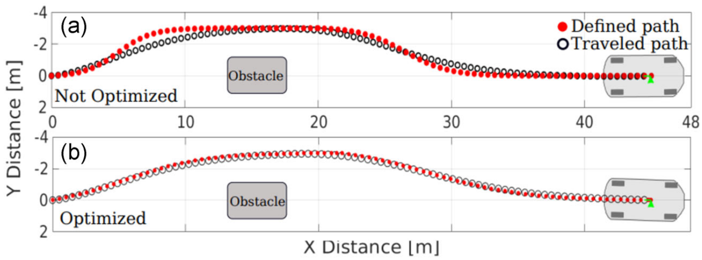

FIGURE 7 The comparison of trajectory following (a) not optimized case; (b) optimized case

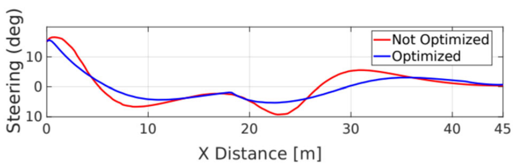

FIGURE 8 The comparison of and steering angle changes in the optimized and non-optimized case

6 | TESTBED

In this section, our customized automated vehicle and the testing environment are described.

6.1 | iseAuto automated shuttle

Our study was carried out by using the automated shuttle, iseAuto, at Tallinn University of Technology (TalTech), Estonia. The iseAuto is an automated shuttle belonging to the autonomous vehicles research group and operating in the campus for experimental and study purposes (see Figure 11). Previously proposed mechatronic design methodologies (Christophe et al., 2009; Sell et al., 2008) were implemented to focus on early design stages to develop the iseAuto shuttle from a scratch. The iseAuto project's objective was to build an open-source automated shuttle and establish a smart city testbed (Sell et al., 2020) in the TalTech campus. The further concept is to integrate an autonomous shuttle service with industrial parks as a part of Industry 4.0 concept (Sell et al., 2019). The automated shuttle and the testbed are connected to its digital twin, allowing designers to execute all development in simulation first. The simulation environments, interfaces, and concepts are described in detail in Medrano-Berumen et al. (2020), Malayjerdi et al. (2020).

The iseAuto high-level software architecture is based on ROS (Robotic Operating System). Perception, detection, and planning are performed by Autoware (Kato et al., 2018) an open-source ROS-based autonomous driving stack. Many advanced algorithms are already implemented, such as lane following, obstacle avoidance, traffic light detection, lane detection, etc. Lidars and Global Navigation Satellite System (GNSS) are used for localization and path following. The vehicle is equipped with two Velodyne Lidars at the top front (VLP-32) and top rear (VLP-16) of the vehicle, and two front sides Robosense RS-Bpearl to decrease blind spots. Furthermore, one RS-Lidars-16 is installed at the front bumper to detect small objects in front of the vehicle that are not in the other Lidars' field of view. Figure 12 shows the position of the Lidar sensors on the shuttle. Processes such as calibration, filtering, and concatenation were performed on the Lidars' point cloud to optimize perception capabilities.

The Lidars' raw point cloud is often noisy, contains outliers, and may cause errors in the detection process. Hence, it is essential to use a multi-step point-cloud filtering system that increases the perception algorithm performance and accuracy. First, ground filtering is applied to raw data to separate the ground points. Ring ground filter (Narksri et al., 2018) in Autoware is used as filtering method. This filter cuts unnecessary points, and avoids false detection of ground points as objects.

Next, from the Point Cloud Library (PCL), four different filtering methods are applied to filter the raw point cloud data: Voxel Grid, PassThrough, Statistical Outlier Removal and Radius Removal. Finally, the points are ready to be processed by the Euclidean clustering algorithm. These clustered points are used in the A-star algorithm to derive a grid map that is then used as a global scanning method to find an optimal path.

6.2 | Testsite experiment

After evaluating and verifying the new proposed method through simulation, as described in Section 5, the best found parameters were implemented on the iseAuto for testing a real scenario at the TalTech AV test site. The shuttle main processing unit is a PC with Ubuntu 18.04 operating system, AMD Ryzen threadripper 1950x, and two GeForce GTX-1080-Ti GPU. The experiments were conducted on a standard road with two lanes as shown in Figure 13. The shuttle speed was limited to 15 km/h (Figure 14).

7 | RESULTS AND DISCUSSION

To assess the performance of the maneuvers an extensive experimental campaign using the iseAuto automated shuttle was conducted. The steering angle feedback was recorded during the

FIGURE 9 (a) Simulation of an overtaking scenario in the SVL simulator. (b) The ROS visualization shows the vehicle sensor data, waypoint, and position in the map

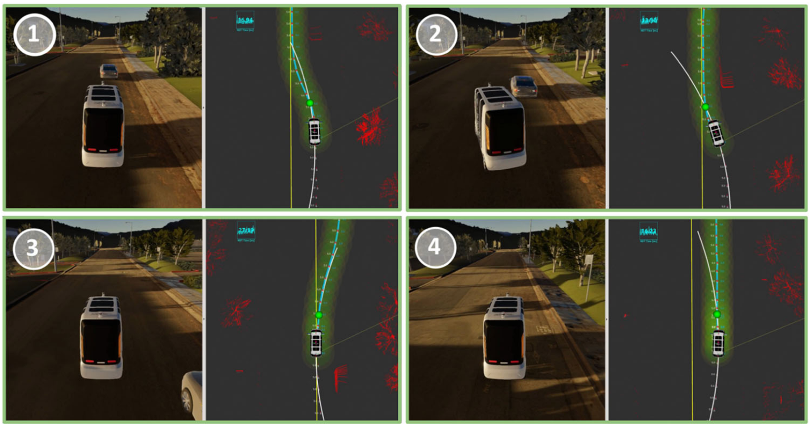

FIGURE 10 Four characteristic time instants in SVL and the corresponding ROS simulation environment, describing the overtaking maneuver based on the smooth sigmoid-based path

FIGURE 11 TalTech iseAuto autonomous shuttle (left) and 3D eagle view map of the test site in TalTech campus (right)

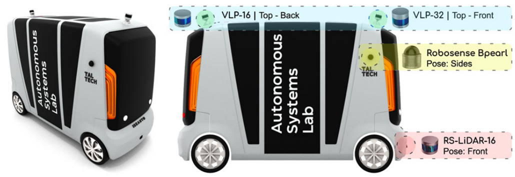

FIGURE 12 The iseAuto shuttle bus with the indication of the position of its three different types of Lidars

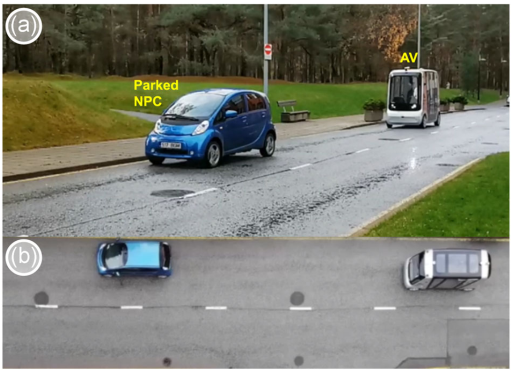

FIGURE 13 Automated shuttle iseAuto and a blue car stopped in the area where the experiments were carried out. Front view (a) and Drone view (b)

process both in simulation and experimental setup. The results show smoothness of the motion, efficiency and high reliability during the operation.

7.1 | Validation

To validate the optimized sigmoid-based paths, the results from the proposed method in simulation and in the experimental setup were recorded and reported in Figure 15b. The error between the simulation and the experiment is below 10% demonstrating the effectiveness of the method. Figure 15a shows the steering data for the two methods in simulation. The proposed optimized sigmoid-based method generates smoother steering angle changes than the guided Hybrid A-start, as the spikes visible on the blue-line at time 0 and 12 s were eliminated. Figure 15b shows the simulation result with the corresponding data from the experiment. Additionally, smooth trajectories yield to an improved use of the steering motors, since the angle range is reduced of about 5°. This reduces the long term usage of the motors, guarantees long-term steering motor performance and prevents unexpected failures.

This result validates the reliability of the experimental setup with respect to the simulation environment. In other words, the simulation of the overtaking maneuver presented in this paper shows that it is not essential to test newly developed algorithms directly on the automated shuttle; instead, they can be initially tested and verified

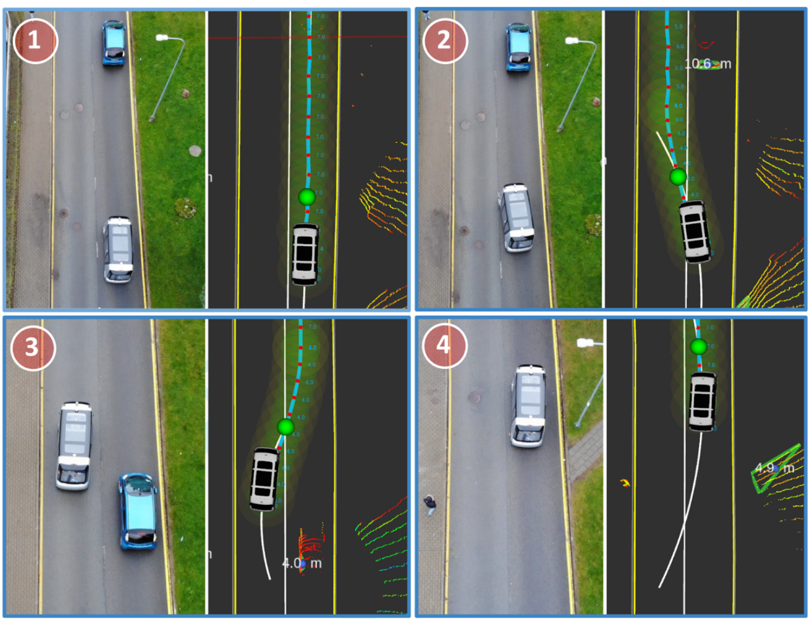

FIGURE 14 Different captured time frames of real overtaking experiment (drone view on left, ROS visualization on right)

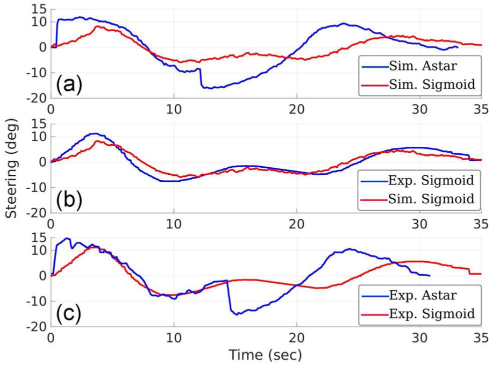

FIGURE 15 (a) Steering angle simulation data of the two different methods; guided Hybrid A-star and optimized Sigmoid. (b) Steering angle data of the simulation versus the experiment on sigmoid method. (c) Steering experiments data from the two methods

through the simulation platform. Although the stopped car scenario selected for the overtaking examination did not cover all the complex situations, it is an initial step for studying and introducing a verification platform for further research. Scenarios such as passing moving objects or vehicles coming from the opposite direction can be easily created and tested in the simulation without any danger.

7.2 | Experimental comparison

Similarly to the simulation environment, a long experimental campaign was conducted with both the sigmoid-based, the Hybrid A-star, and the Guided Hybrid A-star overtaking algorithm, while recording the steering data. In Figure 15c, the steering data of the sigmoid-based proposed method is compared against the corresponding data of the A-star-based method during the real experiments. As in Figure 15a, this plot also shows that the steering angle in the sigmoid-based path is smoother than the corresponding steering angle using the Guided Hybrid A-star method.

TABLE 3 Experimental results reported from four case studies that were carried out with three different methods

Method	Parameter	Case(a)	Case(b)	Case(c)	Case(d)
Hybrid A-star	Duration(s)	NA	NA	NA	NA
	Computational time (s)	77.36	36.56	159.32	112.51
	Maximum wheel angle (rad)	**0.22**	**0.36**	**0.35**	**0.32**
	Path length (m)	29.62	31.05	39.06	44.88
	Number of iterations	154	72	318	224
	Created path	3	3	13	8
	Overtaking result	Failed	Failed	Failed	Failed
Guided Hybrid A-star	Duration (s)	23.36	25.80	33.90	39.30
	Computation time (s)	10	10	10	10
	Maximum wheel angle (rad)	**0.31**	**0.36**	**0.35**	**0.32**
	Path length (m)	29.62	31.05	39.06	44.88
	Number of iterations	1	1	2	3
	Created path	2	2	2	2
	Overtaking result	Pass	Pass	Pass	Pass
Optimized Sigmoid curve	Duration	25.23	28.32	39.40	45.60
	Computation time (s)	10	10	10	10
	Maximum wheel angle (rad)	**0.28**	**0.30**	**0.29**	**0.30**
	Path length (m)	40.23	45.36	53.42	61.34
	Number of iterations	2	2	3	4
	Created path	2	2	3	4
	Overtaking result	Pass	Pass	Pass	Pass

Note: Bold values indicate the smoothening efficiency of the optimized sigmoid method.

7.3 | Case study

To further demonstrate the effectiveness of the proposed method, four different case study have been proposed in simulation and experimental setup. In the experimental setup the trajectory driven by the vehicle was recorded using a drone from the top view (see Figure 13a). The results are shown in Table 3.

7.3.1 | Overtake using guided lane change

In the first experiment an alternative waypoint was created manually on the opposite lane. Such waypoint is generated as a safe path for the lane change process during the overtaking maneuver. Figure 16 shows the two waypoints, in each figure the bold blue line represents the original path, and the dash line at the vehicle left side is the alternative one. As shown in Figure 16a, the green ball, which is the local control goal, is on the original waypoint. When the autonomous shuttle detects an obstacle on the path, the alternative path changes to the main path. As seen in Figure 16b, the green ball suddenly switched to the alternative waypoint that causes a rough and sharp motion for lane changing Figure 16c. Then the same lane change happens when driving back to the original path after passing the obstacle. In Figure 16d the original waypoint is enabled so the green ball jumps on it. As seen in Figure 16e cause the same issue that happened on 16c. This fast switch between routes might cause safety issues for passengers and technical problems for the steering mechanism due to its restriction. To solve this issue, in another test, the Hybrid A-star algorithm was used for overtaking the same scenario.

7.3.2 | Overtake with Hybrid A-star path planning algorithm

In the second experiment, the Hybrid A-star algorithm is customized for automated shuttle overtaking (Figure 17). As shown in Figure 17a–e, the Hybrid A-star algorithm generated different paths in different time-stamps. As seen in Figure 17a when the autonomous shuttle detects an obstacle on its waypoint, the Hybrid A-star creates a path for obstacle avoidance. Then the shuttle starts driving on the newly generated path. As shown in Figure 17b at 6 m from the obstacle, suddenly the Hybrid A-star updates its path close to the obstacle resulting in an unexpected shuttle hard brake due to safety range sensors detecting an obstacle in the close range. In the next iteration, (see Figure 17c) the Hybrid A-star generates a new avoiding path. Following the generated path, and once the obstacle is overtaken, the Hybrid A-star updates its path once more (see Figure 17d). At this point there is a conflict of different generated paths as the shuttle attempts to turn right but the Hybrid A-star updates its waypoint again 17e.

Based on the experimental result, the Hybrid A-star algorithm cannot be considered safe and reliable for the application in our vehicle. In the next experiment, the guided Hybrid A-star algorithm performance is evaluated.

7.3.3 | Overtake with the guided Hybrid A-star

In this section, the guided Hybrid A-star algorithm performance is described from a practical overtaking operation point of view (Figure 18). In this method, four safe points are defined for two

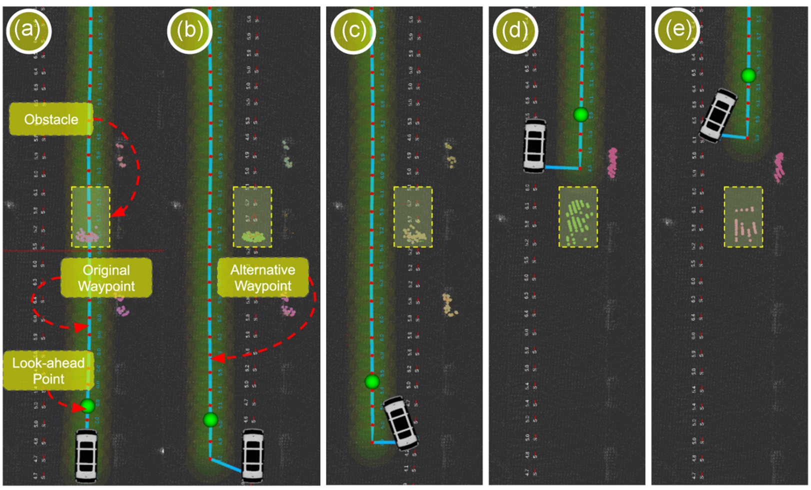

FIGURE 16 Guided lane change algorithm in the first overtaking experiment: (a) initial obstacle detecting situation; (b) switching the trajectory to the alternative waypoint for avoiding the obstacle; (c) driving at maximum speed and steering angle to reach the new waypoint; (d) switching the trajectory to the original waypoint after passing the obstacle. (e) driving at maximum speed and steering angle to reach the original waypoint

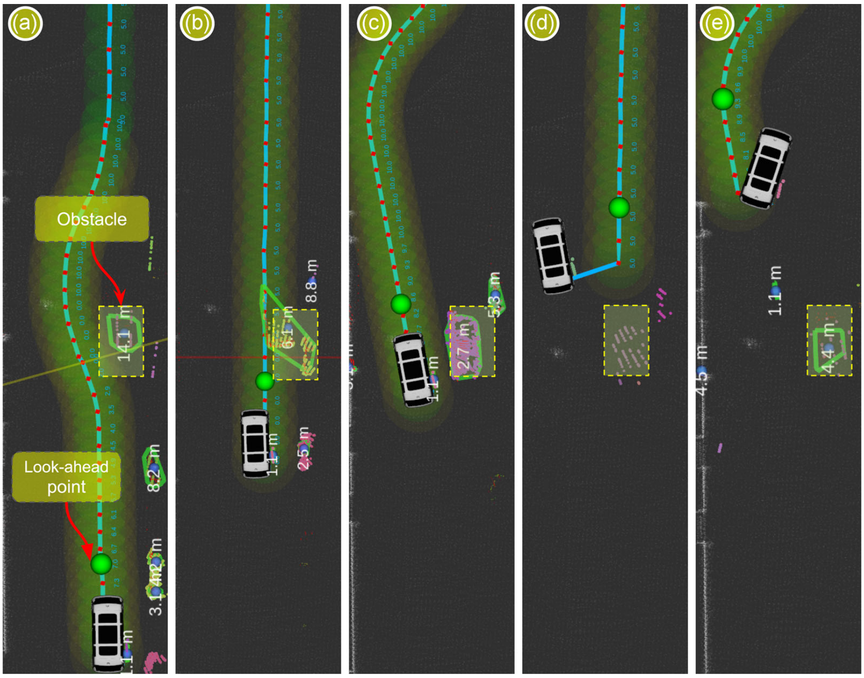

FIGURE 17 Hybrid A-star iterations for the overtaking maneuver in the test-bed experiment; (a) obstacle detection at 14.2 m and creating a avoidance path; (b) change of the avoidance path to a straight path at 5.8 m from the obstacle; (c) an additional avoidance path is created; (d) while driving on the previous path, the next iteration creates a new path; (e) the autonomous shuttle attempts to drive on the generated path with previous iteration, suddenly the iteration generates a conflicting path choice

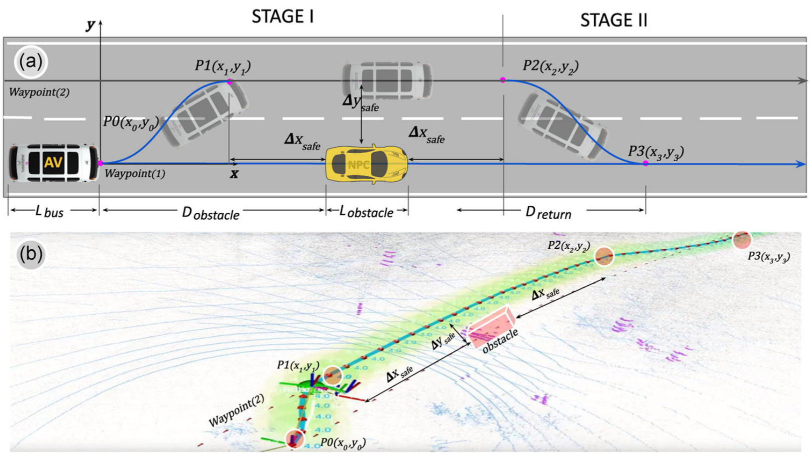

FIGURE 18 Overtaking maneuver description with the guided Hybrid A-star method. Four references points P_0, P_1, P_2, P_3 are defined for Hybrid A-star to create a path (a). ROS visualization from experiment (b)

overtaking stages. P_0 is the safety distance between the automated shuttle and the obstacle. P_1 is the goal for the Hybrid A-star to create a smooth path from P_0 to P_1. The P_1 point is based on the safety distance from the obstacle in the longitudinal direction Δx_{safe} and lateral direction Δy_{safe}. When the automated shuttle reaches the point P_1, driving on a straight line toward the next predefined point P_2. Also this point is based on Δx_{safe} and Δy_{safe}. Then P_2 is set as the starting point for the Hybrid A-star, and P_3 as the end point to create a path.

The point P_3 is on the main waypoint and fulfils the safety distance requirement from the obstacle.

An additional experiment is shown in Figure 18b, here the automated shuttle calculates the car length. Based on the vehicle length, the algorithm calculates the position of the points P_1, P_2, and P_3, and finally the automated shuttle drives on the overtaking path. Reliability and safety issues in the Hybrid A-star algorithm are solved with this method. However, there are two big drawbacks. First, the automated shuttle still requires the second predefined waypoint for choosing the points P_1, P_2. Second, the automated shuttle must run the Hybrid A-star algorithm twice, one for each overtaking stage.

7.3.4 | Overtake using the optimized sigmoid-based method

The performance of the proposed sigmoid-based method is described using four practical overtaking operations (see Figure 19).

Case study (a): In this case study, similarly to the simulation environment, the experiment is illustrated by using different time frames in which the first is the top view captured by a drone (see Figure 14). The second is the corresponding screenshot from the ROS visualization software, which shows the pre-designed waypoint and the current shuttle position. Figure 14 (frame 1) shows that the automated shuttle correctly detected the vehicle and indicated using a red line in the ROS visualization. Hence, the automated shuttle should stop. Clearly, the obstacle vehicle is not shown in the ROS visualization because the pre-defined path could not predict the presence of the vehicle, demonstrating the high adaptive capability to this dynamically changing environment. When the shuttle stops, the overtaking operation can begin, and the new waypoint is generated (frame 2). Observe that Figure 14 (frame 2) also includes an indication of the distance between the automated shuttle and the obstacle in the ROS visualization. Frame 3 shows that the vehicle passed the car and started to return to its original lane or waypoint, also in this case the frame includes an indication of the longitudinal distance between the automated shuttle and the overpassed obstacle. Finally, the last frame shows that the shuttle is back to the original lane and continues its mission.

Case study (b): During the initial stage of an overtaking maneuver, it is very hard to calculate the total path length as the length of the obstacle is not known a-priori and thus there is no guarantee about the total duration of the maneuver. In this case a perception algorithm using the right-side Lidar data, is added to proposed method as an additional decision making step (see Figure 20).

The 3D overtaking area in Figure 20(1) is created at the autonomous shuttle right side. The length, in this case, is 16 m (covering 6 m from the back side to 10 m on the front side), the height and depth are 2 m. Also the Yolov3, a real time object detection system (Redmon & Farhadi, 2018) that uses the autonomous shuttle right camera is combined with the lidar perception. This perception algorithm is implemented as a service, and it runs for overtaking maneuvers. At the beginning of the STAGE I, the perception service is called by the overtaking algorithm. When the autonomous shuttle is at the

FIGURE 19 Different scenarios for overtaking. Overtake from a long vehicle (a), Overtake from a long and small vehicle (b). Overtake from a long vehicle and two small vehicle (c)

second lane, before going to STAGE II, the service calculates the overtaken vehicle length and check the 3D dimensional box (Figure 20(1)) for lane change permission. If the overtaking area is free, the overtaking algorithm uses the obstacle length to initiate the STAGE II starting at a safety distance from the overtaken obstacle.

Case study (c): In this experiment the automated shuttle overtakes two vehicles. As seen in Figure 21b, the performance of proposed method for overtaking is tested in different scenarios. In this case, the autonomous shuttle calls the perception service, then the service calculates the length of the overtaken obstacle and detects another vehicle on the road, see Figure 20(3). As a consequence, the overtaking algorithm does not have the lane change back permission and continues driving until permission as in Figure 21(b). Once the vehicle has the permission, the second STAGE II makes a path for lane change back at the safe distance from the second vehicle position.

Case study (d): In this experiment, the performance of the perception algorithm is tested to overtake three cars on the road. In this scenario, after passing the second car, another car detected by perception algorithm. So the autonomous shuttle continues driving until permission. Once the vehicle has the permission, make a lane change back for returning to the main lane. As seen in (see Figure 21c) the automated shuttle overtakes three cars successfully.

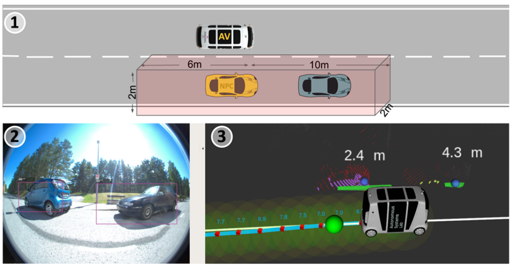

FIGURE 20 Perception algorithm using the right-side lidar data. (1) The top-view of the bus in which the obstacle box is shown. (2) Right side camera view. (3) ROS visualization of the vehicle path

8 | CONCLUSION

In this paper, an optimized sigmoid-based overtaking maneuver generator for an autonomous shuttle is experimentally studied. To overcome the limitation of the current state-of-the-art algorithms, a modified path planning algorithm based on the sigmoid curve is proposed.

Smoothness, safety, and reliability were the main criteria for the automated shuttle overtaking process that were in the core values of our design. The proposed overtaking algorithm has been developed, formulated and implemented on a real automated shuttle, and experiments were conducted on real road scenarios. The proposed methodology for overtaking from one vehicle can be extended to overtaking multiple vehicles. High-fidelity simulations were used to validate the efficiency and to predict the behavior of the proposed algorithm. The results showed that the proposed method efficiently reduced the steering effort and removed sharp changes that led to uncomfortable and unsafe operations. According to our experimental results, the proposed method over performs other state of the art methods, specifically the Hybrid A-star and the guided Hybrid A-star.

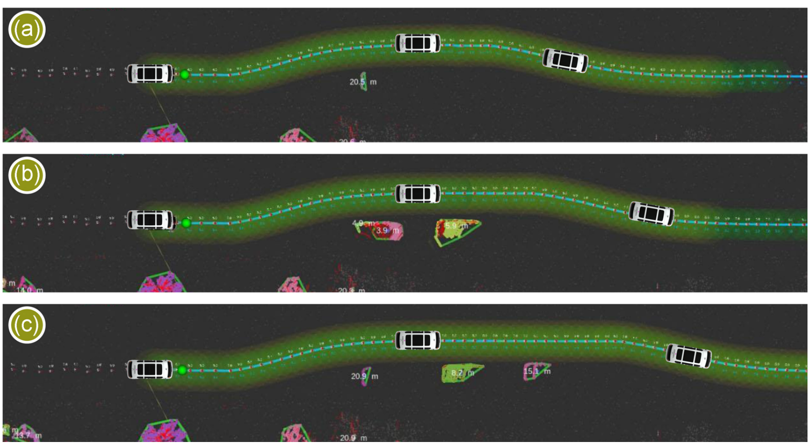

FIGURE 21 Different scenarios of real overtaking experiment with proposed method in the ROS visualization environment. (a) Overtake a long vehicle; (b) overtake a long and small vehicle case; (c) overtake a long vehicle and two small vehicle

Future investigations are required to focus on advanced obstacle avoidance methods drawn from this study for complex scenarios such as moving vehicles, extend and improve perception for fast and precise detection, and overtaking with high speed. It is required to use the developed method for movement prediction in this regard. Furthermore, an extra decision-making step should to be added to the algorithm based on information from the camera image and machine learning algorithms.

ACKNOWLEDGMENTS

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study has received funding from two grants: the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No 2014-2020.4.01.20-0289.

CONFLICT OF INTERESTS

The authors declare that there are no conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Ehsan Malayjerdi http://orcid.org/0000-0002-6526-6059

REFERENCES

Ammour, M., Orjuela, R. & Basset, M. (2020) Trajectory reference generation and guidance control for autonomous vehicle lane change maneuver. In *2020 28th Mediterranean Conference on Control and Automation (MED)* (pp. 13–18). Saint-Raphaël, France: IEEE 2020 28th Mediterranean Conference on Control and Automation (MED).

Ammour, M., Orjuela, R. & Basset, M. (2021) Collision avoidance for autonomous vehicle using MPC and time varying sigmoid safety constraints. *IFAC-PapersOnLine*, *54*(10), 39–44.

Ben-Messaoud, W., Basset, M., Lauffenburger, J.-P. & Orjuela, R. (2018) Smooth obstacle avoidance path planning for autonomous vehicles. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (pp. 1–6). Madrid, Spain: IEEE.

Carroll, K. P., McClaran, S. R., Nelson, E. L., Barnett, D. M., Friesen, D. K. & William, G. N. (1992) Auv path planning: an a* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In *Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology* (pp. 79–84). Washington, DC, USA: IEEE.

Chae, H. & Yi, K. (2020) Virtual target-based overtaking decision, motion planning, and control of autonomous vehicles. *IEEE Access*, *8*, 51363–51376.

Chen, Y. & Huang, Y. (2018) An overtaking obstacle algorithm for autonomous driving based on dynamic trajectory planning. In *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 1315–1322). Huangshan, China: IEEE.

Christophe, F., Sell, R., Bernard, A. & Coatanéa, E. (2009) Opas: ontology processing for assisted synthesis of conceptual design solutions. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 49026, pp. 249–260). San Diego, California, USA: ASME.

Claussmann, L., Carvalho, A. & Schildbach, G. (2015) A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams. In *2015 European Control Conference (ECC)* (pp. 2976–2982). Linz, Austria: IEEE.

Dixit, S., Fallah, S., Montanaro, U., Dianati, M., Stevens, A., Mccullough, F. & Mouzakitis, A. (2018) Trajectory planning and tracking for autonomous overtaking: state-of-the-art and future prospects. *Annual Reviews in Control*, *45*, 76–86.

Dolgov, D., Thrun, S., Montemerlo, M. & Diebel, J. (2010) Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, *29*(5), 485–501.

Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T. & Jurišica, L. (2014) Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, *96*, 59–69.

Erke, S., Bin, D., Yiming, N., Qi, Z., Liang, X. & Dawei, Z. (2020) An improved a-star based path planning algorithm for autonomous land vehicles. *International Journal of Advanced Robotic Systems*, *17*(5), 1729881420962263.

Feng, S., Qian, Y. & Wang, Y. (2021) Collision avoidance method of autonomous vehicle based on improved artificial potential field algorithm. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *235*(14), 3416–3430.

Hegeman, G., Brookhuis, K. & Hoogendoorn, S. (2005) Opportunities of advanced driver assistance systems towards overtaking. *European Journal of Transport and Infrastructure Research*, *5*(4).

Huang, X., Zhang, W. & Li, P. (2019) A path planning method for vehicle overtaking maneuver using sigmoid functions. *IFAC-PapersOnLine*, *52*(8), 422–427.

Huang, Y., Ding, H., Zhang, Y., Wang, H., Cao, D., Xu, N. & Hu, C. (2019) A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach. *IEEE Transactions on Industrial Electronics*, *67*(2), 1376–1386.

Isermann, R., Schorn, M. & Stählin, U. (2008) Anticollision system Proreta with automatic braking and steering. *Vehicle System Dynamics*, *46*(S1), 683–694.

Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M. & Kitsukawa, Y. (2018) Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)* (pp. 287–296). Porto, Portugal: IEEE.

Laghmara, H., Boudali, M.-T., Laurain, T., Ledy, J., Orjuela, R., Lauffenburger, J.-P. & Basset, M. (2019) Obstacle avoidance, path planning and control for autonomous vehicles. In *2019 IEEE Intelligent Vehicles Symposium (IV)* (pp. 529–534). Paris, France: IEEE.

Lambert, E., Romano, R. & Watling, D. (2019) Optimal path planning with clothoid curves for passenger comfort. In *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)* (Vol. 1, pp. 609–615). SciTePress.

Lattarulo, R., Marcano, M. & Pérez, J. (2017) Overtaking maneuver for automated driving using virtual environments. In *International Conference on Computer Aided Systems Theory* (pp. 446–453). Springer.

Li, C., Wang, J., Wang, X. & Zhang, Y. (2015) A model based path planning algorithm for self-driving cars in dynamic environment. In *2015 Chinese Automation Congress (CAC)* (pp. 1123–1128). IEEE.

Li, S., Li, Z., Yu, Z., Zhang, B. & Zhang, N. (2019) Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control. *IEEE Access*, *7*, 132074–132086.

Li, X., Sun, Z., Cao, D., He, Z. & Zhu, Q. (2015) Real-time trajectory planning for autonomous urban driving: framework, algorithms, and verifications. *IEEE/ASME Transactions on Mechatronics*, *21*(2), 740–753.

Liu, T., Deng, Z., Tang, X., Wang, H. & Cao, D. (2019) Predictive freeway overtaking strategy for automated vehicles using deep reinforcement learning. In *2019 3rd Conference on Vehicle Control and Intelligence (CVCI)* (pp. 1–6). Hefei, China: IEEE.

Liu, T., Huang, B., Deng, Z., Wang, H., Tang, X., Wang, X. & Cao, D. (2020) Heuristics-oriented overtaking decision making for autonomous vehicles using reinforcement learning. *IET Electrical Systems in Transportation*, *10*(4), 417–424.

Liyang, S., Yu, H., Xuezhi, C., Changhao, J. & Miaohua, H. (2020) Path planning based on clothoid for autonomous valet parking. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (pp. 389–393). Chengdu, China: IEEE.

Lu, B., He, H., Yu, H., Wang, H., Li, G., Shi, M. & Cao, D. (2020) Hybrid path planning combining potential field with sigmoid curve for autonomous driving. *Sensors*, *20*(24), 7197.

Majidi, M., Arab, M. & Tavoosi, V. (2017) *Optimal real-time trajectory planning of autonomous ground vehicles for overtaking moving obstacles*. Technical report, SAE Technical Paper.

Malayjerdi, M., Kuts, V., Sell, R., Otto, T. & Baykara, B. C. (2020) Virtual simulations environment development for autonomous vehicles interaction. In *IMECE2020*. ASME.

Medrano-Berumen, C., Malayjerdi, M., Akbaş, M. İ., Sell, R. & Razdan, R. (2020) Development of a validation regime for an autonomous campus shuttle. In *2020 SoutheastCon* (pp. 1–8). IEEE.

Milanés, V., Llorca, D. F., Villagrá, J., Pérez, J., Fernández, C., Parra, I., González, C. & Sotelo, M. A. (2012) Intelligent automatic overtaking system using vision for vehicle detection. *Expert Systems with Applications*, *39*(3), 3362–3373.

Mo, S., Pei, X. & Wu, C. (2021) Safe reinforcement learning for autonomous vehicle using Monte Carlo tree search. *IEEE Transactions on Intelligent Transportation Systems*.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D. & Ettinger, S. (2008) Junior: the Stanford entry in the urban challenge. *Journal of field Robotics*, *25*(9), 569–597.

Naranjo, J. E., Gonzalez, C., Garcia, R. & De Pedro, T. (2008) Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. *IEEE Transactions on Intelligent Transportation Systems*, *9*(3), 438–450.

Narksri, P., Takeuchi, E., Ninomiya, Y., Morales, Y., Akai, N. & Kawaguchi, N. (2018) A slope-robust cascaded ground segmentation in 3d point cloud for autonomous vehicles. In *2018 21st International Conference on intelligent transportation systems (ITSC)* (pp. 497–504). IEEE.

Palatti, J., Aksjonov, A., Alcan, G. & Kyrki, V. (2021) Planning for safe abortable overtaking maneuvers in autonomous driving. *arXiv preprint arXiv:2104.00077*.

Rassõlkin, A., Sell, R. & Leier, M. (2018) Development case study of the first Estonian self-driving car, iseauto. *Electrical, Control and Communication Engineering*, *14*(1), 81–88.

Redmon, J. & Farhadi, A. (2018) Yolov3: an incremental improvement. arXiv.

Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S. & Možeiko, M. (2020) Lgsvl simulator: a high fidelity simulator for autonomous driving. *arXiv preprint arXiv:2005.03778*.

SAE On-Road Automated Vehicle Standards Committee. (2014) Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J*, 3016, 1-16.

Sell, R., Coatanéa, E. & Christophe, F. (2008) Important aspects of early design in mechatronic. In *6th International Conference of DAAAM Baltic Industrial Engineering*, Tallinn.

Sell, R., Leier, M., Rassõlkin, A. & Ernits, J.-P. (2018) Self-driving car iseauto for research and education. In *2018 19th International Conference on Research and Education in Mechatronics (REM)* (pp. 111–116). Delft, Netherlands: IEEE.

Sell, R., Rassõlkin, A., Wang, R. & Otto, T. (2019) Integration of autonomous vehicles and industry 4.0. *Proceedings of the Estonian Academy of Sciences*, *68*(4), 389–394.

Sell, R., Soe, R.-M., Wang, R. & Rassõlkin, A. (2020) Autonomous vehicle shuttle in smart city testbed. In *International Forum on Advanced Microsystems for Automotive Applications* (pp. 143–157). Springer.

Shao, X., Liu, J. & Wang, H. (2018) Robust back-stepping output feedback trajectory tracking for quadrotors via extended state observer and sigmoid tracking differentiator. *Mechanical Systems and Signal Processing*, *104*, 631–647.

Shinpei, K. (2017) Autoware: ROS-based OSS for urban self-driving mobility. In *ROSCon Vancouver*. 2017. Open Robotics.

Shufeng, W. & Junxin, Z. (2018) A research on overtaking lane planning for intelligent vehicles based on improved artificial potential field method. *Automobile Technology*, *3*(2), 2–7.

Silva, J. A. & Grassi, V. (2018) Clothoid-based global path planning for autonomous vehicles in urban scenarios. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4312–4318). IEEE.

Viana, Í. B., Kanchwala, H. & Aouf, N. (2019) Cooperative trajectory planning for autonomous driving using nonlinear model predictive control. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)* (pp. 1–6). Graz, Austria: IEEE.

Wahde, M. (2008) *Biologically inspired optimization methods: an introduction*. WIT Press.

Wahid, N., Zamzuri, H., Amer, N. H., Dwijotomo, A., Saruchi, S. A. & Mazlan, S. A. (2020) Vehicle collision avoidance motion planning strategy using artificial potential field with adaptive multi-speed scheduler. *IET Intelligent Transport Systems*, *14*(10), 1200–1209.

Wang, P., Gao, S., Li, L., Sun, B. & Cheng, S. (2019) Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm. *Energies*, *12*(12), 2342.

Xie, S., Hu, J., Ding, Z. & Arvin, F. (2021) Collaborative overtaking of multi-vehicle systems in dynamic environments: a distributed artificial potential field approach. In *2021 20th International Conference on Advanced Robotics* (ICAR).

Zhu, S., Gelbal, S. Y. & Aksun-Guvenc, B. (2018) Online quintic path planning of minimum curvature variation with application in collision avoidance. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 669–676). Maui, HI, USA: IEEE.

How to cite this article: Malayjerdi, E., Sell, R., Malayjerdi, M., Udal, A. & Bellone, M. (2022) Practical path planning techniques in overtaking for autonomous shuttles. *Journal of Field Robotics*, 39, 410–425.
https://doi.org/10.1002/rob.22057

Appendix 6

VI

Igor Astrov, Andres Udal, Heiko Pikner, and Ehsan Malayjerdi. A model-based LQR control of an obstacle avoidance maneuver of a self-driving car. In 2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI):473–478. IEEE, 2022

A Model-Based LQR Control of an Obstacle Avoidance Maneuver of a Self-Driving Car

Igor Astrov[1], Andres Udal[2], Heiko Pikner[3], and Ehsan Malayjerdi[3]

[1] *Finest Twins Smart City Center of Excellence*
[2] *Department of Software Sciences*
[3] *Department of Mechanical and Industrial Engineering*
Tallinn University of Technology
Tallinn, Estonia
Igor.astrov@ieee.org, andres.udal@taltech.ee, heiko.pikner@taltech.ee, ehsan.malayjerdi@taltech.ee

***Abstract*—The safe and effective computer control of different types of autonomous vehicles is becoming an increasingly important task in modern Cyber-Physical Systems (CPS) and in future transportation solutions of the Smart Cities. Paper discusses the linearized mechanical model and Linear Quadratic Regulator (LQR) control mathematics of a Self-Driving Car (SDC) of 1200 kg weight class. We demonstrate in the Simulink/MATLAB environment the applicability of this cost functions based advanced optimized control approach for reliable performing of a two-step obstacle avoidance maneuver. Study is related to improvement of control algorithms of a real autonomous shuttle ISEAUTO operating in the campus of Tallinn University of Technology.**

Keywords—autonomous vehicles, bicycle model, LQR control, self-driving cars, MATLAB simulation, smart city.

I. Introduction

The self-driving autonomous vehicles (AVs) are becoming a very important part of different types of modern cyber-physical systems (CPSs) and future transportation paradigms of world economy in general [1]. To achieve mandatory high level of safety at operational level, the software solutions of AVs need both adequate mathematical models of AVs and reliable control algorithms that could outperform the driving skills of human drivers of traditional vehicles [2].

In Tallinn University of Technology, one of priority R&D fields during recent years has been the development of the self-driving cars and city shuttle minibuses, e.g. [3-5]. Modelling and simulation of various airborne and waterborne (both underwater and surface) AVs has been a noticeable research theme as well, e.g. [6-9]. Those activities have been recently integrated with a wider technical and social development of Tallinn-Helsinki twin Smart City ideas [10, 11].

To achieve autonomous driving both in tactical and operational level, several hardware-software subsystems must function in organized manner [3-5]. First the sensing subsystem (GPS, IMU, encoders, cameras, LIDAR, etc.) should collects available vehicle and environment information. Using this information, the trajectory planning software module should generate the desired route (positions, orientations and vehicle velocities) using the actual vehicle position and velocity data. Finally, the automatic control should generate the control actions (acceleration, steering and braking) to realize the route at operational level.

The automatic control one of the most important subtasks as it is responsible for the safe execution of desired maneuvers. For that reason, the use of mathematical kinematic and dynamic vehicle models is necessary.

Thus the motion planning and control problems are two different but highly related tasks. The first consists of computing a feasible trajectory (in terms of vehicle's dynamics) for the vehicle considering the surrounding obstacles such as other vehicles, pedestrians and nonmoving objects. The second is acting on the actuators, i.e. the gas pedal, brake pedal and steering wheel, in order to track the trajectory obtained by the motion planner, while ensuring the stability of the system and, if possible, a smooth drive.

Moreover, the level of abstraction also differs: dealing with obstacles is one of the main task of the motion planner while the controller usually completely ignores them. The trajectory given by the motion planner is assumed to be safe within a certain margin and the task of the controller is to follow as well as possible the given trajectory without considering the obstacles. Failure of the controller to follow the safe trajectories will endanger the whole system.

The level of difficulty of a vehicle guidance control problem is in general lines determined by two aspects: the type of control (lateral, longitudinal or mixed) and the complexity of the model to be controlled (kinematic, linear dynamic, nonlinear simplified dynamic or non-linear dynamic). In this paper we address one of the most complex configurations - the non-linear dynamic problem. Here we offer and simulate a specially designed control system with Linear Quadratic Regulator (LQR) for the linearized model that takes effectively into account the original nonlinear dynamical model of a Self-Driving Car (SDC). The simulation results demonstrate a good performance of lateral y coordinate stabilization task In addition, this LQR approach, realized with standard tools of Simulink/MATLAB, is demonstrating low computational costs thus making possible the real-time applications.

II. Nonlinear Model of SDC

In order to fulfil the requirement of accuracy while maintaining the model complexity and computational costs within reasonable limits, the well-known dynamic bicycle model has been used to model the SDC. By using the position coordinates in the global coordinate system x, y, the yaw angle (heading angle) θ, the vehicle velocities in vehicle's coordinate system v_x, v_y, and the yaw rate ω as the states, the most important parts of the lateral and longitudinal dynamics may be described, see Fig. 1.

The bicycle model is a simplification of the four-wheel vehicle configuration where the two front and the two rear wheels are replaced by a single (mass-less) front and rear wheel that are located at the longitudinal axis of the car, see Fig. 1. This 3 DOF (degree of freedom) planar model is capable of rendering the planar longitudinal, lateral and yaw dynamics of the vehicle. The pitch and roll dynamics of a vehicle are considered to be negligible in this model.

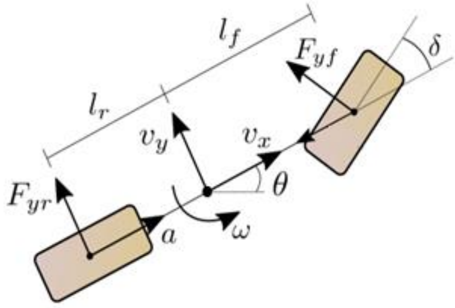

Fig. 1. Bicycle model of a car [12]. Model is based on assumption that the pairs of rear and forward wheels can be replaced by single wheels.

The kinematics of this system can be presented as [13]

$$\dot{x} = v_x \cos\theta - v_y \sin\theta \tag{1}$$

$$\dot{y} = v_x \sin\theta + v_y \cos\theta \tag{2}$$

$$\dot{\theta} = \omega \tag{3}$$

where (x, y) denote the coordinates of the center of mass of the vehicle in the earth-fixed frame, θ is the yaw angle (i.e. heading angle) of the vehicle, the velocities v_x, v_y denote the longitudinal and lateral speeds in the body frame, and ω denotes the yaw rate (i.e. rotation rate).

The dynamics of the analyzed SDC can be presented by the following equations [12], [14]

$$m\dot{v}_x = F_x + m v_y \omega \tag{4}$$

$$m\dot{v}_y = -m v_x \omega + 2(F_{yf} \cos\delta + F_{yr}) \tag{5}$$

$$I\dot{\omega} = 2(l_f F_{yf} \cos\delta - l_r F_{yr}) \tag{6}$$

where m and I denote vehicle's mass and yaw inertia, respectively; F_{yf} and F_{yr} denote the lateral tire forces acting on the front and rear wheels, respectively (in coordinate frames aligned with the wheels), F_x is the driving force of the vehicle, δ is the front steering angle, l_f and l_r are the distances from the center of gravity to the front and rear wheel axes, respectively. Note that the forces acting to the wheels are multiplied by factor of two to take into account the use of one wheel instead of two.

In case of longitudinal acceleration a the driving force of the vehicle F_x in (4) is calculated as

$$F_x = ma \tag{7}$$

where m is a mass of vehicle.

The lateral tire forces F_{yf} and F_{yr} in (5)-(6) acting on the front and rear wheels can be expressed as [15]

$$F_{yf} = C_f \left(\delta - \tan^{-1}\left(\frac{v_y + l_f\dot{\theta}}{v_x}\right)\right) \tag{8}$$

$$F_{yr} = -C_r \tan^{-1}\left(\frac{v_y - l_r\dot{\theta}}{v_x}\right) \tag{9}$$

where the constants C_f and C_r are the forward and rear tire stiffness parameters taking into account the difference of real turning trajectories from idealized trajectories of free rolling.

Next, to complete the task specification, the integration for the center of mass coordinates and yaw angle must be used:

$$x(\tau) = \int_0^\tau \dot{x}(t)\,dt, y(\tau) = \int_0^\tau \dot{y}(t)\,dt, \theta(\tau) = \int_0^\tau \dot{\theta}(t)\,dt, \tag{10}$$

where initial state $x(0) = 0, y(0) = 0, \theta(0) = 0$.

III. Parameters of Nonlinear Model of SDC

It is known that some parameters of SDC may strongly depend from conditions of load and environment. Any changes in these parameters can less or more significantly change the behavior of the system. In this section, the parameters of SDC are evaluated for the shuttle ISEAUTO, realized on the basis of Mitsubishi i-MiEV electric car [16].

The following parameters of this vehicle can be obtained by direct measurements

$$m = 1160 \text{ kg}, l_f = 1.275 \text{ m}, l_r = 1.275 \text{ m} .$$

The moment of inertia I around z -axis for parallelepiped is calculated as follows [17]:

$$I = \frac{m}{12}\left(l_x^2 + l_y^2\right) . \tag{11}$$

Hence, assuming that length $l_x = 3.6$ m and width $l_y = 1.5\ m$, from (11), we find for yaw inertia moment

$$I = 1470.3 \text{ kg m}^2 .$$

Let us now evaluate the values of C_f and C_r.

Following a straight track with a small yaw angle θ at steady state, we can state estimations

$$v_x \approx const, v_y \approx 0 . \tag{12}$$

In idealized conditions the yaw rate can be related to steering angle as [18]

$$\dot{\theta} = \frac{v_x}{(l_f + l_r)} \tan\delta . \tag{13}$$

Combining (8)-(9) and (12)-(13), we obtain

$$F_{yf} = C_f \left(\delta - \tan^{-1}\left(\frac{l_f \tan\delta}{l_f + l_r}\right)\right) \tag{14}$$

$$F_{yr} = C_r \tan^{-1}\left(\frac{l_r \tan\delta}{l_f + l_r}\right) . \tag{15}$$

At that the next geometrical inequalities hold

$$0 < \frac{l_f}{l_f + l_r} < 1 , \tag{16}$$

$$0 < \frac{l_r}{l_f + l_r} < 1 . \tag{17}$$

Following a straight track with a small steering angle δ at steady state, and using the small angle approximation $\tan^{-1}(\tan x) = x$, we obtain by combining (14)-(17)

$$F_{yf} = \frac{C_f l_r \delta}{l_f + l_r} , \tag{18}$$

$$F_{yr} = \frac{C_r l_r \delta}{l_f + l_r} . \tag{19}$$

Further, for a straight track with a small constant yaw angle θ and small steering angle δ at steady state, we can use

$$\cos\delta \approx 1, \quad \dot{\omega} \approx 0 . \tag{20}$$

Combining (6) and (20), we obtain

$$l_f F_{yf} \approx l_r F_{yr} . \tag{21}$$

Similarly, combining (18)-(19) and (21), we obtain

$$C_r \approx C_f\ (l_f/l_r)\ . \tag{22}$$

Further, assuming a straight track with a small steering angle δ and slow changing v_y, we can use

$$\cos\delta \approx 1,\ \dot{v}_y \approx 0 \tag{23}$$

that simplifies the lateral motion equation (5) to

$$m v_x \omega \approx 2F_{yf} + 2F_{yr}\ . \tag{24}$$

Combining (3), (13), (18), (19), (22) and (24) and assuming $\tan\delta \approx \delta$ at steady state, we come to an assessment

$$C_f \approx \frac{m v_x^2}{2(l_f + l_r)}\ . \tag{25}$$

At that should be noted that modeling of vehicle-road systems is complex because the vehicle trajectory depends on a wide variety of tire, car and road parameters. For example, when road friction changes or when the nonlinear tire domain is reached, the wheel cornering stiffness may vary from 50% to 150% of estimated value [19]. The maximum value of stiffness coefficient C_f may be estimated from (25) as:

$$C_{fmax} \approx \frac{m v_{xmax}^2}{2(l_f + l_r)} \tag{26}$$

where v_{xmax} is a maximum value of velocity v_x.

Finally, using (22), (26) and $v_{xmax} = 50\ \mathrm{km/h}$, we obtain the tire stiffness coefficient estimations

$$C_f = 43875\ \mathrm{N/rad},\ C_r = 43875\ \mathrm{N/rad}\ .$$

IV. State-Space Model of SDC

In order to apply the LQR-controller approach, at first the linearized model of object control with conventional system and input matrices *A* and *B* and state, output and input vectors *X*, *Y* and *U* should be formulated.

Let us first consider the definition for velocity v_x that we handle as a separate directly specified parameter in the present study. By dividing the two parts of (4) to m and by taking into account (7), we find

$$\dot{v}_x = a + v_y \omega\ . \tag{27}$$

From where, in case of small values v_y, ω the further simplification follows:

$$\dot{v}_x \approx a\ . \tag{28}$$

Alternatively, if we consider constant speed case with zero a, then

$$v_x = V_c \tag{29}$$

where V_c is a fixed constant (given speed).

Let us now consider the linearization of equations (1)-(2) through the expansion of nonlinear functions in Taylor series with the first linear terms in the vicinity of point 0. Hence, from (1)-(2) and (29), we have

$$\dot{x} = V_c - v_y \theta\ , \tag{30}$$

$$\dot{y} = V_c \theta + v_y\ . \tag{31}$$

The lateral tire forces F_{yf} and F_{yr} at the front and rear wheels in (5)-(6) can be expressed as [12]

$$F_{yf} = C_f\left(\delta - \frac{v_y + l_f \omega}{v_x}\right)\ , \tag{32}$$

$$F_{yr} = -C_r\left(\frac{v_y - l_r \omega}{v_x}\right)\ . \tag{33}$$

Combining (29) and (32)-(33), we obtain

$$F_{yf} = C_f \delta - \frac{C_f v_y}{V_c} - \frac{C_f l_f \omega}{V_c}\ , \tag{34}$$

$$F_{yr} = -\frac{C_r v_y}{V_c} + \frac{C_r l_r \omega}{V_c}\ . \tag{35}$$

By dividing of two parts of (5) to m, assuming that $\cos\delta \approx 1$ at steady state, and using (34)-(35). we find

$$\dot{v}_y = \frac{2C_f}{m}\delta - \frac{2(C_f + C_r)}{m V_c} v_y + \left(\frac{2(C_r l_r - C_f l_f)}{m V_c} - V_c\right)\omega\ . \tag{36}$$

Similarly, by dividing of two parts of (6) to I, assuming that $\cos\delta \approx 1$ at steady state, and using (34)-(35). we find

$$\dot{\omega} = \frac{2C_f l_f}{I}\delta + \frac{2(C_r l_r - C_f l_f)}{I V_c} v_y - \left(\frac{2\left(C_f l_f^2 + C_r l_r^2\right)}{I V_c}\right)\omega\ . \tag{37}$$

The conventional matrix formulation of state-space model can be expressed as

$$\dot{X} = AX + BU\ , \tag{38}$$

$$Y = CX + DU\ . \tag{39}$$

At that the state and input (control) vectors are denoted as

$$X = \begin{bmatrix} y \\ v_y \\ \omega \\ \theta \end{bmatrix},\ U = \delta \tag{40}$$

and the system matrix A in (38) can be formulated as

$$A = \begin{bmatrix} 0 & 1 & 0 & V_c \\ 0 & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{41}$$

with the elements of matrix obtained from (36)-(37) as

$$a_{22} = -\frac{2(C_f + C_r)}{m V_c},\ a_{23} = \frac{2(C_r l_r - C_f l_f)}{m V_c} - V_c\ ,$$

$$a_{32} = \frac{2(C_r l_r - C_f l_f)}{I V_c},\ a_{33} = -\frac{2\left(C_f l_f^2 + C_r l_r^2\right)}{I V_c}\ .$$

The input matrix B in (38) can be described as

$$B = \begin{bmatrix} 0 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \tag{42}$$

where the elements are obtained from (36)-(37) as

$$b_2 = \frac{2C_f}{m},\ b_3 = \frac{2C_f l_f}{I}\ .$$

To complete the state-space presentation, the unit output matrix C (equal output and state vectors) and zero feedthrough matrix D in (39) may be specified:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\ . \tag{43}$$

V. The Control System

In optimal control one attempts to find a controller that provides the best performance with respect to some given cost function. When the mathematical model of the controlled object is linear and the functions that appear in the cost function have a quadratic form, we have a problem of continuous-time LQR optimal control.

Considering the linear model stationary, stabilizable and detectable, the problem of LQR regulation in infinite time is to calculate the optimal feedback gain matrix K such that the feedback control law

$$U = -KX \quad (44)$$

minimizes the cost function

$$J = \int (X^T Q X + U^T R U) dt \quad (45)$$

which was subjected to the constraint equation (38).

The matrices Q, R in (45) can be chosen by applying the following rule [20]:

$$Q_{ii} = \frac{1}{x_{i\,max}^2} \quad (46)$$

$$R_{ii} = \frac{1}{u_{i\,max}^2} \quad (47)$$

where $x_{i\,max}$ and $u_{i\,max}$ denote the maximum acceptable values for output and input signals, respectively.

Combining (39), (40), (43), and (46)-(47), we obtain

$$Q = \begin{bmatrix} \frac{1}{y_{max}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{v_{y\,max}^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\omega_{max}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\theta_{max}^2} \end{bmatrix}$$

and

$$R = \left[\frac{1}{\delta_{max}^2}\right]$$

where for simulation example the following acceptable parameter values were used: $y_{max} = 5$ m, $v_{y\,max} = 1$ km/h, $\omega_{max} = V_c/(l_r + l_f) = 1.0893$ rad/s , $\theta_{max} = \pi/2$ rad , $\delta_{max} = \pi/4$ rad.

VI. Simulation Results

Below the simulation results of the SDC are discussed for a trajectory consisting of sequential movement towards two desired lines to avoid an obstacle. The Simulink-style block-scheme for optimal LQR control is presented in Fig. 2. The constant velocity condition (29) was applied during this maneuver.

Note that the absolute velocity of vehicle with the direction of motion $\theta + \tan^{-1}\left(\frac{v_y}{V_c}\right)$ is defined as

$$v = \sqrt{V_c^2 + v_y^2} \ .$$

General equation for uniform motion along an axis x with constant velocity $v_x = V_c$ can be expressed simply as

$$x(t) = V_c t \quad (48)$$

where t is a current instant of time.

Let us consider a trajectory that can be divided in time to two intervals $(0, \tau_1)$ and (τ_1, τ_2) where the first interval is for achieving of the first reference output y_{ref1}, and the second interval is for achieving of second reference output y_{ref2} . Hence, considering (48) and defined time instants, we find

$$x_{sd} \approx V_c \tau_1 \quad (49)$$

where x_{sd} may be called safety distance for this maneuver by coordinate x.

The prohibited area for movement can thus be evaluated as a rectangle with a length $l_{rx} \approx x_{sd}$ from (49) and a width $l_{ry} \approx y_{ref1} - y_{ref2}$.

In control task the plant input signal consists of the steering angle of front wheels δ. The output signal vector consist of the coordinate y, the lateral speed v_y, the yaw rate ω, and the yaw angle θ.

For simulation experiment the following parameters were used: the velocity $v_x = 10\,\frac{\text{km}}{\text{h}}$ from (29); the y coordinate references $y_{ref1} = 5$ m, $y_{ref2} = 1$ m, the setting time $\tau_1 =$ 25 s for (49) and the gain matrix from MATLAB lqr-function for (44) $K = [0.1571 \quad 2.1344 \quad 0.1246 \quad 1.6723]$. Note that x_{sd} from (49) for this case equals approximately 69 m.

The input and output signals of this simulation experiment are explained by Fig. 3. The state parameters - velocity and yaw angle for are shown in Fig. 4. The resulting trajectory of SDC is shown in Fig. 5. One should notice smooth transition to desired lines y_{ref1}, y_{ref2} with negligible overshoot 2-3% of output signal.

A screen view of the illustrative video of SCC movement generated by MATLAB is shown in Fig. 6. The "Simulation 3D Vehicle with Ground Following" block (see Fig. 2) implements a 4-wheel car in 3D simulation environment. This block of animation uses the (x, y) position and yaw angle θ of the vehicle to adjust the elevation, roll angle and pitch angle of the vehicle so that it follows the ground terrain. The "Simulation 3D Scene Configuration" block implements a 3D simulation environment so that user can see the world around the vehicle and virtually test the functioning of control system for given model of SDC. .

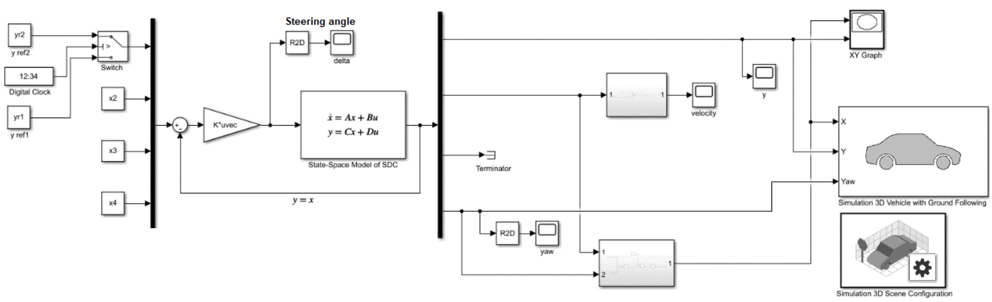

Fig. 2. Simulink-style block diagram of control system with LQR controller (triangular schematic element) for linearized SDC model.

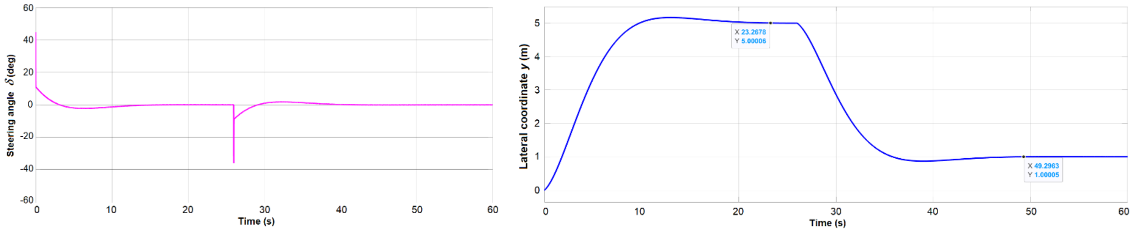

Fig. 3. Input and output values of control: steering angle (left) and coordinate y (right).

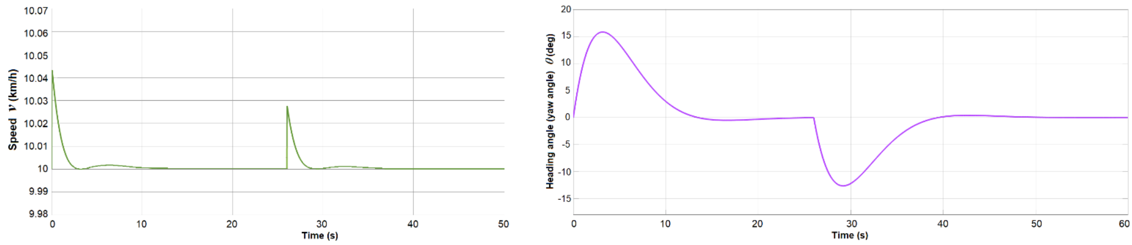

Fig. 4. State values of the controlled object: velocity (left) and yaw angle (right).

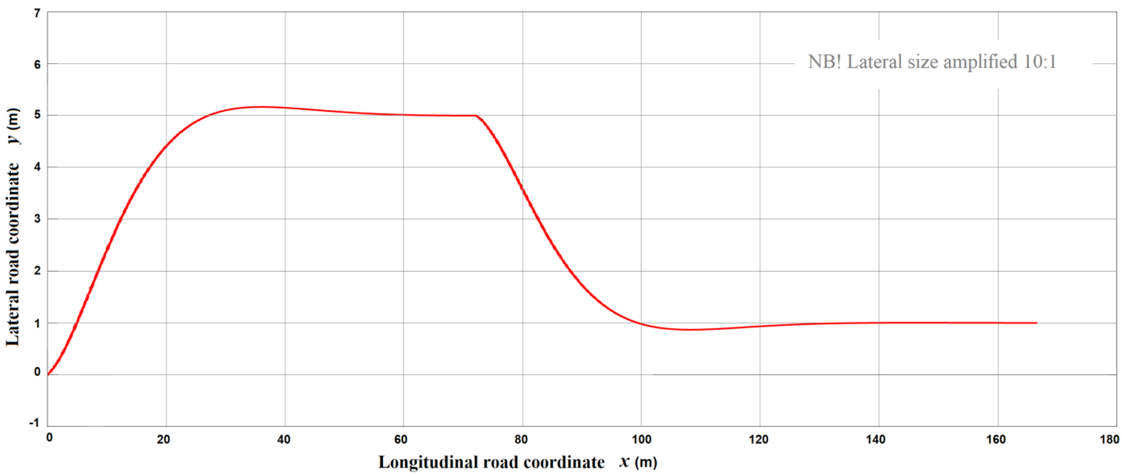

Fig. 5. Two stages of calculated trajectory: left shift 5 m from initial lane $y = 0$ and return to right lane $y = 1$ m. Note the enlarged lateral coordinate scale.

Fig. 6. Visual presentation of SDC motion in Simulink/MATLAB environment by using software tools “Simulation 3D Vehicle with Ground Following” and “Simulation 3D Scene Configuration”.

VII. Conclusions

In this paper we described a modeling and simulation technique of a self-driving car of 1200 kg weight class. Actual application task was the 2-stage (overtake and return) obstacle avoidance maneuver at constant speed with controlling of the steering angle. For this maneuver the applicability of LQR (Linear Quadratic Regulator) controller was demonstrated via numerical simulations in Simulink/MATLAB environment. Application of LQR controller is a promising approach as it is optimal by default and it allows to take into account the cost of different characteristics of the vehicle movement.

Obtained simulation results demonstrate smooth and reliable performance of the LQR controller in achieving of lane changes with small 2-3% overshoot.

Further applications of the described car model versions, both nonlinear and linear, and LQR type controller are associated with the development of autonomous shuttle ISEAUTO minibuses for use in the campus of Tallinn University of Technology [3,4] and in development of smart city future mobility environment [10,11]. One particular real application of the LQR controller may be the minimization of the steering engine power usage in maneuvering of the autonomous shuttles.

Acknowledgment

This work was supported via funding by two grants: the European Union's Horizon 2020 Research and Innovation Programme grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, grant No. 2014-2020.4.01.20-0289.

References

[1] V. Tsiatsis, S. Karnouskos, J. Holler, D. Boyle, and C. Mulligan, “Autonomous vehicles and systems of cyber-physical systems,” in Internet of Things: Technologies and Applications for a New Age of Intelligence, 2nd ed., London: Academic Press, 2019, pp. 209–305.

[2] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” Human Factors, vol. 37, March 1995, pp. 32–64.

[3] R. Wang, R. Sell, A, Rassõlkin, T. Otto, and E. Malayjerdi, “Intelligent functions development on autonomous electric vehicle platform,” Journal of Machine Engineering, vol. 20, no. 2, pp. 114−125, 2020.

[4] R. Sell, M. Leier, A. Rassõlkin, and J.-P. Ernits, “Autonomous last mile shuttle ISEAUTO for education and research,” Int. Journal of Artificial Intelligence and Machine Learning, vol. 10, no. 1, pp. 18−30, 2020.

[5] R. Sell, A. Rassõlkin, R, Wang, and T. Otto, “Integration of autonomous vehicles and Industry 4.0,” Proc. Estonian Academy of Sciences, vol. 68, no. 4, pp. 389−394, 2019.

[6] A. Pedai, I. Astrov, and A. Udal, “Comparison of single-rate and two-rate neural control approaches for coaxial rotor/ducted-fan TUAV for situational awareness applications,” Proc. IEEE 16th Int. Symposium on Intelligent Systems and Informatics, SISY 2018, Subotica, Serbia, September 13-15, pp. 63–68, 2018.

[7] A. Pedai, I. Astrov, A. Udal, and R. Sell, “Single-rate versus three-rate neural assisted control approaches for coaxial rotor ducted fan TUAV for situation awareness applications,” Proc. 13th Annual IEEE Int. Systems Conference, SysCon 2019, Orlando, USA, April 8-11, pp. 457−463, 2019.

[8] I. Astrov and A. Udal, “Neural predictive tracking control of catamaran model sailboat for situation awareness applications,” Proc. IEEE 24th Int. Conference on Intelligent Engineering Systems, INES 2020, Reykjavik, Iceland, July 8-10, pp. 153−158, 2020.

[9] I. Astrov, A. Udal, A. Pedai, and R. Sell, “Simulink/MATLAB based comparison of neural and basic tracking control for an autonomous surface vessel for situation awareness applications,” Proc. IEEE Joint 19th Int. Symposium on Computational Intelligence and Informatics and 7th IEEE Int. Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics, CINTI-MACRo 2019, Szeged, Hungary, November 14–16, pp. 105–110, 2019.

[10] M. Bellone, A. Ismailogullari, T. Kantala, S. Mäkinen, R.-M. Soe, and M. Kyyrö, “A cross-country comparison of user experience of public autonomous transport,” European Transport Research Review, vol.13, Article # 19, 2021.

[11] R. Sell, R.-M- Soe, R. Wang, and A. Rassõlkin, “Autonomous vehicle shuttle in Smart City testbed,” In: Zachäus C., Meyer G. (eds) Intelligent System Solutions for Auto Mobility and Beyond. AMAA 2020. Lecture Notes in Mobility. Springer, Cham., pp. 143–157, 2020.

[12] E. Alcalá, V. Puig, and J. Quevedo, “LPV-MPC control for autonomous vehicles,” Proc. 3rd IFAC Workshop on Linear Parameter Varying Systems, LPVS 2019, Eindhoven, Netherlands, November 4–6, pp. 106–113, 2019.

[13] M. G. Plessen, D. Bernardini, H. Esen, and A. Bemporad, “Spatial-based predictive control and geometric corridor planning for adaptive cruise control coupled with obstacle avoidance,” IEEE Trans. Control Systems Technology, vol. 26, no. 1, pp. 38–50, 2018.

[14] K. Yeom, “Kinematic and dynamic controller design for autonomous driving of car-like mobile robot,” Int. Journal of Mechanical Engineering and Robotics Research, vol. 9, no.7, pp. 1058–1064, 2020.

[15] P. Polack, F. Altché, B. Novel, and A. Fortelle, “The kinematic bicycle model: a consistent model for planning feasible trajectories for autonomous vehicles?,” Proc. 2017 IEEE Intelligent Vehicles Symposium, IVS 2017, Los Angeles, USA, June 11-14, pp. 1–7, 2017.

[16] Iseauto. [Online]. Available from: https://www.iseauto.taltech.ee [Accessed on 10 August 2021].

[17] Parallelepiped rotating around its central axis. [Online]. Available from: http://www.claredot.net/en/sec-MomMasInertia/Mom_11.php [Accessed on 10 August 2021].

[18] J. A. Matute, M. Marcano, S. Diaz, and J. Perez, “Experimental validation of a kinematic bicycle model predictive control with lateral acceleration consideration,” Proc. 10th IFAC Symposium on Intelligent Autonomous Vehicles, IAV 2019, Gdansk, Poland, July 3–5, pp. 289–294, 2019.

[19] G. Baffet, A. Charara, and D. Lechner, “Estimation of vehicle sideslip, tire force and wheel cornering stiffness,” Control Engineering Practice, vol. 17, November 2009, pp. 1255–1264.

[20] M. C. Simone and D. Guida, “Identification and control of a unmanned ground vehicle by using Arduino,” UPB Scientific Bulletin, Series D: Mechanical Engineering, vol. 80, January 2018, pp.141–154.

Curriculum Vitae

1. Personal data

Name	Ehsan Malayjerdi
Date and place of birth	1 September 1978 Yazd, Iran
Nationality	Iran

2. Contact information

Address	Kadaka tee 141, 19086 Tallinn, Estonia
Phone	+372 59172888
E-mail	ehsan.malayjerdi@taltech.ee

3. Education

2018–2022	Tallinn University of Technology, School of Engineering, Mechanical Engineering, PhD studies
2014–2016	Azad University of Mashhad, Faculty of Computer Engineering, Artificial Intelligence, MSc
2000–2006	Azad University of Yazd, Faculty of Computer Engineering, Computer Engineering, BSc

4. Language competence

Persian	native
English	fluent
Estonian	basic

5. Professional employment

2021– …	Tallinn University of Technology, Early stage researcher
2014–2018	Robotmaker company, Softwarer developer
2014-2017	Computer Graphic Society (CG Society), software developer
2013-2014	Ministry of ICT, software developer
2007-2013	IRI Post Company, software developer

6. Field of research

- Autonomous vehicles

Elulookirjeldus

1. Isikuandmed

Nimi	Ehsan Malayjerdi
Sünniaeg ja -koht	01.09.1978, Yazd, Iraan
Kodakondsus	Iraani

2. Kontaktandmed

Aadress	Kadaka tee 141, 19086 Tallinn, Estonia
Telefon	+372 59172888
E-post	ehsan.malayjerdi@taltech.ee

3. Haridus

2018–2022	Tallinna Tehnikaülikool, PhD
2014–2016	Azad University of Mashhad, Arvutitehnika teaduskond, Tehisintellekt, MSc
2000–2006	Azad University of Yazd, Arvutitehnika teaduskond, arvutitehnika, BSc

4. Keelteoskus

Pärsia	emakeel
Inglise	kõrgtase
Eesti	algtase

5. Teenistuskäik

2021– ...	Tallinna Tehnikaülikool, nooremteadur
2014–2018	Robotmaker ettevõte, tarkvaraarendaja
2014-2017	Computer Graphic Society (CG Society), tarkvaraarendaja
2013-2014	IKT-ministeerium, tarkvaraarendaja
2007-2013	IRI Post Company, tarkvaraarendaja

6. Teadustöö põhisuunad

- Autonoomsed sõidukid

ISSN 2585-6901 (PDF)
ISBN 978-9949-83-882-0 (PDF)