TAL TECH

DOCTORAL THESIS

Cyber-Physical Control System for Self-Driving Vehicles

Heiko Pikner

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2024

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
44/2024

Cyber-Physical Control System for Self-Driving Vehicles

HEIKO PIKNER

TAL TECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Mechanical and Industrial Engineering

The dissertation was accepted for the defence of the degree of Doctor of Philosophy on 10 July 2024

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

Co-supervisor: Professor Kristo Karjust,
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia

Opponents: Dr. Ernő Horváth,
Scientific leader of the Autonomous Transport Systems Center
University of Györ
Györ, Hungary

Professor Simona Ramanauskaitė,
Department of Information Technology
Vilnius Gediminas Technical University
Vilnius, Lithuania

Defence of the thesis: 03 September 2024, 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.

Heiko Pikner _____
signature

ISSN 2585-6898 (publication)
ISBN 978-9916-80-186-4 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-187-1 (PDF)
DOI https://doi.org/10.23658/taltech.44/2024
Printed by Koopia Niini & Rauam

Pikner, H. (2024). *Cyber-Physical Control System for Self-Driving Vehicles* [TalTech Press]. https://doi.org/10.23658/taltech.44/2024

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
44/2024

Isejuhtivate sõidukite küberfüüsikaline juhtsüsteem

HEIKO PIKNER

TAL TECH
KIRJASTUS

Contents

List of Publications

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

I **H. Pikner**, R. Sell, K. Karjust, E. Malayjerdi, and T. Velsker, "Cyber-physical control system for autonomous logistic robot," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pp. 699–704, IEEE, Apr. 2021

II **H. Pikner**, R. Sell, J. Majak, and K. Karjust, "Safety system assessment case study of automated vehicle shuttle," *Electronics*, vol. 11, p. 1162, Apr. 2022

III **H. Pikner**, R. Sell, and E. Malayjerdi, "Level 4 commercial autonomous vehicle control system transition to an open-source solution," *Proc. Eston. Acad. Sci.*, vol. 73, no. 2, pp. 124–133, 2024

IV **H. Pikner**, M. Malayjerdi, M. Bellone, B. Baykara, and R. Sell, "Autonomous driving validation and verification using digital twins," in *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, pp. 204–211, INSTICC, SciTePress, 2024

Author's Contributions to the Publications

I The author designed and constructed an entire low-level control architecture for the autonomous mobile robot, implemented an experimental validation methodology in cooperation with a Kulinaaria production plant, and contributed to the analysis.

II The author collaborates with the risk evaluation model development. Based on the developed risk evaluation model and key automotive standards, the author developed a methodology for the new safety architecture of the TalTech iseAuto.

III The author proposed a novel approach for transferring low-level control systems with distinct electronics and mechanical specifications from one autonomous vehicle to another. After integrating the critical control systems responsible for steering, accelerating, and braking into the target shuttle, the author validated the autonomous shuttle's reliability and safety with a series of experiments. The author analyzed the results, discussed them, and drew conclusions for registering the target autonomous vehicle as a legal vehicle on the roads in Estonia.

IV The author proposed the initial idea to expand the high-fidelity simulation to a simulated low-level control system. Furthermore, the author developed the concept of a hardware in-loop simulation environment when a vehicle self-drives inside a simulation and simultaneously generates all the traffic on the data network. Based on the evaluation, the author suggested testing scenarios that would be too hazardous to conduct in real traffic scenarios, which led to further research.

Abbreviations

ABS	Anti-Lock Braking System
AD	Autonomous Driving
ADAS	Advanced Driver Assistance System
AEC	Automotive Electronics Council
AHP	Analytical Hierarchy Process
AI	Artificial Intelligence
ALKS	Automated Lane-Keeping Systems
AMR	Autonomous Mobile Robot
API	Application Programming Interface
ASIL	Automotive Safety Integrity Level
AUTOSAR	Automotive Open System Architecture
AV	Autonomous Vehicle
BLDC	Brushless DC
BSP	Board Support Package
CAN	Controller Area Network
CPS	Cyber-Physical System
C&C	Communication and Collaboration
DBC	Database CAN file
DC	Direct Current
DT	Digital Twin
DOE	Design-Of-Experiment
DUT	Device Under Test
ECU	Electronic Control Unit
EPS	Electric Power Steering
ESC	Electronic Stability Control
FAHP	Fuzzy Analytical Hierarchy Process
FTA	Fault Tree Analysis
GPIO	General-Purpose Input and Output
GPS	Global Positioning System
HARA	Hazard Analysis and Risk Assessment
HIL	Hardware-In-the-Loop
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
ISO	International Organization for Standardization
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LiDAR	Light Detection And Ranging
LIN	Local Interconnect Network
MBD	Model-Based Design
MIL	Model-In-the-Loop
MCDM	Multi-Criteria Decision-Making
NASA	National Aeronautics and Space Administration
NIS	Negative Ideal Solution
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer

PID	Proportional Integral Derivative
PIS	Positive Ideal Solution
PWM	Pulse Width Modulation
RH	Research Hypotheses
ROS	Robot Operating System
RT	Research Task
RTOS	Real-Time Operating System
SAE	Society of Automotive Engineers
SIL	Software-In-the-Loop
SOTIF	Road vehicle's safety of the intended functionality
PCB	Printed Circuit Board
PIL	Processor-In-the-Loop
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
UDP	User Datagram Protocol
UNECE	United Nations Economic Commission for Europe
U.S.	United States
VIKOR	Vlse Kriterijumska Optimizacija I Kompromisno Resenje
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V&V	Verification and Validation
XIL	X-In-the-Loop

1 Introduction

The thesis is divided into five main topics and sections. Chapter 1 introduces the background, describes the motivation, and outlines the goals of the thesis. Subsequent chapters define the research problem and objectives, present the research hypotheses, outline research tasks, and finally address the scientific and practical novelties of the research. Chapter 2 discusses the methods that form the low-level control system design, validation, and verification framework. Chapter 3 addresses case studies to evaluate the methodology's performance under actual conditions developed in Chapter 2. Finally, Chapters 4 and 5 conclude all the results gathered during this research and recommend future research topics.

1.1 An Overview of the Nature of the Research Problem

In recent times, the advancement of Autonomous Vehicles (AVs) has sparked hope for a genuinely driverless society and promised enhanced transportation efficiency, traffic safety, and energy conservation. These AVs, including self-driving cars and small-scale Autonomous Mobile Robots (AMRs), are not just technological developments but paradigm-changing innovations reshaping various industries, from automotive to logistics, mining, and machinery. Over the past century, the automobile has evolved into a primary mode of transportation, fueling the exponential growth of the automotive industry. Its ability to mass-produce safe, reliable, and affordable vehicles has been vital to this expansion [1].

The emergence of AVs can be traced back to the early years of the new millennium. A significant milestone occurred in 1998 when the ARGO vehicle, a precursor to modern AVs, achieved a remarkable feat. It completed an Autonomous Driving (AD) test over 2000 km on an Italian highway, marking the inception of driverless vehicle technology and setting the stage for future advancements [2]. The modern concept of AVs, which centers on using sensors to perceive their surroundings and computer technologies to make informed decisions, was first demonstrated during the DARPA Grand and Urban Challenges from 2005 to 2007 [3].

Traditionally, vehicles have relied on skilled drivers to navigate from one location to another. However, technological advancements have been introduced in a transformative era, converting conventional and mechanical modes of transport into intelligent and information-rich vehicles. Electronics and software integration has given rise to Advanced Driver Assistance Systems (ADAS) [4, 5], one of the most successful and widely adopted technologies in commercial vehicles. Its primary function is to offer speed control. ADAS may additionally provide multiple basic assisted features like collision avoidance, driver potential obstacles alert, lane departure signal, lane centering aid, traffic alerts issuing, automated lighting, or other functionalities. This feature has significantly enhanced safety and convenience for drivers and passengers, saving lives and preventing injuries, thereby underscoring the positive impact of technological advancements in the automotive industry [6].

Since the concepts of AV first appeared in research communities, reliability and safety have always been the focus of AV-related technologies. Implementing a fully autonomous system will not automatically guarantee reduced or eliminated crashes [7]. The "vision zero" mentioned on The European Union's transport roadmap refers to the goal of eliminating traffic fatalities and injuries by 2050, which is a crucial selling point for the AV industry [8]. A study reveals that although AD offers tremendous advantages for individuals and society, several user-related aspects must be addressed before this technological innovation is ready to enter the mass market [9].

In the case of a human driver, the sentience and brain play the roles of sensors and computers in perceiving the environment and making decisions. Controlling the steering wheel and brake/throttle paddles by hands and feet guaranteed the vehicle's safety. For AVs, much research has focused on the perception-decision (sensor-computing) aspect, which commits to a comprehensive understanding of the environment and flawless decision-making. New AD technologies are based on Artificial Intelligence (AI) driven decision-making. AI is applied to specific functions of the AD control algorithm, such as object detection. However, in some perspectives, low-level control systems are more essential to AV's safety than the perception-decision stage. There is little tolerance for mistakes in the AV's critical steering, speed, and brake control systems. Therefore, the failure-proof and accuracy tests of the AV's low-level controlling system are necessary before deploying the vehicles into real traffic.

The following subchapters provide overviews and state-of-the-art information on various aspects of the automotive industry.

- An in-depth exploration of Cyber-Physical Systems (CPSs) explains the fusion of computational elements with physical processes and their implications for automotive innovation.
- The introduction of Industry 4.0 and the emerging paradigms of Industry 5.0 examine impacts on automotive systems and manufacturing practices powered by automotive electronics.
- Key automotive standards govern manufacturing protocols, safety regulations, and performance benchmarks, and their importance cannot be overstated.
- Background information on risk analysis models and methodologies helps validate automotive systems and ensure functionality and safety.
- A detailed overview of automotive electronics highlights the functionalities of electronic components within vehicles.
- The background of the model-based design approach uses models to simulate and analyze the behavior of vehicle systems, improving the design process and reducing development time.
- The background of validating automotive systems outlines the processes to ensure that all components and systems meet the required standards. This includes rigorous testing and analysis to confirm that vehicles are safe and perform as expected under various conditions.

Finally, an overview of the most important companies dealing with autonomous vehicles abroad and in Estonia is provided. Then, a brief specification of TalTech's autonomous shuttles and other state-of-the-art robots showcases their innovative contributions to case studies and validation processes addressed in Chapter 3.

1.1.1 An Overview of Cyber-Physical System (CPS)

Modern vehicle low-level control systems are based on CPS and are responsible for the vehicle's fundamental operations. They enable ADAS and other AD features. CPS represents a critical component of modern automotive technology, allowing vehicles to become more intelligent, responsive, and capable of meeting the evolving demands of mobility.

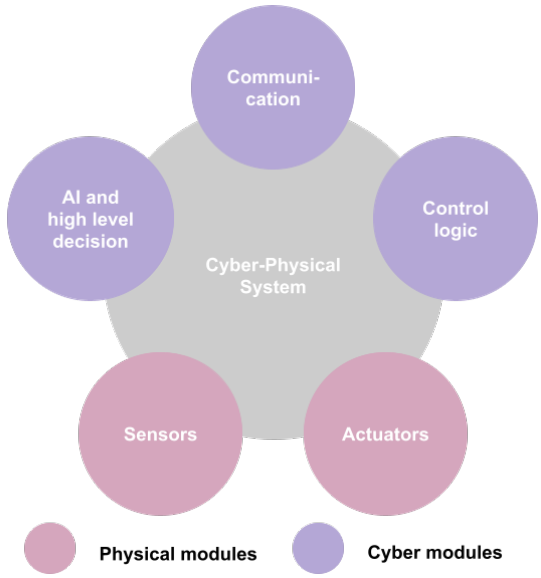

Figure 1: Cyber-Physical System [10].

CPSs are dynamic systems facilitating effective real-time Communication and Collaboration (C&C) among physical components like control systems, sensors, actuators, and computational capabilities [11], as shown in Figure 1. Consequently, AVs are prominent adopters of the CPS paradigm, known as a mobile CPS. Since mobile CPS is more than just a subclass of CPS, it faces specific challenges. It incorporates additional features such as mobility, unstable mobile networks, energy consumption, and highly dynamic environments [12].

The mobile CPS is essential for controlling AVs or AMRs using modules interacting with the physical world. The modules are distributed; therefore, communication over some data bus is important to the system. Collaboration between modules creates some global behavior. Embedded computers connected over networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa [13]. Computational resources can be divided into AI based on high-level decision-making and lower-level control logic.

AI and high-level decision-making are based on some special computers that may run Robot Operating Systems (ROS) [14]. The low-level control logic is near or inside the actuator or sensor modules. It handles a regulation for actuators and makes the first information processing for information received from sensors. Also, it controls and forwards information between the modules. The most common sensors for vision and 3D imaging systems are Light Detection And Ranging (LiDARs) [15], radars, and cameras. There may be sensors for localization, such as Global Positioning System (GPS) and inertial sensors, and sensors that can detect the presence of nearby objects without any physical contact. Such sensors are ultrasound or infrared distance sensors.

Automotive applications demand actuators with a broad power range. Small actuators, such as mirror positioners or door locks, consume less than a watt of electrical power. New by-wire ADAS or AD functions, such as electrical steering, traction control, and braking, require several hundred or even kilowatts. The sole opportunity to achieve the cost and reliability targets for automotive applications featuring such complex control systems

lies in integrating feedback sensors, actuators, embedded control electronics, and communication interfaces into a unified mechanical system [16]. Older Direct Current (DC) motors and newer electronically controlled Brushless DC (BLDC) electric motors seem to remain the dominant actuator choice in automotive applications [17].

Designing and implementing robust and efficient CPS is complex. The 5-level CPS structure, namely the 5C architecture, provides a holistic framework that helps organize, understand, and address the various technical challenges and considerations in deploying CPS across different industries and applications [18–20].

1.1.2 Industry 4.0 and 5.0 in Automotive Systems

As shown in Figure 2, the Industrial Revolution was characterized by a comprehensive overhaul of production methods and technological advancements. Substantial enhancements in production output, economic growth, and living standards distinguish this transformative period.

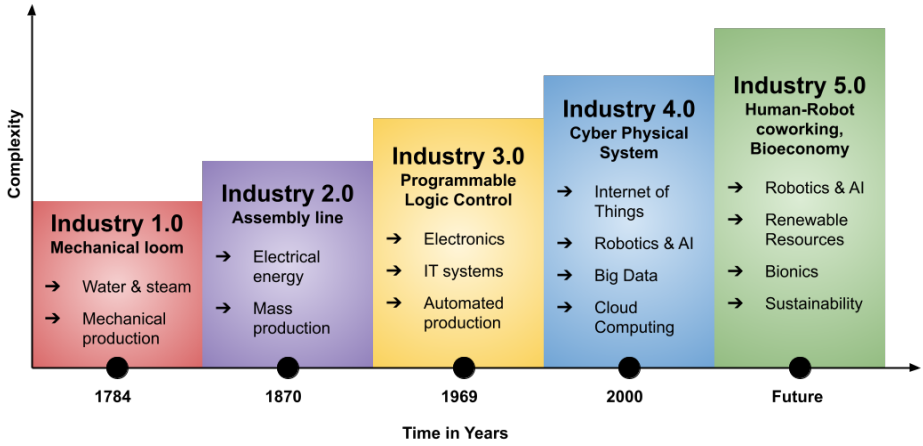

Figure 2: Industrial revolutions from Industry 1.0 to Industry 5.0 [21].

Industry 4.0, called “the era of CPS" [21], is a digital transformation of manufacturing, production, and related industries and processes for creating value. It has substantially impacted the global economy and the operational methodologies of businesses across the industry. This technological paradigm has facilitated heightened agility, efficiency, and environmental consciousness among companies. One of the most prominent features of 4.0 lies in integrating interconnected technologies, enabling seamless data exchange, process optimization, cost reduction, and quality enhancement for businesses.

Industrial 4.0 mainly contains CPS, Internet of Things (IoT), and cloud computing but will also rely on smart devices in addition to CPS. CPS is integral to Industrial 4.0 by blurring the gap between the digital and physical worlds. CPSs have led to many rapid technological disruptions in the industry [22].

The next industrial revolution, Industry 5.0, is announced as the next significant advancement. If Industry 4.0 is still ongoing, then Industry 5.0 has just begun and is progressing in parallel. This paradigm is anticipated to prioritize the synergy between humans and machines, enabling individuals to leverage their capabilities fully while enhancing workplace safety, efficiency, and significance. Combining high-power machinery and highly trained technicians allows companies to promote an efficient, sustainable, and se-

cure production process. Personal focus and resilience are key pillars of Industry 5.0. The goal is to make manufacturability sustainable from an economic, ecological, and societal perspective (green transition). This considers the competitiveness and energy efficiency of the industrial sector to increase the sustainability of production processes and make the industry more resource- and energy-efficient [22, 23].

Industry 4.0 and 5.0 emphasize connected and automated systems, with studies exploring the integration of AVs into industrial processes [24]. Regardless of the industry, functional safety and cybersecurity challenges remain primary concerns for implementing and deploying AVs. Industry 4.0 requires high levels of digitalization to process all the information generated in virtual representations or cyber versions of the physical world. From the product development and engineering point of view, realistic system simulations and the Digital Twin (DT) of an AV are beneficial to ensure proper complex system development and interactions between robots and humans.

The National Aeronautics and Space Administration (NASA) introduced the definition of DTs in 2012 [23]. It refers to a digital replica of a physical system or process that mirrors all its static and dynamic characteristics [25]. This virtual counterpart uses real-time data and simulations to mirror its physical counterpart's behavior, characteristics, and performance to optimize operations, enhance efficiency, and facilitate predictive maintenance.

1.1.3 Background of the Key Automotive Standards and Regulations

Technological innovations and progress in the automotive industry, especially with driver-assist and automated driving systems, have created a need for regulations, guidelines, and specifications to ensure vehicles' safe and reliable operation. To qualify for an automotive position, manufacturers must meet specific industry standards throughout the component manufacturing and testing. These standards cover various aspects of AV technology, including vehicle design, functionality, performance, testing, and deployment.

The design of automotive electronics relies on key standards that qualify failure mechanisms. Automotive Electronics Council (AEC) standard AEC-Q100 applies packaged integrated circuits [26], while AEC-Q200 sets a global stress resistance standard for all passive electronic components [27]. The Society of Automotive Engineers (SAE) also developed SAE USCAR2, which outlines performance testing requirements for electrical terminals and connectors [28]. The safety-related key standards developed by the International Organization for Standardization (ISO), Electrotechnical Commission (IEC), and Institute of Electrical and Electronics Engineers (IEEE) are ISO 26262, SAE J3016, ISO/IEC 21448, and ISO/SAE 21434. Verification and Validation (V&V) are defined in the ISO-IEC-IEEE 24765.

First published in 2011, ISO 26262—A, B, C, and D is an international standard for functional safety in the automotive industry. It provides guidelines and requirements for the development of safety-critical automotive systems. The standard defines the Automotive Safety Integrity Level (ASIL) as a risk classification system. A represents the lowest degree, and D represents the highest degree of automotive hazard. It addresses possible hazards caused by the malfunctioning behavior of vehicle safety-related systems, including the interaction of these systems [29, 30]. Fault Tree Analysis (FTA) is a systematic, deductive failure analysis method used to determine the various combinations of hardware and software failures and human errors that could cause undesired events (failures or errors). It employs a top-down approach, starting with a potential system failure and then identifying the possible causes that could lead to that failure [31].

The V-model testing approach in ISO 26262, as shown in Figure 3, provides a structured and systematic framework for testing automotive systems to ensure functional safety throughout the development process. It consists of two main phases. One left side rep-

resents the planning and preparation phase, while the right represents the testing and validation phase. Each step on the left side of the V-model corresponds to a complementary step on the right side, emphasizing the relationship between planning and testing throughout the development lifecycle. The first phase of system requirements analysis involves identifying and analyzing the safety requirements for the automotive system under development. Once the requirements are established, the system architecture and components are designed to meet them while considering safety. The second testing phase includes module, integration, and system testing. Each component is tested independently to ensure it functions correctly and meets specified requirements. The final phase is validation and verification to confirm that it meets the safety goals and requirements of ISO 26262 [32].

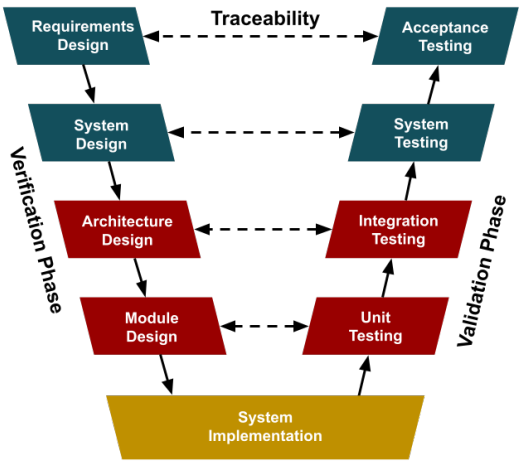

Figure 3: V-model testing approach in ISO 26262 based on [30].

SAE's Levels of Autonomy, also known as SAE J3016, categorize vehicles based on the extent to which human intervention is required to monitor the driving environment. According to the SAE, there are six levels of driving automation, as shown in Figure 4, ranging from Level 0 (no automation) to Level 5 (full automation). Numerous car manufacturers have introduced vehicles with level 2 autonomy, where the driver must remain engaged and ready to intervene if necessary [33].

ISO 21448 is a standard titled "Road vehicles safety of the intended functionality" (SOTIF). The standard addresses situations where a vehicle's intended functionality may not fully align with its operational context or environment. This can include scenarios where the vehicle's sensors, algorithms, or control systems may not adequately perceive or respond to unexpected events or conditions, leading to potential safety risks. Standard is relevant for ADAS, AVs, and other complex automotive systems. The interaction between human drivers, automated systems, and dynamic environments introduces additional safety challenges beyond those addressed by ISO 26262 [36].

ISO/SAE 21434 is an international standard titled "Road vehicles — Cybersecurity engineering." This standard provides guidelines and requirements for implementing cybersecurity measures in the automotive industry. It is designed to address modern vehicles' increasing cybersecurity threats, particularly as vehicles become more connected and autonomous [37].

ISO/IEC/IEEE 24765 is a joint standard titled "Systems and software engineering — Vo-

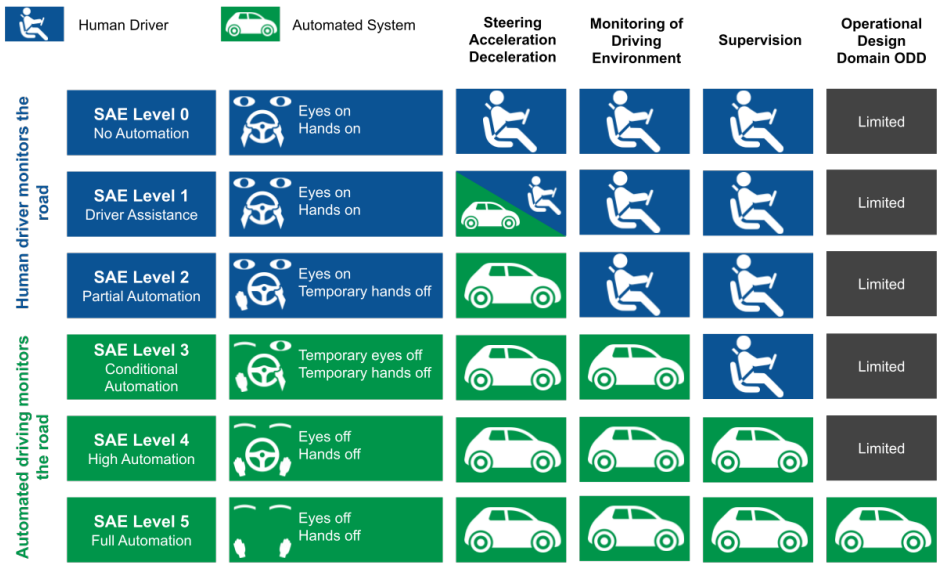

Figure 4: Five levels of AD by SAE J3016, based on [34, 35].

cabulary" that provides a comprehensive set of terms and definitions commonly used in systems and software engineering. The standard defines V&V as a process of assessing whether the requirements for a system or component are complete and accurate. It involves evaluating whether the products generated during each development phase meet the specified requirements or conditions set by the preceding phase [38].

UN Regulation No. 157, developed by the United Nations Economic Commission for Europe (UNECE), establishes uniform provisions concerning the approval of vehicles with Automated Lane-Keeping Systems (ALKS). It defines requirements for ALKS functionality, performance, and testing [39].

Various countries and regions have developed their regulations and guidelines for AV testing and deployment. In Estonia, the Sohjoa Baltic project has researched, promoted, and piloted the use of driverless electric minibusses in public transport from 2017 onwards. An automated driverless vehicle cannot obtain car registration because it does not comply with European law (e.g., UNECE rules) or the Estonian road traffic law (Traffic Act) regulations. In conclusion, it is possible by law to conduct test operations in Estonia with AVs (SAE levels 0–4). Still, it requires a test plate certificate, and every vehicle must have a responsible driver inside or outside [40].

1.1.4 Background of the Risk Analysis Model

With their potential to revolutionize the automotive industry, AVs bring numerous risks due to the multiple complex issues requiring comprehensive analysis. While the public eagerly anticipates a future with zero traffic accidents facilitated by AVs, it's essential to recognize that the technology and its related factors are still undergoing intense development. ISO 26262 is an international standard that provides guidelines and requirements for functional safety in the automotive industry. Consequently, risk analysis plays a pivotal role in ensuring AVs' safe and responsible advancement.

Evolutionary optimization techniques are a class of algorithms inspired by principles from biological evolution [41]. In engineering design, these techniques are most com-

monly utilized to handle mixed-integer variables and provide convergence to a global optimum [42–46]. Evolutionary algorithms are not optimal for handling decisions in uncertain scenarios. Hence, the subsequent utilization of Multi-Criteria Decision-Making (MCDM) methods becomes essential. MCDM is a methodological approach to helping choose between alternatives when faced with conflicting criteria or objectives with different levels of importance or relevance [47].

The Fuzzy Analytical Hierarchy Process (FAHP) is an extension of the Analytical Hierarchy Process (AHP) that incorporates fuzzy logic to handle imprecise or uncertain judgments in decision-making. AHP is a widely used multicriteria decision-making method developed by Thomas L. Saaty [48]. It enables decision-makers to analyze complex decisions by structuring them hierarchically and evaluating alternatives based on pairwise comparisons of criteria and alternatives. In traditional AHP, decision-makers provide crisp (precise) judgments when comparing alternatives and criteria. The AHP method is often criticized for using unequal scales and the inability to handle the uncertainties and accuracy in pair-wise comparison adequately [49]. FAHP addresses this limitation by allowing decision-makers to express their judgments using fuzzy linguistic terms, such as "very important," "moderately important," or "slightly important," instead of precise numerical values [50].

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is a multicriteria decision-making method used to determine the best alternative from a set of options based on their similarity to an ideal solution. It was developed by Hwang and Yoon in 1981 to deal with decision problems involving multiple criteria [51]. This system is not intended to replace the function of a leader in making decisions but only to assist in making a decision more quickly and precisely, according to the desired criteria or at least close to the desired criteria. Choice alternatives are expected to provide a list of references to decision-makers before making a decision [52]. According to TOPSIS, the most preferred alternatives should have the shortest distance from the Positive Ideal Solution (PIS) and the farthest distance from the Negative Ideal Solution (NIS) [53]. The TOPSIS method is widely used in transportation and intelligent vehicle systems [54–56]. In [56], a hybrid approach combines the TOPSIS and AHP methods.

Risk prioritization is a complicated MCDM problem that requires consideration of multiple feasible alternatives and conflicting tangible and intangible criteria. A novel hybrid approach that integrates the AHP, the TOPSIS, and the VIKOR method within the framework of MCDM provides a comprehensive analysis of the risks associated with AVs, which is crucial for identifying shortcomings and understanding their implications for users [53].

1.1.5 An Overview of Automotive Electronic Systems

Automotive electronics is the branch of electronic engineering focused exclusively on developing highly complex vehicle CPS designed to meet specific standards, incorporating design principles focused on safety and meeting functional safety requirements. The developers must face complex functional requirements, such as well-designed hardware and software functions, and non-functional challenges, such as portability, reusability, cost reduction, safety, and reliability [57].

The significance of electronic systems within a car's overall cost has steadily risen over the years, climbing from roughly 1% of the value of United States (U.S.) vehicles in 1950 to approximately 35% by 2020. Predictions indicate that by 2030, this figure could reach as high as 50% of the final cost of a car. This upward trend is primarily attributed to the persistent demand from consumers for enhanced electronic devices, Information Technology (IT) services, and connectivity, as well as the continuous integration of automation

features leading to AV development [58].

Electronic Control Units (ECUs) are microprocessor-based embedded systems responsible for executing control algorithms and managing the operation of various vehicle subsystems. ECUs are the building blocks of CPSs interacting with each other over in-vehicle buses. ECUs play a critical role in the operation and performance of modern vehicles, integrating and coordinating one or multiple specific functions. Today's vehicles may contain 100 ECUs or more [59].

Each ECU typically contains a dedicated microcontroller that runs its software or firmware and requires power and data connections. The microcontroller and other necessary components are soldered onto a Printed Circuit Board (PCB), typically containing a standard microcontroller, task-specific signal processing circuits, power supply, and power electronics components. Additionally, specialized drivers may be included for communication for data interfaces. The manufacturer selects electronics components based on his overall concept and due to cost, performance, or strategic aspects [60].

Manufacturers use standards-based frameworks for ECU hardware and software development, such as the Automotive Open System Architecture (AUTOSAR) [61]. This framework allows manufacturers to mix and match different automotive subsystems to target specific cost levels, performance, and functionality [58]. Most automotive companies are relatively conservative about open source, restricting the availability of AUTOSAR to the general research and education community [62]. Some critical in-vehicle ECUs, such as vehicle central gateway, ADAS, ALKS, and AI computers for self-drive, have more powerful computing and data storage capabilities for complex resource and task management or may even have operating systems and applications [59].

The automotive industry's Original Equipment Manufacturers (OEMs) have well-established supply chains to produce parts and components for use in new vehicles. These components can be reused for testing or small-scale production, but they are heavily protected, and access to source code or design files is closed. Countless open-source and open-hardware projects are available, but integrating them as a unified system is a significant and time-consuming challenge. The electronics design and other features of such projects may not be at the required level, often employing simplistic circuitry, the cheapest components, and a not-safety-oriented design.

The ECU containing standard microcontroller firmware usually operates in real-time environments where timely and deterministic execution of control tasks is critical. A Real-Time Operating System (RTOS) [63] provides the framework for scheduling and prioritizing control tasks, ensuring that critical control functions are executed within specified time constraints. This is essential for maintaining safety and stability in dynamic driving conditions. RTOS provides an environment for feedback control algorithms that maintain desired vehicle dynamics and performance. These algorithms continuously compare sensor measurements with desired setpoints or reference values and adjust control signals to minimize errors [57].

ECUs interact with each other over multiple in-vehicle buses. These automotive communication interfaces may utilize standardized protocols such as Controller Area Network (CAN) [64, 65], Local Interconnect Network (LIN) [66, 67], Ethernet [68], or FlexRay [69]. FlexRay and Ethernet, recently adapted for vehicle use, are new network communication systems targeted specifically at next-generation automotive or "by-wire" applications. By-wire applications demand high-speed bus systems that are deterministic, fault-tolerant, and capable of supporting distributed control systems [70].

AVs may communicate with other vehicles, infrastructure, and central control systems to exchange information about road conditions, traffic patterns, and potential hazards.

Vehicle-to-vehicle (V2V) [71] and Vehicle-to-Infrastructure (V2I) communication systems enhance situational awareness and coordination using WiFi, cellular communication (4G, 5G EDGE), and Bluetooth [72].

1.1.6 Background of the Model-Based Design

Developing and modeling vehicle low-level control systems is a complex task. Model-Based Design (MBD) is a methodology based on the V-model testing approach in ISO 26262 for designing complex systems, including control, signal processing, and communication systems. It's widely used in various fields, such as motion control, industrial equipment, aerospace, and automotive applications. MBD uses models for information transfer and interaction among designers and engineers. It enhances communication accuracy and quality, saves many iterative manual work, and improves system design efficiency [73]. MBD's main steps are the following:

- The system model and constitutional modules are developed.
- Based on the model, the program code can be generated automatically.
- System validation through X-In-the-Loop (XIL) tests is carried out [73].

The model is central to the entire design cycle, encompassing algorithm design, analysis, testing, and control system validation. The use of MBD significantly eliminates ambiguity and inconsistency. Furthermore, auto-code generation is typically employed for firmware development. Auto-code generation is based on system models using tools such as AUTOSAR, Simulink coder [74], and others. Therefore, any modifications to the control algorithm can easily lead to corresponding updates in the software. This approach saves considerable manual work and simplifies software version management. Consequently, the overall design efficiency of the control system is substantially enhanced [75].

The control system V&V is carried out through XIL tests. These tests include Model-In-the-Loop (MIL), Software-In-the-Loop (SIL), Processor-In-the-Loop (PIL), and Hardware-In-the-Loop (HIL) tests. Typically, MBD carries out the following four types of XIL tests to check its designs:

1. The MIL test involves evaluating the entire system model against various functional requirements and constraints. It excludes hardware or software integration testing and serves as an initial assessment to identify any design inconsistencies within the model.
2. SIL test, the auto-generated system-level codes are evaluated within the modeling framework. This process is crucial for identifying and rectifying software code errors during the early stages of design development.
3. PIL tests involve compiling the source code and loading it into the target processor. PIL tests aim to ensure that the control software functions properly on the processor and prevent bugs caused by compiler errors.
4. The HIL test, the highest level of XIL tests, focuses on assessing the functional response and performance of the software and hardware components based on the inputs received from the associated embedded system in a realistic environment.

The sequence of tests from MIL to HIL progressively encompasses the system's model, software, and hardware aspects [73].

1.1.7 Background of the Validating Automotive Systems

The V&V process is stated in the ISO-IEC-IEEE 24765 standard intended to verify specific predefined requirements, typically described in a technical specification. However, the standard also states that verification ensures the correctness of the system, confirming that it has been built according to predefined requirements and technical specifications [38]. On the other hand, validation ensures the system's suitability for its intended purpose, confirming that software quality meets the needs and expectations of its users [76]. Both V&V are essential processes in building reliable and effective systems, and they complement each other to ensure the overall quality and success of the system.

The AV contains deterministic and stochastic components, which present a significant challenge for V&V. Deterministic systems exhibit predictable behavior with known inputs and outputs typified by vehicle hardware and electronics. Conversely, stochastic processes, such as object detection, yield probabilistic outputs. In a deterministic system, each component undergoes validation individually at the elementary level, while at the integration level, the V&V process is conducted for all components working together. The stochastic system means that verifying its entire probability distribution is needed.

Due to the multitude of scenarios and driven kilometers necessary for V&V of AVs, three main ways can be described: simulation, track drive, and road testing. Real-world testing would take decades to accumulate over tens of billion accident-free kilometers, which alone is not a reliable safety indicator [77]. Among all testing methods, high-detail simulations show better performance considering cost and time [78,79]. Leveraging physics engines and DT of real-world environments can significantly reduce testing time and expense and test any upcoming potential feature in varying Operational Design Domains (ODDs), such as weather conditions or traffic patterns. While AI-based AV controllers are effective in real-world situations, they may disregard physical rules, resulting in atypical decisions. As a result, the significance and complexity of validation and verification V&V of AD functionalities increases [80, 81].

AV simulations, exemplified by platforms like CARLA [55], LGSVL [82], and AWSIM simulator [83], primarily leverage the concept of DT to validate and verify the safety and performance of AVs. Autoware [56], an open-source software project, is dedicated to AD. At the same time, CARLA and AWSIM focus on game-engine-based simulation and provide assets for constructing environments, including urban details and road users. Within these simulations, the abstraction level of models typically aligns with the operational abstraction of the Device Under Test (DUT). The multiple Design-Of-Experiment (DOE) allows for the comprehensibility of various scenarios (environment, dynamic actors) and correctness criteria (pass/fail). The DT, encompassing the vehicle under test and its operational surroundings, serves as a direct input to the simulator, defining the environmental domain and its characteristics, such as buildings, vegetation, and road configurations. The simulator uses the Autoware stack to incorporate scenario definitions within the DT environment. Validation reports are generated based on the outcomes.

When constructing a DT, it must closely mirror the real-world application to ensure high precision and superior results. In the initial development of the DT for an AV, the sensor configuration must align with that of the actual AV, and the 3D graphical model is a replica of the vehicle's body, as it is utilized for collision detection. The virtual environment must also resemble the real test area in features and conditions.

The simulator is designed to facilitate the creation of any virtual environment and the target vehicle, offering greater flexibility and compliance for conducting various tests. It also allows for the generation of test scenarios that incorporate pre-built features. A single scenario can include multiple events to formulate a complex test plan.

Upon executing a simulation, the simulator supplies virtual sensor inputs to the control algorithms, such as those provided by Autoware.ai. The perception algorithms receive this raw data, which is then processed by different units. Subsequently, the software determines the necessary actuation command and communicates it back to the simulator environment via a ROS bridge. Rosbridge provides a JavaScript Object Notation (JSON) Application Programming Interface (API) to ROS functionality for non-ROS programs [84].

Depending on the test scenario, a range of safety and performance Key Performance Indicators (KPIs) can be established, with the relevant data being captured during the simulation runs. This data can later be scrutinized to identify vulnerabilities and corner cases where the DUT failed to meet the set metrics [85, 86].

Low-level CPS can also be V&V using simulation. Modern automotive designs increasingly rely on distributed low-level embedded control systems, underscoring the importance of exploring the effects of various design choices on system safety and reliability. A highly efficient approach to conducting such investigations is through a meticulously detailed HIL simulation. HIL testing is a technique where signals from a controller are connected to a test system that simulates real-world applications. It tricks the DUT into thinking it is in the assembled product, providing valuable insights into system behavior and performance [87].

The AVs feature multiple ECUs linked to a central gateway, necessitating cybersecurity integration tests to meet the homologation requirements. Typically, vehicles are equipped with a gateway server, often supplemented by additional domain servers. These servers manage many ECUs responsible for real-time control of actuators like Electronic Stability Control (ESC), Electric Power Steering (EPS), and Anti-Lock Braking System (ABS), usually comprising multicore embedded controllers. These ECUs are interconnected with the domain controller or gateway server via a real-time automotive-specific bus, such as CAN, FlexRay, or Ethernet. Compliance with standard ISO/SAE 21434 mandates cybersecurity-related verification and validation processes. For verification, car manufacturers employ a network test suite comprising over 2000 test cases, which must be successfully passed. The testing procedures for vehicle communication infrastructure dictate the cybersecurity attack patterns assessed before the release of an ECU or vehicle onto the road [37, 88].

In conclusion, the development, adaptation, setup, and debugging of low-level control systems needed for AD is only possible using XIL testing. For example, the ABS and ESC require many field tests. The application of modern real-time simulation technologies based on HIL simulation allows for decreased field test count [89].

1.1.8 Background of Autonomous Vehicle Technology

The low-level control system framework developed in the thesis is intended to improve existing AVs or for use on new vehicles. Understanding the state of AV development is crucial in this context. Autonomous minibusses or shuttle buses with 6 - 12 seats have been developed worldwide for low-speed environments, such as university campuses, industrial parks, and areas with minimal traffic. This strategy prioritizes safety by operating at low speeds, enabling swift deployment. Subsequent technological enhancements and the accrued knowledge from experience will pave the way for expansion into high-speed settings. Ultimately, the aim is for the vehicle's capabilities to match those of a human driver across all driving scenarios. Current shuttle buses serving public roads reached Levels 3 and 4 of the autonomy characteristics, according to SAE J3016 [90]. This means the vehicles are fully automated and operate in a defined operational domain without onboard human control devices. The ODD sets the limits in which the conditions of the vehicle are designed to operate in terms of geographical area, weather and road conditions, speeds

and traffic density, etc. Control systems differ from traditional setups, lacking a steering wheel, brake/throttle paddles, and other controls.

Numerous companies and organizations were actively working to bring AVs to market. Baidu, a Chinese multinational technology company, and Waymo, a subsidiary of Alphabet (Google), have been key players in AV technology since 2009, extensively testing self-driving cars on public roads. Additionally, various startups, such as EasyMile, Uber, Navya, and others, have emerged to advance automated vehicle technology. Estonia is also at the forefront of AV technology, with companies like AuVe Tech, Starship Technologies, and Clevon (Former name: Cleveron Mobility) developing advanced self-driving vehicles and robotic delivery systems. Declarations from car manufacturers like Tesla and Ford about imminent fully autonomous cars [91] have yet to happen, and repeated delays have occurred [92].

One notable AV shuttle series is the iseAuto 1.0, as shown in Figure 5, designed and collaboratively developed by the TalTech Autonomous Vehicles research group and company Silberauto [93–95]. The TalTech iseAuto project aimed to create an open-source AV shuttle and establish a smart city testbed on the university campus, facilitating various urban mobility and autonomy-related research initiatives. Since 2018, the testbed has been utilized for numerous AV studies, including the present study. The vehicle utilizes open-source software and incorporates a modular design, reducing manufacturing costs. The project showcased its first public demonstration in September 2018 and has proven successful.

Figure 5: TalTech iseAuto 1.0 newer (left) and older (right) version (H. Pikner).

The development of Estonia's first self-driving vehicle is ongoing, receiving a fresh look and updated interior. TalTech's iseAuto v2.0 (Figure 6) will soon have a more dynamic and memorable appearance, and both the software and hardware are undergoing significant updates. The findings and knowledge acquired from Taltech iseAuto 1.0 and this study will help to achieve street-legal status for the new shuttle as fast as possible, ensuring strict adherence to safety standards and regulations.

The Navya Evo (Figure 7) is a self-driving shuttle manufactured by the French company Navya [96], founded in 2014 and later rebranded as Gama. Gama specializes in the development of autonomous mobility systems and associated services. Widely recognized as a mature and established product, the shuttle has undergone pilot testing worldwide. The Navya Evo remains a significant player in the evolving landscape of AVs, contributing to the exploration of self-driving technologies in various regions and use cases.

At TalTech, an integrated team comprising students and researchers from the Department of Mechanical and Industrial Engineering has initiated the development of a new robot designed for industrial use and tasked with transporting boxes. Leveraging knowledge and experience gleaned from past self-driving vehicle projects, notably the self-driving platform iseAuto [97, 98], as well as research on methodologies for unmanned

Figure 6: TalTech iseAuto v2.0 (H. Pikner).

Figure 7: Navya Evo shuttle (H. Pikner).

ground vehicle development, the team embarked on this endeavor [99, 100].

Figure 8: Universal easily expandable AMR prototype BoxBot 1 (H. Pikner).

The prototype of the AMR "BoxBot" (Figure 8) was completed for Kulinaaria OÜ in 2019. Development of the second prototype (Figure 9) continued in 2020. BoxBot was designed to autonomously transport goods in warehouses and factories, boasting a load capacity of up to 100 kg and dimensions of 750 mm in length, 340 mm in width, and 230 mm in height. It achieves a maximum travel speed of 2 m/s and operates for at least 6 hours. Omni wheels allow the robot to turn on the spot and move laterally. The new version of BoxBot employs enhanced technology and can be remotely controlled via a DT. It also has an improved LiDAR system, a camera, and a laser to indicate the robot's travel trajectory, facilitating indoor autonomous navigation. The robot is a collaborative development between TalTech and Kulinaaria OÜ.

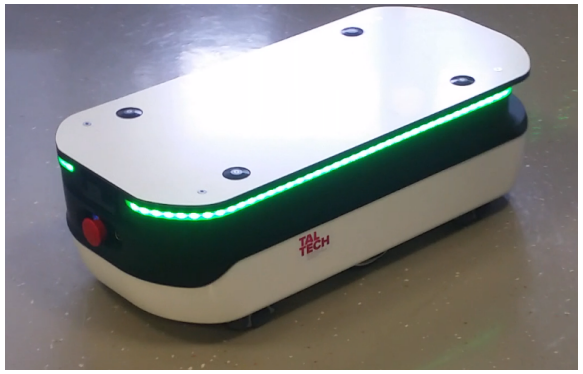

Figure 9: BoxBot 2 (H. Pikner).

1.2 Definition of the Research Problem and Objectives

Recent advancements in AV technology have made driverless transportation the norm. However, eliminating human drivers significantly increases the complexity of low-level CPSs. In the past, drivers could manipulate the steering wheel and brake/throttle paddles with their hands and feet, ensuring the vehicle's safety. If something goes wrong, the driver can take action and safely bring the vehicle to a halt. However, with the driver completely absent or transferred to a remote control center, system failures may go undetected, potentially leading to hazardous situations or accidents.

This doctoral research focuses on a scientific approach to improving the low-level CPS of AVs using modular and MBD approaches. The objective is to create a more universal CPS framework that can be applied to various use cases in vehicles and robots. Standardized concepts for ECUs, firmware, network design, configuration, and error detection can be developed to achieve this. These methods can then be introduced as part of a comprehensive framework and should be tested on multiple case studies.

The next focus is low-level CPS implementation, verification, and validation challenges, utilizing risk analysis models and V&V methodologies. The aim is to find ways to validate automotive systems through comprehensive simulation using DT of low-level control systems, ensuring functionality and safety following key automotive standards, governing manufacturing protocols, and safety regulations.

This doctoral thesis proposes a modular low-level CPS framework for AVs to improve their sustainable integration of novel methods, development processes, and tools tailored specifically to the automotive domain. MBD emerges as a potential methodology for designing, implementing, and managing distributed systems. The framework includes modular electronic design and data bus concepts based on risk analysis models and key automotive standards. Safety and security are achieved by utilizing the V&V methodology and advanced fault detection and recovery mechanisms.

1.3 Research Hypotheses

Table 1: Relationship between Research Hypotheses (RH) and the included papers.

	Paper I	Paper II	Paper III	Paper IV
RH1	✓		✓	
RH2		✓	✓	
RH3		✓		
RH4				✓

The main Research Hypotheses (RH) of this thesis are:

RH1 Methodology functions as a practical design, validation, and verification framework for vehicle hardware and software components that may allow porting a low-level control system or its subsystems to different vehicles or autonomous robots, which can significantly improve development speed, system overall reliability, and cost.

RH2 Advanced fault detection and recovery mechanisms for low-speed AVs can present unique failure modes and safety hazards at the system and unit levels. It is essential to validate that these algorithms consistently produce safe and efficient behavior across various driving scenarios.

RH3 Estimating several different types of risks and evaluating multiple criteria is challenging in developing AV systems. Numerical solutions may help identify and assess potential hazards and vulnerabilities.

RH4 Due to the multitude of scenarios and driven kilometers necessary for validation, low-level hardware can be simulated or implemented inside the validation platform, assessing strengths, weaknesses, and opportunities, which may reveal the intricacies of constructing TDs with high predictive value.

Our research plan is designed to answer the key hypotheses mentioned in the four research articles. Table 1 summarizes the relationship between the research hypotheses and the corresponding article.

1.4 Research Tasks

The main Research Tasks (RT) of the thesis are:

RT1 Analyze and explore key automotive standards, governing manufacturing protocols, and safety regulations to ensure compliance and adherence to industry best practices, ultimately increasing the reliability and safety of AVs.

RT2 Analysis of the computational (cyber) units for low-level, real-time fault monitoring and crash detection is essential to oversee accidents and control signals issued by other cyber units or sensors. If a safety-critical issue arises, actions can be initiated to solve the problem safely.

RT3 Analyze existing control systems to identify weaknesses and issues crucial for developing a modular framework and improving control strategies, particularly for different self-driving platforms or AVs.

RT4 Validate the developed framework to transfer the low-level control system from one shuttle to another. The accomplishment also paves the way for integrating proven autonomous technologies into various older AV models, extending their useful lifetime and reducing costs.

RT5 Conduct research to develop a mathematical model for risk analysis that incorporates non-functional aspects, including real-time responsiveness, sensor reliability, communication robustness, and environmental uncertainties.

RT6 Analyze and explore various V&V methodologies, which can be integrated into the framework to validate the functionality and performance of the AVs' low-level control systems using the DT concept. This allows for real-time monitoring, testing, and analysis of the system's behavior.

1.5 Contribution and Dissemination

This thesis addresses comprehensive research on low-level CPSs and algorithms focusing on developing a modular, user-friendly, and expandable low-level control system framework to meet the unique requirements of dependable AMRs or AVs. The framework includes modular electronic design and data bus concepts based on risk analysis models and key automotive standards. Safety and security are achieved by utilizing the V&V methodology and advanced fault detection and recovery mechanisms. The knowledge and results included in this research have led to the development of expandable AMR prototypes, "BoxBot 1" and "BoxBot 2", which were experimented with in the Kulinaaria factory in Tallinn. The successful experiment involved transforming Navya Evo's shuttle into an open-source solution. Further, it culminated in creating the new parallel-built shuttle, TalTech iseAuto 2.0.

The findings of this research have been introduced in two international conference papers and two peer-reviewed international journals.

Scientific Novelties:

- This methodology functions as a practical design, validation, and verification framework for vehicle hardware and software components to improve safety.
- An MCDM risk evaluation model for safety system assessment. The approach may simplify decision-makers' judgments and handle the uncertainty caused by these judgments. The risks identified are universal and applicable to outdoor AMRs and other low-speed AVs.
- Improved strategies to offer a combined platform for a controlled and consistent testing environment, facilitating rapid prototyping and evaluation of the XIL tests for safety validation.

Practical Novelties:

- Verified framework for an AV low-level CPS, validated with AMRs in a real industrial environment and Navya Evo.
- Experiments with expandable AMR prototypes in industrial environments to discover the KPIs.
- An experiment is conducted to transfer the low-level control system from one vehicle to another, particularly when the specifications of the target vehicle are not well-known. Rigorous experiments and tests are carried out on low-level and high-level components to ensure the safety and reliability of the new solution.
- Comprehensive guidelines of key automotive standards, governing manufacturing protocols, and safety regulations to ensure compliance and adherence to industry best practices, ultimately increasing the reliability and safety of AVs.

By leveraging dissemination strategies presented in this thesis, researchers and practitioners can accelerate the adoption and integration of AV low-level control systems into future autonomous transportation systems, developing safer, more efficient, and more sustainable mobility solutions.

2 Low-Level Control System Design, Validation, and Verification Framework

This chapter discusses the methods that form the low-level control system design, validation, and verification framework. The proposed framework encompasses several critical concepts related to the electronic control module, its firmware, and the automotive network design. Additionally, the verification and validation of the low-level control system emphasize the importance of functional safety and requirement analysis. Simulation-based X-in-the-Loop methods and real-world testing procedures are covered to ensure comprehensive testing. Furthermore, the chapter addresses the development of hazard analysis and risk assessment, underscoring the critical safety aspect.

The system architecture concept was first introduced in the "Autonomous Last Mile Shuttle ISEAUTO for Education and Research." It presents an account of the experience of developing a vehicle from scratch in one year using a stock electric vehicle, widely available sensors, and open-source software [101]. The concept is divided into a vehicle control architecture located inside the vehicle and an external system, such as fleet orchestration, as shown in Figure 10.

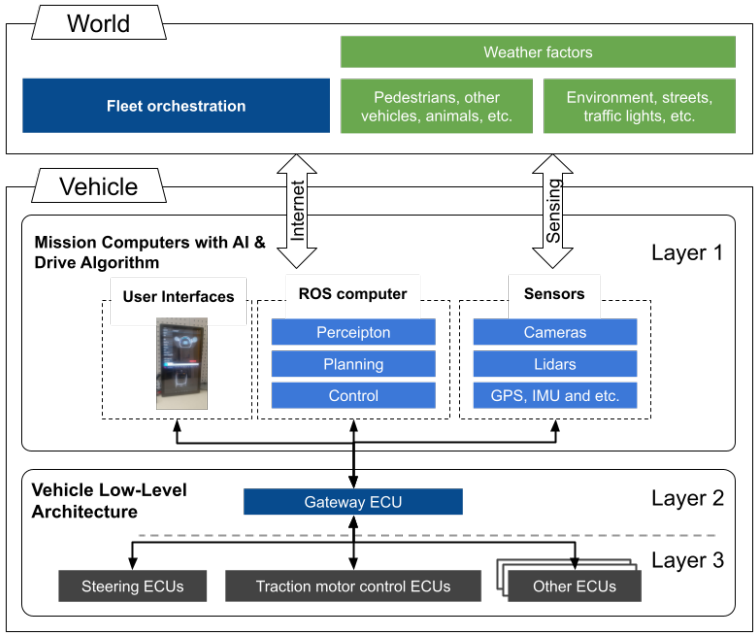

Figure 10: General vehicle control architecture (H. Pikner).

The vehicle control architecture also consists of two layers. The first layer, 1, is based on Ethernet and includes one or more mission computers with AI & drive algorithms and high-level sensors that interact with mission computer-level software. Autonomy is achieved by running Autoware on top of the ROS on a mission computer. To minimize delays, this mission computer communicates with the low-level controllers via a dedicated Ethernet connection. Autoware integrates components for 3D localization, 3D mapping, path planning, path following, vehicle control (including acceleration, braking, and steering), data logging, and object and obstacle detection, among other things. The number of cameras, lidars, and radars used can vary depending on the vehicle platform. Also, user interfaces

may be present in some vehicle variations and must be able to communicate directly with the Gateway ECU or the mission computer.

Layers 2 and 3 contain mission-critical CPS, consisting of a Gateway ECU (layer 2) and slave controllers (layer 3) integrated with actuators and sensors. The Gateway ECU's main task is to run system-specific algorithms and act as a gateway to forward information from and to the mission computer with the minimum delay. It handles communication with ROS, listens to data from the vehicle's multiple CAN or other networks, and communicates with the slave controllers over networks.

Fleet orchestration (as shown in Figure 11) is accessible via wireless communication and is available to multiple vehicles. These systems are responsible for fleet control, real-time diagnostics, data collection, and remote control center infrastructure communication.

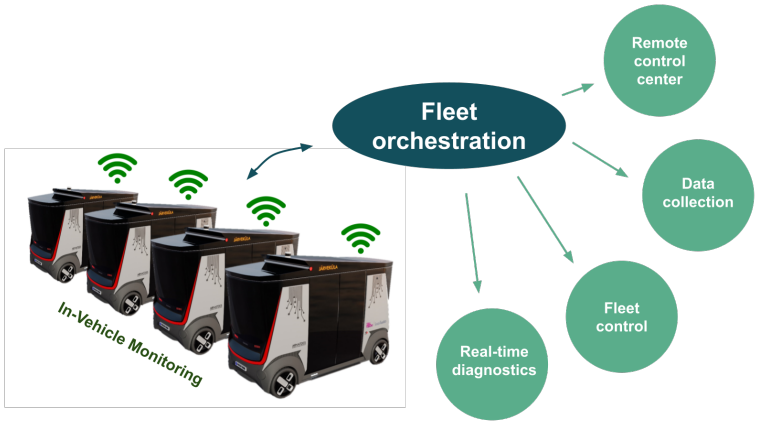

Figure 11: Fleet orchestration architecture concepts (H. Pikner).

Automotive CPS integrates embedded systems, control theory, real-time systems, software, and electronic engineering. Automotive in-vehicle networks comprise numerous ECUs, sensors, and actuators that operate many control loops that are closed over in-vehicle networks. In these configurations, functionalities are achieved through distributed tasks. The effectiveness of these functionalities depends on message delay, jitter, and task execution times connected with specific controller designs. Historically, the focus has been on mathematical models, their analysis, and high-level simulation. During this process, several simplifying assumptions have been made in the design of ECUs. These cause computation times to exceed when assessing control laws and underestimate the times taken for control message communication. AV may not have a human driver inside the vehicle, causing the complexity of CPS to increase and the semantic gap between high-level control models and their actual implementations to expand [102]. As a result, a growing need exists to adopt a more comprehensive CPS design approach.

2.1 Proposed Framework Design Concepts

This section provides an overview of the evaluation methodology proposed in this study. This methodology is a practical design, validation, and verification framework for vehicle hardware and software components, as referenced in RH1. A framework utilizes FTA to introduce an ASIL-oriented hardware design for safety-critical automotive systems. ASIL is crucial in the ISO 26262 safety standard, measuring risk for specific system components. For V&V, the standard ISO-IEC-IEEE 24765 focuses on verifying specific, pre-defined re-

quirements typically outlined in technical specifications. Other key automotive standards and regulations are listed in Chapter 1.1.3, as RT1 requires. Figure 12 illustrates the main steps of the framework.

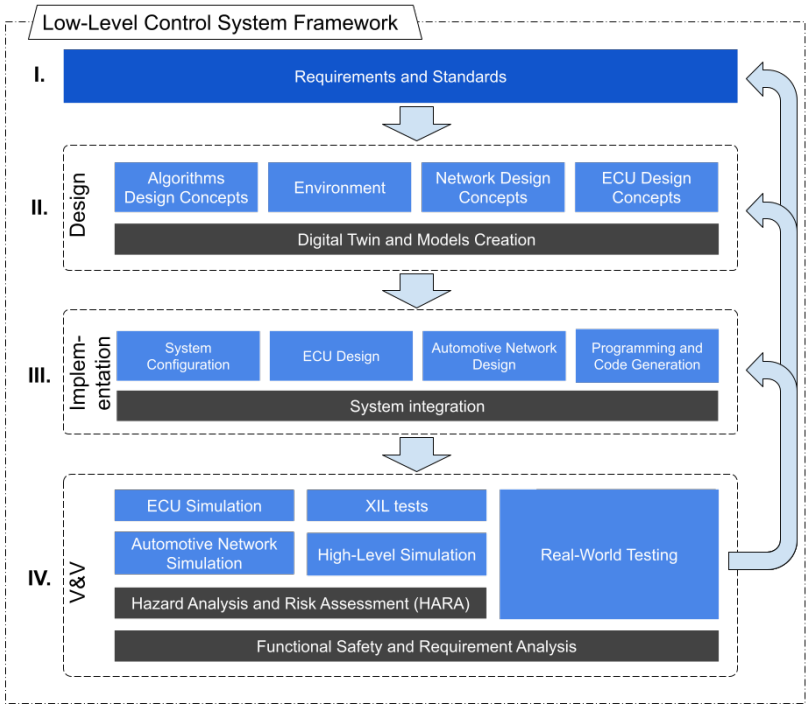

Figure 12: Low-level control system framework consists of four phases (H. Pikner).

The MBD concept is used for the low-level control system development process [103, 104]. It integrates continuous testing of the proposed design at all stages of development via XIL protocols. MBD framework can detect design flaws early in the development cycle, simplifying and speeding up the design correction process.

Phase I, requirement and standards analysis, focuses on determining the tasks that the ideal system should accomplish. The requirements usually outline the system's functional, interface, performance, data, and security needs.

Phase II, the design phase, entails creating system hardware and software architecture and a DT. The architecture must cover all elements, including a list of ECUs and networks, a concise description of each functionality, interface relationships, dependencies, architectural diagrams, and technological details. The ECU design process involves decomposing the designed system into smaller units or modules. Each module is described in detail or through the developed model to facilitate the next phase of model-based code generation or direct coding. The low-level design document or program specifications will encompass the ECU's detailed functional logic, comprehensive interface details, all dependency issues, error message listings, and complete lists of inputs and outputs. The methodology for ECU testing is also formulated during this stage. ECUs are developed for various automotive functions, while communication networks like CAN or Ethernet are designed to enable data exchange within the vehicle.

Phase III, the implementation phase, is a critical stage in developing an automotive

system and involves the manufacturing, programming, and model-based code generation of vehicle hardware and software components. The hardware components, which include ECUs, sensors, actuators, and communication networks, must be meticulously developed to meet system requirements and withstand the harsh automotive environment. Software implementation entails coding the functionalities defined during the design phase II. This can be accomplished through direct coding or model-based code generation. Once the hardware and software components are ready, they must be integrated. This step ensures that the software can effectively control the hardware components, allowing the system to perform its intended functions.

Phase IV, the validation phase, ensures that the designed system meets all requirements through steps. These include ECU simulation to verify functionality and XIL tests conducted in a simulated environment with real hardware components in some cases. DT can be used for virtual testing and validation, as well as real-world testing in actual vehicles or controlled environments, which are integral components of the validation process. Verification assesses whether the system adheres to predefined requirements. It involves evaluating risks associated with system components using a risk evaluation model and ensuring compliance with automotive standards such as ISO 26262. Together, these phases provide the development of safe, reliable, and compliant automotive systems.

The low-level control system framework includes a feedback loop. The result of the V&V phase can be fed back to a previous phase, resulting in a development cycle of cause and effect that forms a feedback loop. For example, suppose the HIL test indicates that the ECU is not working as expected. In that case, it is possible to revert to the requirements phase again, refine the requirements, and then proceed to design, implement, and test again.

This solution's versatility ensures seamless integration of low-level control systems into various types of vehicles or robots, thereby reducing the development time and resources required to implement autonomous functionalities. Such accessibility can accelerate the advancement of autonomous technology, paving the way for a safer and more efficient future of transportation. The subsequent subchapters will discuss all steps in detail.

2.1.1 Electronic Control Module Design Concepts

Each type of ECU component must comply with international automotive application standards. Automotive microcontrollers qualify according to the AEC-Q100 standard and have a wide range of automotive interfaces. The chosen specialized hardware must allow the achievement of safety goals [105]. Passive components qualifying to AEC-Q200 and automotive connectors are used to design new ECUs. Automotive connectors must be crimp-type connectors to establish reliable connections and save time. For example, the Wire-Lock low-mating-force automotive-grade connector system is a good option and is USCAR-2 V2-compatible.

The number of controllers varies depending on the vehicle platform. Modularity can be achieved by developing a distinct set of types of ECUs, each capable of handling a specific set of tasks:

1. Gateway ECU handles the gateway and main controller functionality. Additional Gateway ECUs can be added to one vehicle if more data communication networks are needed.

2. The drive controller manages complex systems, such as vehicles' driving and traction-related functions.

3. Steering ECU manages motors for vehicle steering and handles information from force, angle, and temperature sensors.

4. The Power-Controlling ECU controls power buses inside the vehicle. Each controller has multiple power buses, which can measure current, have multiple switchable power channels, and act as an E-fuse.

5. The Brake ECU manages brake system components, such as pressure sensors, brake fluid levels, and actuators.

This kind of modular approach necessitates the ability to configure and maintain uniform firmware across all controller types. Each type of ECU contains a configuration parameter list, facilitating easy modification and fine-tuning. To improve security, each ECU broadcasts error flags. If a safety-critical fault is present, the mission computer, Gateway ECU, or other controllers can decide whether it is safe to continue driving or if the vehicle must be parked and assistance requested.

The overall safety-related design is important. The primary concern is ensuring that the microcontroller continues to execute the program and issues an alert if a fault arises. A technical solution could involve a watchdog or co-microcontroller monitoring the main microcontroller. If the main microcontroller malfunctions, one of the simplest remedies might be a restart. Additionally, if the main microcontroller transmits an error flag message on the communication network at a specific interval and this message disappears, another ECU could execute commands to halt the vehicle. A more sophisticated solution involves a separate communication network for safety-related co-controllers within the ECU. This way, if the primary communication network fails, the safety-related controllers can still communicate amongst themselves and halt the vehicle. This solution is designed for scenarios where halting the vehicle necessitates the collaboration of multiple ECUs. An error flag must also be issued when the ECU fails to achieve a recommended actuator position according to the feedback within a specified time interval. This could indicate a mechanical fault, a power unit fault, or an actuator motor fault.

Furthermore, error flags can be broadcast over wireless communication, which is helpful for the data collection system, real-time diagnostics, and for predictive maintenance. This approach ensures the vehicle's safety and efficiency while providing valuable data for ongoing improvement.

The internal electronics of the ECU can be designed to be robust. This means that neither over-voltages nor under-voltages (provided these remain within the predetermined limits), electrical interference, nor short circuits applied to power inputs, digital IO, or data interfaces can disrupt the operation of the microcontroller. If the ECU serves as a power source for the sensors, the power supply to the microcontroller should remain stable, even if this source is short-circuited or if an excessive current is drawn. The ISO 26262 standard requires that automotive applications tolerate at least one critical fault to maintain intended functionality or achieve or maintain a safe state [29]. The ASIL risk classification system must be used to mitigate the risks when designing every ECU.

Electronic protection circuits can replace fuse and relay boxes for powering ECUs [89]. They are faster and allow faults to be logged as soon as they occur. Two separately protected power supply lines can be added for the critical controllers. A good candidate is the steering controller. If one power line is faulty or short-circuited, the other will continue to work.

ECUs must transition into a low-power state, such as when the vehicle is unused. There are two solutions to this. A Power-Controlling ECU with controllable power rails is utilized in the first scenario. These can be switched on and off as required by the consumers

connected to them. Another option is for the ECU to be directly connected to the battery power bus and manage the low current state. In this case, the ECU may have an ignition or an enable signal pin. When it is energized or pulsed, the ECU powers up. If a corresponding message is received over the data bus, the ECU powers down, for instance, by executing important procedures before shutting down, such as moving the actuators to the correct position, etc. It is generally advisable to use a combination of both methods. Lights and other simple devices can be connected to power management controllers. More complex ECUs with higher current requirements can manage the shutdown process and be directly connected to the power bus.

2.1.2 Electronic Control Module Firmware Design Concepts

ECU firmware is based on RTOS and is widely recognized for its robust performance in embedded systems. The general firmware architecture for an ECU, as shown in Figure 13, explains various software layers and their interactions, including drivers (Layer 1), middleware (Layer 2), and applications (Layer 3).

Drivers (Layer 1) contains all the important drivers for the microcontroller hardware and its data buses. Peripheral drivers are available from the microcontroller manufacturer. Specific drivers are required for controlling custom PCBs developed as a part of the Board Support Package (BSP). Middleware (Layer 2) runs the FreeRTOS kernel and scheduler, as well as the modules needed for the User Application (Layer 3).

ECU-specific threads in applications (Layer 3) execute various tasks. Some of these threads, such as the CAN, Error Flag, and Configuration, are common across different ECUs. The Gateway ECU has a User Datagram Protocol (UDP) thread that handles all UDP data received on the ethernet interface. UDP is a communications protocol primarily used to establish low-latency and loss-tolerating connections between applications on the Internet [106]. The UDPSend thread sends UDP data to the mission computer. There must be a predefined structured way in both directions to pack a signal inside UDP packets identical to all vehicle or robot variants. Distinct UDP packets for different subsystems are developed, each containing signals from a specific field. Depending on the requirement, these signals could be activated or deactivated as needed. For instance, a UDP packet designed to control a vehicle's air conditioning system might not be necessary for an indoor AMR.

In addition to common threads, ECU-specific threads run specialized algorithms. Examples of these specialized algorithms include Proportional Integral Derivative (PID) regulators, Pulse Width Modulation (PWM) modules, and others. These threads can share common variables which are packed inside structures. This arrangement, along with other RTOS methods, ensures data protection and access controls to available resources, thereby enhancing the overall efficiency and reliability of the system and providing time savings for the firmware design. This modular and layered approach to firmware architecture facilitates flexibility, scalability, and maintainability in the design and operation of vehicle ECUs.

The fundamental concept is that each ECU is self-aware of its operational requirements, the conditions that must be fulfilled, and the limited values of the signals under its responsibility. Each ECU transmits a packet of flags, enabling other ECUs within the same network to verify the online status and operational adequacy of the ECUs they are concerned with. An ECU must execute routines to mitigate hazards if the system is not sufficiently operational. For instance, if the Gateway ECU abruptly vanishes from the network, each ECU must take action within its area of responsibility to stop the vehicle. This could involve stopping the traction motor, switching off the high-voltage system, turning

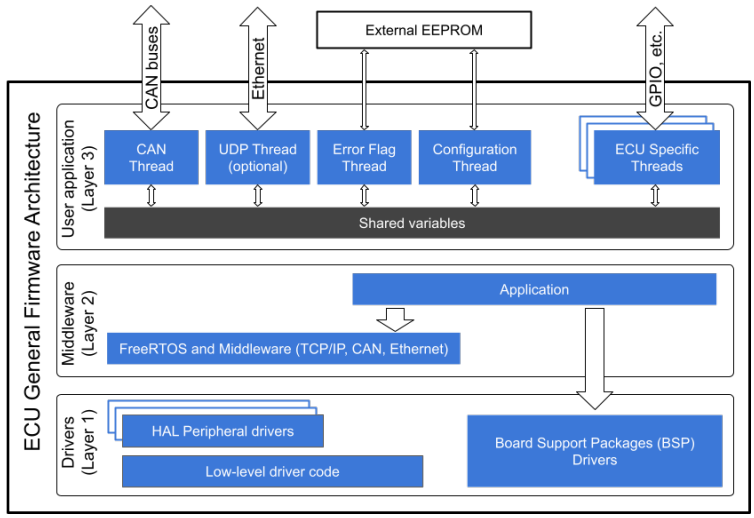

Figure 13: ECU firmware architecture concept (H. Pikner).

on the hazard lights, etc. If the issue is not critical, such as an exceeded motor temperature while everything else is functioning, allowing the vehicle to drive to a safe location and park there might be wise.

ECUs of the same type must share identical software and vehicle-specific functionality, achieved through configuration. ECU is equipped with an EEPROM memory in which the configuration is stored in JSON format with key-value pairs. This approach offers the advantage of human readability, but it necessitates the development of a user interface for entering the configuration and transmitting JSON text files over the CAN network.

2.1.3 Automotive Network Design Concepts

Automotive control systems are often seen as distributed, where primarily ECUs, sensors, and actuators are interconnected to ensure efficient operation. Numerous automotive networks have been developed to fulfill strict requirements. The most common are CAN, LIN, FlexRay, and Ethernet.

Due to the distributed nature of the automotive architecture, the control code needs to be partitioned into software tasks for sampling the sensor readings (T_s), computing the control inputs (T_c), and performing actions (T_a) as shown in Figure 14 [102].

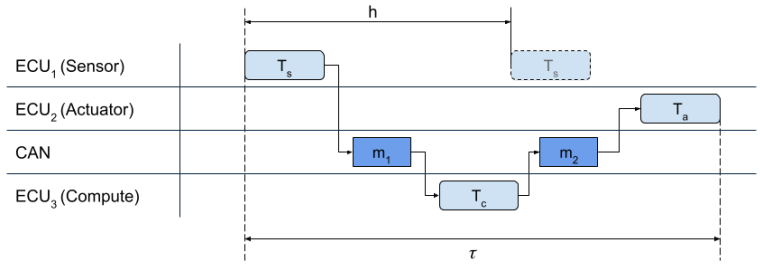

Figure 14: Timing diagram of distributed CPS [102].

In a digital platform where such feedback loops are implemented, operations are carried out at specific time intervals, referred to as sampling instances. The performance of the system depends on two factors:

- The sampling period h,
- The delay τ from the sensor to the actuator also depends on the network message delay m.

The design challenge from the control perspective is that a shorter period enhances control performance but also necessitates additional computational and communication resources. The ultimate objective is to select sampling periods and corresponding control-related parameters for the control tasks in a way that ensures the control functions are robust and achieve the desired performance. All tasks and messages can be scheduled within the resource constraints [102].

A CPS system's security can be enhanced through message encryption. However, as the number of encrypted messages increases, the system's security level improves, but the platform's schedulability decreases due to the additional overhead. Furthermore, extending the sampling periods to preserve schedulability may be necessary, which could further impair control performance. The presence of stringent resource constraints, such as limited communication bandwidth and computational resources, coupled with strict timing requirements for system safety and performance, makes it challenging to integrate these security mechanisms after the initial design stages without violating system constraints or hindering system performance [105].

Incorporating an additional lightweight checksum and counter value into critical data frames helps to improve security and reliability. The controller will only utilize the data frames if the checksum is verified as correct. Also, a checksum makes it more challenging to inject messages into the network [107]. Included counter values increase with every message to help detect any loss of data frames. An error flag is triggered if the anticipated data frame fails to arrive within the correct time interval or the counter value does not match.

Although it is not practical or efficient to encrypt every message in an automotive network due to the overhead and complexity, it is certainly feasible to selectively encrypt specific critical messages. For instance, messages that control the immobilizer could be encrypted as a security measure. This approach provides a balance between maintaining system performance and enhancing security, making it a viable solution for protecting the integrity of the automotive CPS.

2.2 Low-Level Control System Verification and Validation Concepts

V&V of AV systems presents a significant challenge, necessitating the development of research frameworks that can advance the current state-of-the-art. Primary methods for V&V can be identified:

1. Functional safety and requirement analysis.
2. Simulation (XIL) and real-world testing.
3. Hazard Analysis and Risk Assessment (HARA).

2.2.1 Functional Safety and Requirement Analysis

In implementing ISO 26262, each staff member and management team member must understand the risks and action plans associated with the system. These plans, which encompass systematic documentation, scheduled training, and effective issue management, are crucial to success. During the initial phase, an evaluation of the system will be conducted. This evaluation will include a hazard analysis and risk assessment, with a focus on the various components that constitute the system. The objective is to identify potential hazards and evaluate the associated risks, thereby ensuring the safety and reliability of the system. Following this, safety goals (SF) can be established, and Automotive Safety Integrity Levels (ASIL) can be assigned to all identified hazards.

Functional Safety Analysis is pivotal in our safety assessment process, particularly in evaluating a product's safety level. This analysis involves quantitative evaluations, such as Failure Mode Effect and Diagnostic Analysis (FMEDA) and Timing Analysis, and qualitative assessments, such as Dependent Failure Analysis (DFA).

The process of functional safety verification commences with the standard functional verification setup. Specifically, a fault injection that evaluates the safety mechanism primarily needs a description of the workload to execute, the Observation Points (where the effect of faults is observed), and the Detection Points (where the reaction of a safety mechanism is followed). Faults are classified based on their impact on the observation and diagnostic points [108]:

- Dangerous Detected: The effect of the fault is seen on both observation and diagnostic points. This means the injected fault affects the functional output, but the safety mechanism has detected it.

- Dangerous Undetected: The fault's effect is seen on the observation points but not the detection points. In other words, the fault affects the functional output, and the safety mechanism has not detected it.

- Safe: The fault does not affect the observation point. It's important to note that a fault can only be classified as safe if the workload provides good coverage for functional verification.

When setting up the fault injection tests, choosing where to inject the faults is crucial. The safety mechanism under evaluation targets a specific circuit failure mode, so faults should only be injected into the logic belonging to the related failure mode.

2.2.2 Simulation X-In-the-Loop and Real-World Testing

XIL testing offers early validation of system components and enhances cost efficiency by simulating realistic scenarios. This approach reduces the dependence on physical prototypes, saving time and resources during product development. Furthermore, XIL testing supports the continuous improvement and precise adjustment of system components.

Based on this, the most effective approach is constructing a simulated low-level control system model within a DT, given the wide variety of scenarios and the substantial distance required for AV validation, as referenced in RH4. One reliable method involves using MATLAB/Simulink to construct ECUs using Simulink block diagrams [109]. The Database CAN (DBC) file defines the contents of CAN messages [110]. Simulink blocks can handle CAN traffic and, when configured with DBC files, can generate behavior almost identical to the original controller's logic. In this scenario, the actual controller's software, written in C, and the Simulink models can differ somewhat in functionality.

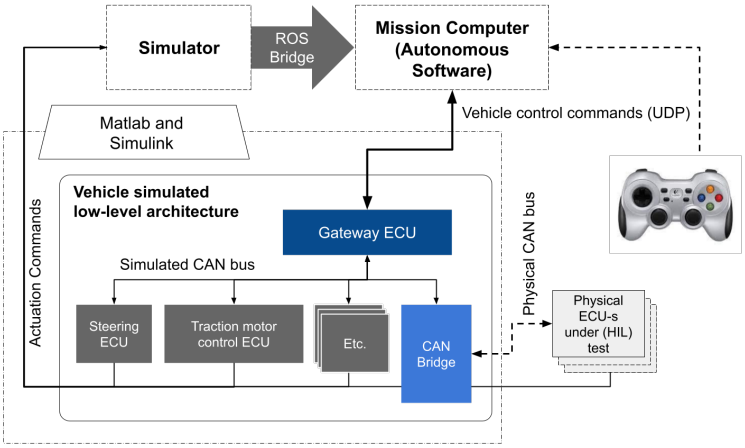

Figure 15: Low-level control system HIL simulation experimental structure.

MATLAB and Simulink allow the generation of a simulated low-level architecture for vehicles, including ECUs, data buses, and mechanical models. The high-level vehicle mission computer, running autonomous software, can generate control signals driven by simulators like CARLA and LGSVL. A bidirectional ROS bridge is replaced so that the simulator sends only sensor and lidar information to the high-level mission computer. Control commands sent out by UDP packets are readable to the simulated/physical gateway ECU. All signals pass through the simulated low-level control system model between the control computer and the simulator. The CAN bridge facilitates a bidirectional connection between physical and simulated data buses, as shown in Figure 15. While the simulation is running, traffic is generated on a simulated data network that can be used to test and develop physical controllers.

The proposed setup enables testing of stand-alone ECUs or vehicle subsystems in a HIL environment when a vehicle self-drives inside a simulation and simultaneously generates all the traffic on the data networks. Such a test system facilitates easy and rapid validation for developing control modules and simulating the entire system operation. Different designed situations and disturbances allow for performing various tests. It also provides testing scenarios that would be too hazardous to conduct in real traffic scenarios. Tests can run for extended periods to control the stability and durability. Furthermore, the parameters of an actual vehicle can be compared against the model, and any discrepancies between the vehicle and the DT in response to the same input might indicate a possible fault.

If the simulation and XIL test are done, then real-world testing allows the evaluation of a vehicle under actual driving conditions. Real-world testing provides a complete vehicle assessment, from mechanical performance to interaction with real environments and situations. It extends to various aspects of vehicle quality, including performance, reliability, and user experience.

2.2.3 Hazard Analysis and Risk Assessment Development Concepts

Risk analysis methodology offers a structured approach to identifying, evaluating, and managing potential risks in a project. As referenced in RH3, the risk analysis methodology provides a foundation for enhancing safety in future advancements. Concurrently, RT5 is about researching and developing a mathematical model for risk analysis. This model

incorporates non-functional aspects such as real-time responsiveness, sensor reliability, communication robustness, and environmental uncertainties.

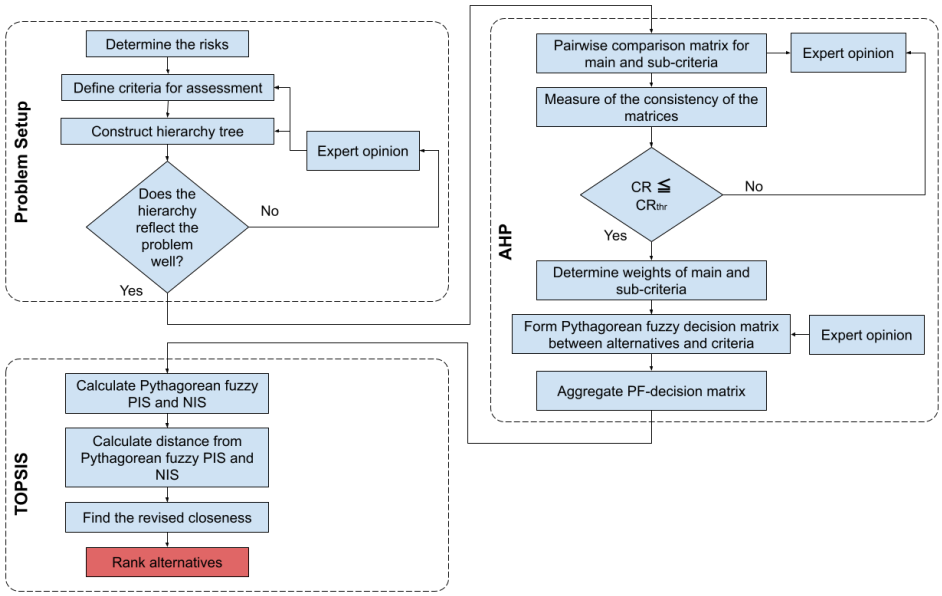

Figure 16: The risk analysis methodology schematic is based on [53].

The proposed risk evaluation model consists of three main modules, as shown in Figure 16.

- Formulation of criteria and risks [111].
- Prioritization of criteria (fuzzy AHP).
- Prioritization of risks (fuzzy TOPSIS).

The selection of the TOPSIS method is justified by its simplicity of implementation and ability to provide a PIS while avoiding a NIS. The PIS comprises all the best criteria values attainable, and the NIS comprises all the worst criteria values attainable. These factors collectively contribute to TOPSIS's suitability for addressing the research objectives and conducting effective decision-making analysis.

The fuzzy AHP and TOPSIS approaches are combined to prioritize the criteria and risks. The proposed fuzzy sets-based approach allows us to apply linguistic assessments corresponding to the judgment's natural representation [53, 112]. The first module covers formulating the criteria and risks for the different types of vehicles [111]. In the following, the fuzzy AHP approach, based on triangular fuzzy numbers (TFN), prioritizes the abovementioned criteria.

1. The criteria evaluation using linguistic variables [113].
2. The pairwise comparison matrix criteria vs. criteria regarding linguistic variables filled by the expert decision-makers.
3. The linguistic scales transfer to triangular fuzzy numbers (TFN).

4. The aggregated evaluation matrix computed by applying a fuzzy geometric mean

$$r_{ij} = (\Pi_{n=1}^{N} c_{ijn})^{1/N} \tag{1}$$

Equation (1), c_{ijn} stands for the fuzzy comparison value in terms of the TFN of criteria i to criteria j given by the n-th expert, and N is the total number of decision-makers involved. The computed values of the pairwise comparison matrix r_{ij} are calculated. Here $r_{ij} = (l_{ij}, m_{ij}, u_{ij})$ are triangular fuzzy numbers, where l, m, and u stand for lower, medium, and upper values, respectively.

5. The aggregation is applied for each row of the aggregated comparison matrix. As a result, the fuzzy comparison values $r_i = (l_i, m_i, u_i)$ can be evaluated as:

$$r_i = (\Pi_{j=1}^{Ncrit} r_{ij})^{1/Ncrit} \tag{2}$$

Equation (2) $Ncrit$ stands for the number of criteria used.

6. The triangular fuzzy weight w_i of criteria i is determined as the normalized value of the r_i

$$w_i = (l_i, m_i, u_i) = r_i \otimes (r_1 \oplus r_2 \oplus ... \oplus r_{Ncrit})^{-1}, ...i = 1, ..., Ncrit \tag{3}$$

7. Crisp weights can be obtained by applying defuzzification to fuzzy weights (different approaches to defuzzification [114]).

$$w_i^{Crisp} = l_i + [(u_i - l_i) + (m_i - l_i)]/3 \tag{4}$$

8. The criteria are prioritized based on normalized crisp weights. The consistency ratio (CR) of the defuzzified matrix is calculated and validated. If $CR < CR_{thr}$, the matrix has acceptable consistency, and if $CR > CR_{thr}$, the judgments should be revised until $CR < CR_{thr}$. CR threshold CR_{thr} is 0.1 (10%) [53]. The normalized crisp weights and criteria ranks can be considered the final results of the fuzzy AHP.

The risk evaluation is performed as follows, considering the results of the applied fuzzy AHP and utilizing the fuzzy TOPSIS approach.

1. The pairwise comparison risk vs. criteria analysis is performed by the same expert group that evaluated the criteria. Similarly to the above, triangular fuzzy numbers and linguistic variables are employed [115].

2. The risk evaluation concerning criteria is performed.

3. Based on the relations, the linguistic "grades" given by decision-makers are transferred to triangular fuzzy numbers (TFN). The aggregation of the decision-makers evaluation matrices is performed by applying the fuzzy arithmetic mean (in the case of Fuzzy AHP applied geometric mean) as:

$$x_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_{ijn} \tag{5}$$

where N is the number of decision-makers and x_{ijn} stands for the rating of risk i to criterion j given by the n-th decision-maker. The fuzzy triangular number $x_{ij} = (l_{ij}, m_{ij}, u_{ij})$ must be calculated.

4. The aggregated fuzzy decision matrix is normalized. The fuzzy weights of the criteria obtained by applying fuzzy AHP are utilized to compute the weighted normalized decision matrix.

5. The distances of each risk to positive and negative ideal solutions are computed as

$$d_i^+ = \sum_{j=1}^{n} d(v_{ij}, v_j^+), i = 1, ..., m, d_i^- = \sum_{j=1}^{n} d(v_{ij}, v_j^-), i = 1, ..., m, \tag{6}$$

Where

$$v_i^+ = (1,1,1), v_j^- = (0,0,0), j = 1,2, ..., n \tag{7}$$

And

$$d(x,y) = \sqrt{\left(\frac{1}{3}\right) \cdot [(l_x - l_y)^2 + (m_x - m_y)^2 + (u_x - u_y)^2]}. \tag{8}$$

6. Based on the positive and negative ideal solution, the similarities are calculated as

$$C_i = \frac{d_i^-}{d_i^+ + d_i^-}, i = 1, ..., m. \tag{9}$$

The risks are ranked based on the values of the similarities.

Estimating several types of risks and evaluating multiple criteria is challenging in developing AV systems. The fuzzy AHP-TOPSIS-based risk analysis approach proposed here provides estimates of the ranks of criteria and risks.

2.3 Safety

Safety-critical automotive applications have strict demands for functional safety and reliability. Safety is determined from a risk analysis, management perspective, and technical standpoint. The following steps are recommended to mitigate major risks:

1. Practical design, validation, and verification framework, based on standards, utilizing multiple methods that improve safety and reliability.

2. Implementing a transparent software development process, including code reviews and signing, is essential. This strategy helps maintain code quality and intercept issues before they are executed on the AV.

3. Regularly testing the low-level CPS and mission computer-level software helps improve safety.

4. Everyone in the development team should understand the risks and action plans associated with the system.

5. Adopting a checklist-based culture helps operate and set up the equipment faster. This method minimizes human error in repetitive tasks that sometimes need to be performed under time pressure.

6. Multiple safety features must be designed at a shallow level to ensure the vehicle stops if an anomaly is detected.

7. The vehicle's top speed must be restricted to a shallow level to improve safety.

3 Case Studies and Results

The chapter delves into the practical application of the methodologies formulated in Chapter 2, showcasing their effectiveness through case studies. Case studies based on TalTech iseAuto v1.0 and v2.0, Navya Evo shuttle, and universal easily expandable AMR prototype BoxBot 1 and 2. These case studies provide valuable insights into the methodologies' real-world performance and their potential to revolutionize AV technologies.

3.1 Autonomous Mobile Robots Case Study

The advancement of intelligent control technology has significantly impacted the logistics sector and led to the emergence of innovative indoor AMR. These robots are deployed within warehouses and manufacturing facilities to enable the movement of goods and parcels within confined spaces. There is a high motivation to reduce the cost of these AMRs and create flexible control systems. A novel, compact AMR tailored for navigating tight spaces and transporting boxes. This solution achieves flexibility and cost efficiency in logistics operations by leveraging cutting-edge technologies such as AI and machine vision.

The compact space inside the AMR was efficiently utilized to integrate the low-level control systems inherited from existing AVs. Specifically, modules were developed to control the differential-drive wheels and the lifting mechanism. This new control architecture is structured into three distinct layers, as described in Figure 17.

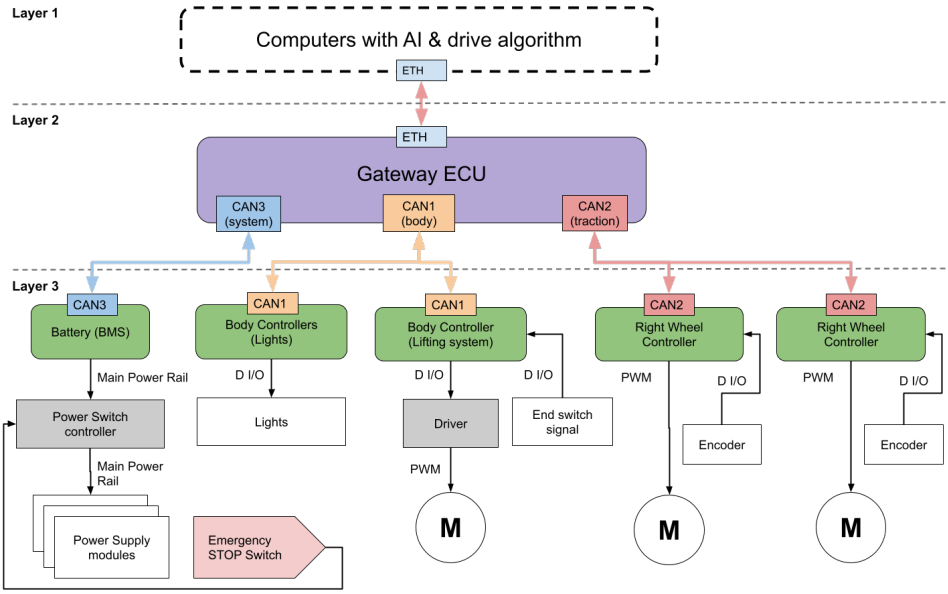

Figure 17: The low-level architecture of the AMR control system (Paper I).

The high-level software solution (layer 1) is based on ROS and modular architecture [116], allowing open-source drivers for multiple sensors and easy integration of third-party software like Autoware [117]. The main sensors are multiple 3D lidars for localization, obstacle detection, safety functions, and path following. Control outputs are twist command, brake, and linear velocity sent over gateway ECU (layer 2) to the low-level controllers using UDP protocol. Gateway ECU communicates with ROS and the sensing and actuation layer

3 over different CAN networks. The CAN1 network is dedicated to the lifting mechanism and other body controllers. The CAN2 network is for the propulsion system. The number of slave controllers may vary depending on the progress of the development.

The AMR underwent rigorous factory testing, achieving an impressive 80% success rate. However, these tests also revealed weaknesses and challenges that must be addressed in the subsequent iteration. Some of these issues were specific to the factory's configuration, while others were more universal, highlighting the dynamic collaboration between humans and robots. For instance, significant alterations in the placement of goods within storage areas could lead to disparities in maps, resulting in the robot losing its localization. This challenge will be addressed in the refinement of the robot's next iteration.

A new and smarter version of the AMR prototype was introduced in 2021. It now has a greater carrying capacity, improved technology, and can be remotely controlled through a DT. The new version has a better LIDAR system, a camera, and a laser for displaying the movement trajectory. The robot moves for up to six hours on a single charge. Also, a new system for monocular visual localization based on AprilTag fiducial marker detection has been introduced. The solution has been designed for autonomous docking in industrial applications. It localizes an AMR toward a docking point marked by fiducial tags [118]. The next approach for analyzing the performance of AMR in transporting goods on the manufacturing plant floor is based on creating and simulating the 3D layout, monitoring KPIs, and using AI for proactive decision-making in production planning. In the food industry, KPIs for AMRs include the number of transportation boxes, transportation time, and robot utilization. These KPIs are crucial as they assess the efficiency and effectiveness of the AMR transportation process. By focusing on these specific KPIs, the transport process can be optimized, costs reduced, customer satisfaction improved, and productivity increased. A case study of the food industry demonstrates the relevance and feasibility of the proposed approach [119, 120].

The Research Hypothesis RH1 stated that the modular design principles inherent in AV's low-level control systems and automotive networks offer the potential for transferring a low-level control system or its subsystems between various vehicles or AMRs. The experience gained from constructing two iterations of the AMR demonstrates the advantages of expediting the development process for new robots. Moreover, this approach facilitates enhanced modularity and universality within the system. This modularity allows for flexible adaptation to different scales of operation and varying logistical requirements. Universality ensures that the performance metrics are applicable across various industry segments, leading to faster, easier, and more cost-effective robot development. Also, RT3 examined current control systems to pinpoint critical weaknesses and issues, which was essential for developing a new iteration of an AMR and allowed the improvement of modular low-level control system design, validation, and verification framework.

3.2 Autonomous Vehicle Communication and Safety Architecture Based on the Risk Evaluation Model

The risk analysis methodology is a part of the low-level control system design, validation, and verification framework requested in RT5. Estimating several types of risks and evaluating multiple criteria is challenging in developing AV systems. The fuzzy AHP-TOPSIS-based risk analysis approach provides estimates of the ranks of criteria and risks. The criteria and risks are defined, and the seven criteria and ten risks are formulated and described. Cyber hacking, low-level software, and electrical failure appear to be the most critical risks in the current case study, as listed in Table 2. The table shows the positive d_i^+ and neg-

ative d_i^- ideal solutions, the similarities C_i, and the final ranking of the risks. The risks were ranked based on the values of the similarities. The crisp weights of the criteria and the similarity values of the risks provide more detailed and valuable information for the further improvement of AV systems [94].

Table 2: Final ranking of the risks (Paper II).

Rank	Description of the risk	d_i^+	d_i^-	C_i
1	Cyber hacking	6.171	0.893	0.1264
2	Low-level software failure	6.173	0.893	0.1263
3	Electrical failure	6.204	0.872	0.1232
4	Mechanical failure	6.269	0.789	0.1118
5	AD software failure	6.300	0.761	0.1078
6	Loss of localization	6.309	0.749	0.1061
7	Information shortage	6.378	0.679	0.0963
8	A drastic change in the environment	6.385	0.669	0.0949
9	Interruption of uplink	6.399	0.656	0.0930
10	Communication bandwidth shortage	6.413	0.645	0.0914

As the results indicate, cyber hacking is the highest risk factor. However, encrypting the entire low-level communication network is not feasible, so building protection at the highest possible layer is reasonable. Low-level software failures are the second highest risk factor and demand significant attention during the low-level CPS design and implementation stages. The proposed model prioritizes risks specific to the TalTech iseAuto 1.0 and 2.0 development, focusing on low-level hardware-software safety issues and improvements [99, 121]. Next, considering the risk analysis results, the iseAuto 1.0 low-level CPS will be upgraded.

The high-level (layer 1) software architecture of the iseAuto is based on the ROS. The AD stack performs perception, detection, and planning. Sensors configuration and position are well-detailed in [85]. Precise filtering and concatenation processes were performed on the LiDARs point cloud to optimize perception. The shuttle AD software runs on ROS, and its customized software architecture is described in [95]. Vehicle speed and direction commands are dispatched to the low-level CPS, which has the crucial task of actual vehicle control.

Figure 18 shows the Taltech iseAuto 1.0 CPS's architecture. In addition to the first layer, the low-level CPS contains two layers: the gateway layer 2 and the sensing and actuation layer 3. The gateway layer contains Gateway ECU, the primary function of which is to serve as a central hub for all nodes. The sensing and actuation layer has a drive controller with complex low-level functionality. The drive controller manages the vehicle's movement and steering, directly controlling the OEM car chassis. Also, a separate safety controller has been implemented to stop the vehicle if a fault is detected. The communication layer operates on distinct CAN networks and Ethernet [10].

The architecture of the TalTech iseAuto version 2.0 CPS, shown in Figure 19, is based on risk evaluation outcomes and the modular CPS concept. The low-level CPS continues to be split into two layers besides the first: the gateway layer 2 and the function-based sensing and actuation layer 3. The gateway layer can be enhanced with more than three CAN buses and other automotive interfaces such as LIN and K-line. The previous system's Gateway ECU had only one UDP connection. A new Gateway ECU permits multiple connections, facilitating information sharing among additional user interfaces. If a more extensive system with extra data buses is required, additional Gateway ECUs can be incorporated. Function-

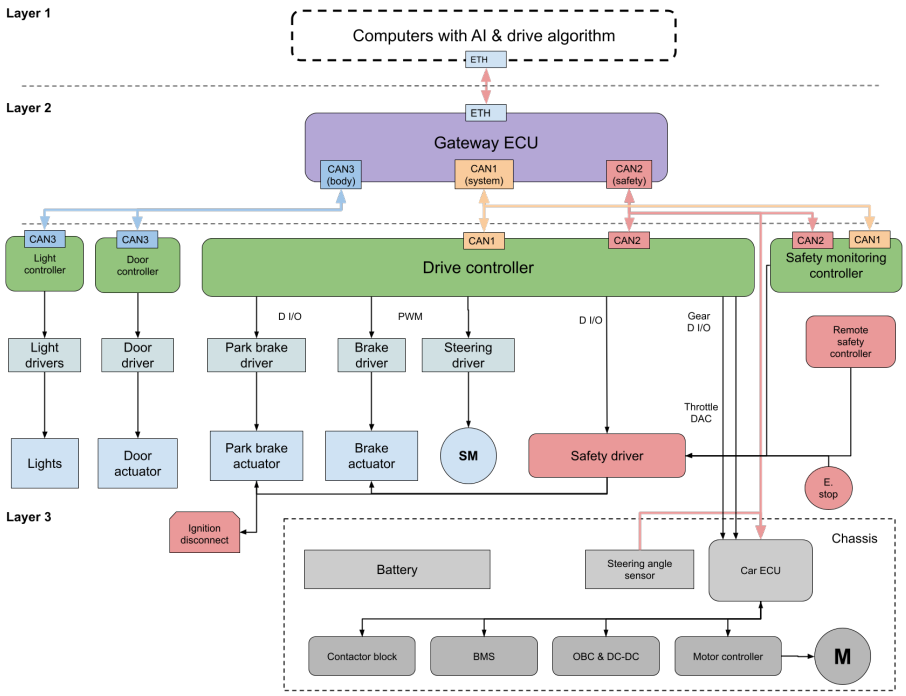

Figure 18: The low-level control solution for TalTech iseAuto v1.0 [10].

based controllers are classified as either critical or non-critical.

Critical controllers are engaged in the vehicle's direct control or the management of the traction battery and its charging process. The role of the drive controller has been diminished; it now controls a vehicle platform only if that platform is in use. The vehicle platform can be replaced with a self-developed version. Numerous sub-functions are distributed among dedicated ECUs. For safety purposes, separate safety controllers and a distinct safety communication bus have been incorporated to allow remote control when a fault or malfunction occurs. The communication is shared between three CAN buses:

- CAN 1 for all system controllers;
- CAN 2 for safety-related controllers and for duplicating critical system messages;
- CAN 3 for vehicle body-related and other low-priority controllers.

RH3 stated that a comprehensive risk analysis model may help to identify and assess potential hazards and vulnerabilities within the AVs' low-level control systems. Indeed, the proposed approach simplifies the decision-maker's judgments and manages uncertainty caused by these judgments. The identified risks are universal, applying not only to a specific autonomous shuttle but also to similar outdoor AMRs and other low-speed automated vehicles. The risk evaluation results are input for further developments and improvements of AVs, particularly the under-development TalTech iseAuto 2.0.

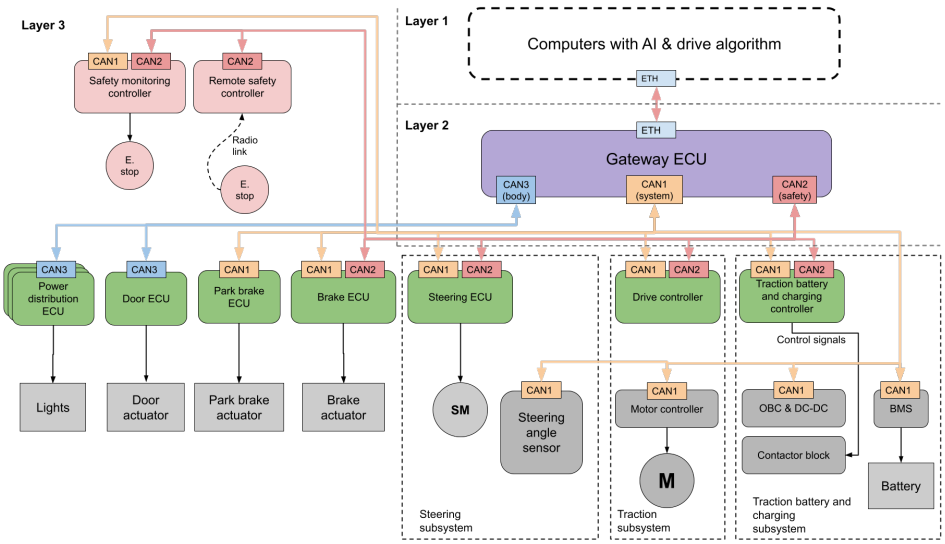

Figure 19: The low-level control solution for TalTech iseAuto v2.0 (Paper II).

3.3 Cyber-Physical Universal Safety and Crash Detection System for Autonomous Vehicles

Automotive security has become more challenging with the advent of advanced modern technologies. Moreover, AVs depend on the absolute reliability of electronic systems to operate flawlessly as designed. In hazardous occurrences, appropriate preventive or corrective actions are taken even if the driver is not in the vehicle. The initial safety system was deployed on Taltech iseAuto in collaboration with the ABB Estonia development team [89], and RT2 requires analyzing and improving it.

The implementation of the safety controller is segmented into three distinct controllers, as depicted in Figure 20. The primary safety controller is engineered to monitor the CAN bus and various signals. It is directly linked to the mission-critical drive controller for monitoring purposes, and in the event of a malfunction, safety relays are triggered.

Safety relays can directly switch the brake actuators and OEM vehicle platform ignition signals in the right sequence, so timing circuits were included. The vehicle emergency stop switches inside the vehicle are also connected to the relay module.

AV can be used without a safety operator inside the vehicle. The operators were close to the vehicle in this case, so a wireless safety button was developed. The LoRa network [123] proved successful in testing. If the safety operator is not close to the vehicle, another solution is to have the remote-control center located anywhere in the world. Both extra controllers directly control the safety relay box.

The existing safety controller's General-Purpose Input and Output (GPIO) pins are connected to the drive controller's analog and digital pins for real-time signal analysis to determine whether the expected signal levels are in the appropriate range [98]. If the signal is out of range, it automatically activates the brake signal, immediately stopping the vehicle. A separate safety/measurement board provides a single point for getting the actual measured control signal values issued by the drive controller. Measurement results are published into the CAN bus. Main functions of the developed safety controller [122]:

- Controlling the Relay/Break supply module for making an emergency shutdown

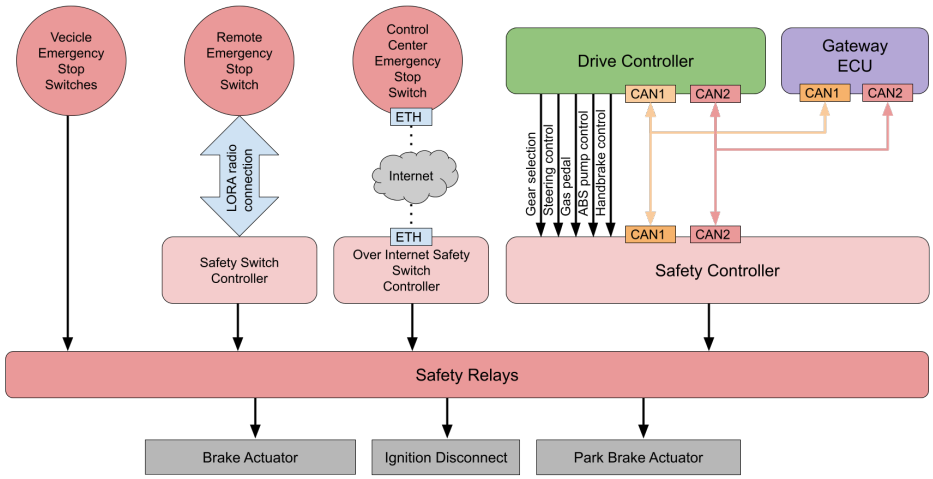

Figure 20: The architecture of the TalTech iseAuto safety system [122].

based on a command from the Gateway ECU. Engaging emergency brake on Gateway ECU request.

- Measuring accelerator, steering angle, brake, and shift position selection signals. Detect any signal abnormalities and perform an emergency shutdown when anomalies occur.
- Measuring the handbrake's actual state (is it applied or not) and forwarding this information to the Gateway ECU over the CAN bus.
- Sending measurement results of the accelerator, steering angle, brake, and shift position selection over the CAN bus to the Gateway ECU.

The safety controller must react immediately to any abnormal conditions listed in Table 3 or an emergency shutdown signal from the Gateway ECU. When abnormal signal values are detected, the safety controller sends a signal to the safety relay module to activate the emergency shutdown.

In addition to the primary safety protocols, a comprehensive set of rules has been designed for the brake system. These rules are tailored to address a variety of events or potential hazard situations that may arise. The design of these rules considers the severity and nature of each scenario, ensuring that the brake system responds with the appropriate level of urgency and precision. By doing so, the system enhances the overall safety and reliability of the vehicle's operation, particularly in critical moments where swift and decisive action is required, as shown in Figure 21. It is divided into three stages (Paper II):

1. Normal braking is usually triggered by a high-level computer or safety lidar. When there is enough space (road), regenerative braking can be used, followed by normal braking if needed;
2. The emergency brake is triggered when the emergency STOP switch is pressed, the front safety lidar detects an obstacle too close, or when the safety monitoring controller is triggered by a fatal error;

Table 3: Abnormal conditions detection [122]

No.	Signal	Condition
1	Accelerator	The accelerator pedal signal consists of two voltage signals. When the accelerator „position“ is controlled, sub and main values must change simultaneously. Also, the Main/Sub should always have a certain ratio. So Main/Sub ~= 0,5 V. The safety controller could check the ratio and that both signals change correctly.
2	Steering angle	Steering control is done with a single 50 Hz PWM signal. Pulse width should be between 1 – 2 ms. If the pulse width drops below 1 ms or goes over 2 ms, the safety controller could flag this as a problem and apply emergency braking.
3	Handbrake on/off	A 5 kHz signal controls the handbrake. The safety controller could measure the signal's frequency, and if it goes out of range, drops significantly below, or goes above 5 kHz, a safety controller could trigger an emergency shutdown. Also, the condition when the drive controller tries to apply and release the brake simultaneously could be flagged as a problem.
4	Shift position	Check that the shift position selection is made according to the specified order: P->R->N->D. Trying to engage multiple shift positions simultaneously would also fail.

3. An emergency shutdown may be followed by emergency braking when the emergency STOP switch is pressed (for example, if there is a fire risk because there is smoke in the cabin), the crash detection system is triggered, or some serious error is detected. An emergency shutdown disables the high-voltage traction battery.

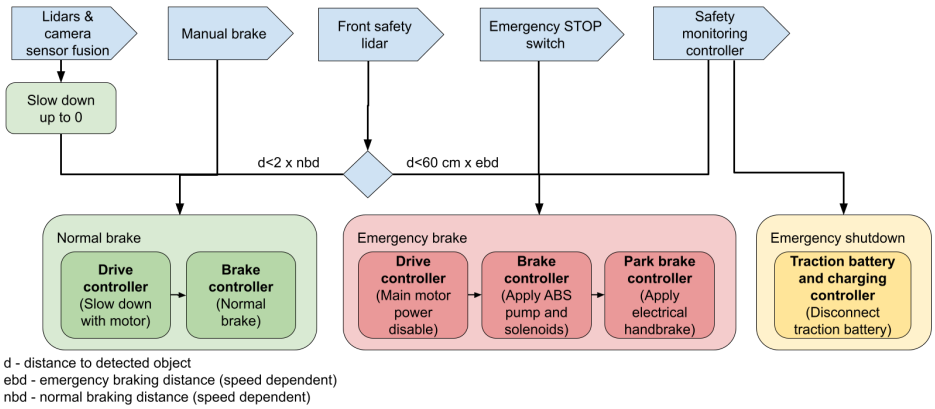

Figure 21: Safety triggering logic (Paper II).

The various testing stages outlined in reference [124] are essential for assessing and confirming the integrity of the safety system. Testing and validating a distributed CPS's

physical and cyber components pose a significant challenge. This comprehensive process encompasses hardware and software evaluation, computer and communication trials, integration validation, and a thorough review of the entire system. Subsequently, experimental procedures across these testing tiers are the following [122]:

- The functionality of all hardware components of our safety system, including the safety controller, relays, and remotes, were individually tested based on the system requirements. This step examined and monitored hardware, especially the GPIO, to verify their operations. After 2 times testing and finding some faults in the design and the performance, all the test criteria in the third test finally passed. Several modifications were made during the tests to the initial hardware schematic related to the noise removal issues. Also, software and computation testing was performed first using simulations and then through implementation on the hardware to check the validity and performance of the control algorithms. Overall, the software part was optimized during the experiment to ensure a high safety factor.

- CAN, Ethernet, and 5G internet networks are the main connection protocols between our safety system components. Each of these protocols was tested individually at least 3 times by sending predefined messages through them to the corresponding unit to ensure the correct functionality of the different ports. No error was observed during the tests. Specifications may vary for other vehicles and AMRs. CAN bus definitions are stored in databases known as DBC files, which provide a standardized format for defining the messages and signals transmitted over the CAN bus. These files are essential for ensuring that different devices on the network can communicate effectively and understand the data being exchanged. When an Ethernet or wireless network is used for communication, the data is transmitted as UDP packets.

- Integration testing is a part of software testing that involves combining individual modules in a group. Thus far, all units were tested solely, but integrating the modules before putting them in the whole system enables us to find possible errors easily. In this case, integration testing carried out three consecutive tests to check the operation of the remote, safety controller, and relays. No malfunction was observed in these units.

- System testing is meant for testing a fully integrated system as all units are integrated to satisfy the overall system requirements. In the final step, all units, including the safety controller, remotes, and relays, were mounted on the platform (AV shuttle iseAuto) to have a fully integrated system for examination. 18 experiments were carried out in two weeks and different scenarios. Examinations were categorized into two main groups. The first was testing the system under normal operation, and the second was under the false data injection.

In normal mode, the vehicle operates regularly, and the safety controller reacts based on its monitoring and safety functions. Eight comprehensive test scenarios were defined, and all safety features, such as emergency button operations, remote control practicality, and signal monitoring, were tested. Some additional monitoring features were developed, such as adding crash sensors to automatically stop the vehicle immediately in case of any crash and a backup battery voltage monitoring to avoid losing the vital power source that runs the safety features.

In conclusion, the safety controller was previously built and tested on the TalTech iseAuto platform. Later, a LoRa-based remote control switch was meant to be used near

the vehicle, and the over-the-internet safety button was added. The results were analyzed as requested on RT2, and new overall safety-related design concepts were included in the framework proposed in Chapter 2. Advanced fault detection and recovery mechanisms for low-speed AVs are listed and validated so that these algorithms can produce safe and efficient behavior across various driving scenarios, referred to in RH2.

3.4 Transition of the Autonomous Vehicle Low-Level Control System

An advanced open-source platform-based low-level control system allows the successful construction of a full-scale AV shuttle named iseAuto [105] and a warehouse AMR named BoxBot [107]. RT4 requires validating the developed framework to transfer the low-level control system from one shuttle to another.

Converting the existing self-driving shuttle into an open-source solution comprises several design stages. The vehicle manufacturer has not disclosed any details about the vehicle's performance or technical solutions. The original shuttle Navya Evo, operated through a joystick, lacked the capability for AD. To accomplish this, an existing in-house developed Gateway ECU [125] is utilized as the central control unit. Additional software layers are added to the Gateway ECU to facilitate the integration of vehicle-specific messages. The updated hardware architecture is shown in Figure 22.

Three distinct CAN buses are identified and connected to the Gateway ECU: CAN1 for traction and battery, CAN2 for steering, and CAN3 for body-related systems. Furthermore, a new control computer equipped with open-source software (Ubuntu, ROS, Autoware) is introduced into the system. An Ethernet network is established to interconnect this new control computer with the existing lidars, cameras, and a mobile internet access point.

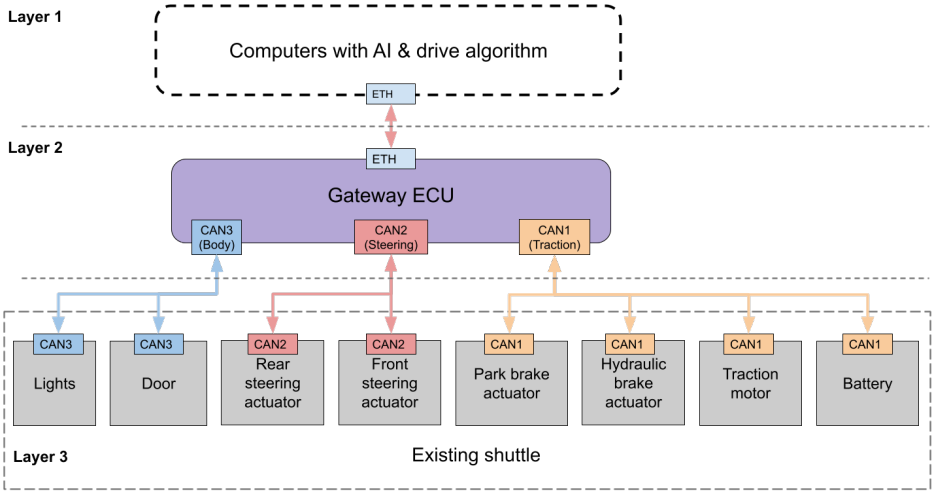

Figure 22: Updated hardware architecture for the shuttle (Paper III).

Once these modifications are implemented, the self-driving shuttle becomes operational and capable of driving either in self-driving mode or under human control. The subsequent focus lies in fine-tuning and testing the vehicle to achieve AD capabilities. An extensive two-month testing phase within a specific city district, following a prescribed 1.1-kilometer route illustrated in Figure 24.

Our experimental evaluation of the Gateway ECU aims to analyze its performance within the context of the AV's old control system. Comparative analysis is conducted us-

ing the original software that accompanied the shuttle and a custom software solution explicitly designed for this study. These tests are carried out along a predefined section of the shuttle's route, and throughout the experiment, the steering data is recorded, as illustrated in Figure 23.

The results, as depicted in Figure 23, demonstrate that the steering angle achieved with custom Gateway ECU consistently outperforms the steering provided by the original software and satisfies RH1. This superior performance is characterized by a smoother trajectory, suggesting enhanced precision and control over the shuttle's movements.

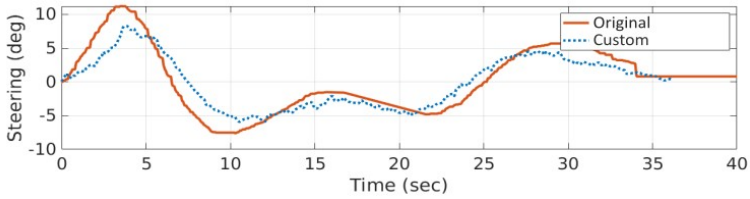

Figure 23: The steering angle of the original software and with a custom controller (Paper III).

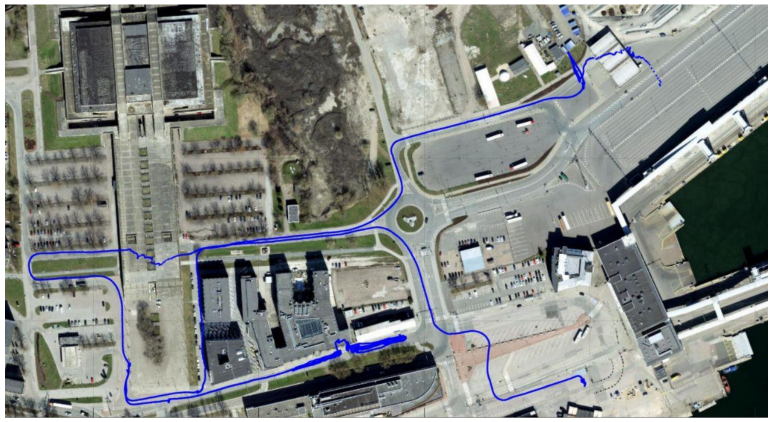

Figure 24: The Designated route that extended for a distance of 1.1 kilometers was recorded in the database (Paper III).

In conclusion, the evaluation strongly suggests that the custom Gateway ECU has the potential to significantly enhance AV steering performance compared to its original software counterpart. This finding highlights the importance of software optimization in achieving smoother AV operations and increased reliability, ultimately contributing to the advancement of autonomous transportation technologies RH1. Specific tests are required to register the autonomous shuttle as a legal vehicle in Estonia. These include verifying the reliability of the shuttle's control system by temporarily disconnecting specific system components. These requirements were considered during the development of the updated safety concept. Each module within the system performs a function related to safety. The new control system completed all initial tests to ensure the AV's safety.

3.5 Autonomous Driving Validation and Verification Using Digital Twins

The initial model of the Taltech iseAuto was utilized and subjected to ongoing development processes to implement its DT effectively. This DT served as the DUT in environments allowing the high-level control system's V&V. These environments were designed

to simulate a wide range of operational scenarios, allowing for comprehensive testing and fine-tuning of the shuttle's control systems. The objective was to ensure that the DT accurately reflected the shuttle's real-world performance and to identify any potential discrepancies or areas for improvement [126, 127].

The next objective is to develop a simulated low-level CPS within a DT, as requested in RT4. Achieving this objective requires determining effective simulation methods that can accurately replicate the behavior and interactions of the CPS components. A robust approach involves using MATLAB and Simulink to construct the ECU logic and algorithms through Simulink block diagrams. This method allows for a visual and modular representation of the system's functionality.

By importing a DBC file into these Simulink blocks, one can replicate the logic of the original controller, ensuring that the simulated system closely mirrors the real-world coun-terpart. This process involves mapping the signals and messages defined in the DBC file to the corresponding blocks in Simulink, thereby creating a comprehensive simulation model.

However, it is essential to note that there is a functionality disparity between the con-troller's C-written software and the Simulink models. This disparity arises because the C-written software is typically optimized for performance and may include low-level hard-ware interactions that are not easily replicated in Simulink. Addressing this gap requires careful V&V to ensure that the simulated model accurately reflects the behavior of the actual system. This step is crucial for identifying potential issues and making necessary improvements before deploying the system in a real-world environment.

The proposed setup enables testing of stand-alone ECUs or vehicle subsystems in a HIL environment when the vehicle self-drives inside a simulation and simultaneously generates all the traffic on the data networks. Such a test system facilitates easy and rapid validation for developing control modules and simulating the entire system operation. Different designed situations and disturbances allow for various tests and provide testing scenarios that would be too hazardous to conduct in real traffic scenarios. Tests can run for extended periods to control stability and durability. Furthermore, the parameters of an actual vehicle can be compared against the model, and any discrepancies between the vehicle and the DT in response to the same input might indicate a possible fault.

The mission computer layer allows AD decisions based on the sensor's input layer or control with a joystick that sends signals inside UDP packets to the Gateway ECU. Function-based controllers are classified as critical or non-critical. Critical controllers are involved in the direct control of the vehicle. The simulated model has two critical ECUs: a Traction motor control ECU and a Steering ECU. Two non-critical simulated controllers are the Instrument Cluster and Vehicle Speed Feedback ECU.

The Gateway ECU model is one of the most critical, and for communicating with the ROS-based mission computer, the ROS to the Gateway ECU odometry packet [128] is analyzed. For example, the fragment of the ROS to the Gateway ECU odometry packet protocol is shown in Table 4. The length of the UDP odometry packet is 45 bytes. The first bytes are the protocol version, indicating the packet generation. Then, packet length and Unix timestamp were followed and used for the packet validation process. Signals are defined as signal ID and signal itself. Signal data types may vary.

The first protocol version and other constants, for example, velocity ID, are localized and checked with a separate UDP client. Then, Simulink block UDP Receive was taken from the Instrument Control Toolbox, creating a connection for UDP incoming data. Next arrived data unbacked. This step separates the signals, which can then be forwarded to

Table 4: ROS to the Gateway ECU odometry packet

No.	Byte	Name	Data type	Unit
1	0-1	Protocol version	uint16	-
2	2-3	Data length	uint16	bytes
3	4-11	Unix timestamp	uint64	ms
4	12	Velocity ID	uint8	-
5	13-16	Velocity	float32	ms
6	17	Steering ID	uint8	-
7	18-21	Steering angle	float32	rad
8	22-45	Other signals	-	-

the simulated CAN bus. Vehicle Network Toolbox [129] blocks can create one or multiple simulated CAN buses. CAN pack or CAN unpack block takes a DBC file as input, which defines signals inside one packet and packet ID. Multifunctional DBC files can also be used as CAN databases, documentation, or real-time visualization of signals when real-time traffic is captured. The simulated Gateway ECU block diagram is shown in Figure 25.

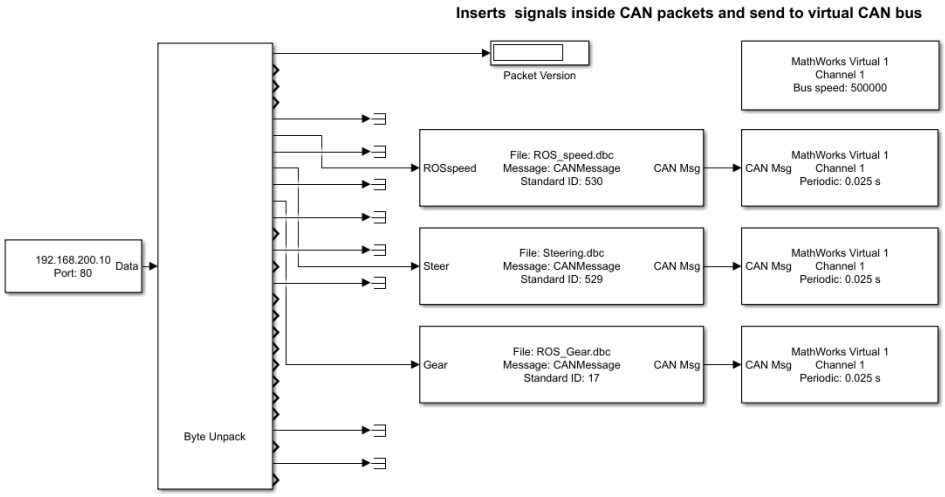

Figure 25: Simulated Gateway ECU block diagram in Matlab Simulink.

The next primary goal is the HIL simulation shown in Figure 26. In this case, the external hardware or ECU as DUT is connected to the simulation, and the simulation provides all the control signals needed for the ECU as is inside the vehicle. A CAN Bridge is developed to route the virtual CAN bus traffic to the physical CAN bus and vice versa. The traction motor is chosen as the DUT. A smaller BLDC motor controlled by a CAN-enabled Vedder electronic speed control was used to simplify the task [130]. For the feedback, an encoder was connected to the system that uses an alternative interface instead of the CAN bus.

MATLAB and Vehicle Network Toolbox supports sending and receiving messages via CAN bus from multiple manufacturers: Kvaser [131], National Instruments [132], PEAK-System [133], Vector [134], and if it is a Linux platform, then even using SocketCAN [135] interface. To get the computer running the simulation connected to the external can bus, one of the cheapest options, the PEAK PCAN USB adapter [136], was used. The CAN Bridge

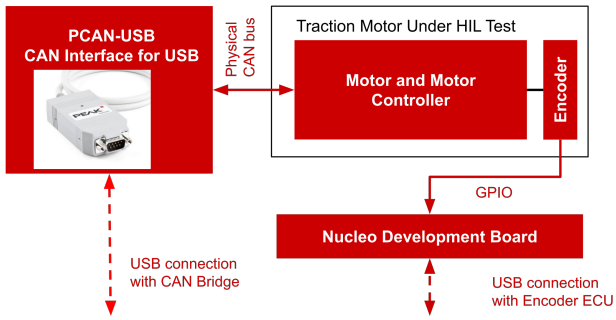

Figure 26: Hardware-in-loop simulation.

transfers data between virtual and physical CAN buses. Additionally, there are no limitations when connecting multiple CAN adapters. This flexibility allows for the integration of different virtual CAN buses as needed. By utilizing multiple CAN adapters, more complex and versatile networks can developed, accommodating various communication requirements.

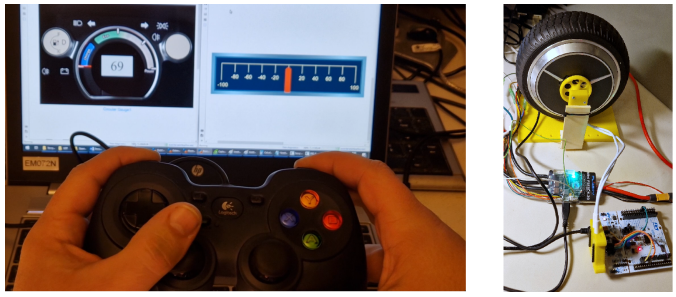

Figure 27: HIL test setup.

As shown in Figure 27, feedback is essential for measuring rotational speed and other parameters in HIL simulation. The ECU can estimate and transmit these parameters over the CAN bus, depending on the scenario. Alternatively, separate sensors or signals may be necessary when interfacing with a DUT. One approach is to use a microcontroller development board supported by MATLAB, connected to an encoder. The model can then run in MATLAB's external mode with serial communication. In this setup, a USB serial interface facilitates communication with the board, and GPIO pins allow direct reading or writing. Additionally, the developed algorithm converts pulses into speed signals sent to a simulated CAN bus as a vehicle speed signal for use by other ECUs.

In conclusion, RT6 requires analyzing and exploring of various V&V methodologies, which can be integrated into the framework to validate the functionality and performance of the AVs' low-level control systems using the DT concept. A set of the vehicle's controllers is simulated. While the simulation is running, traffic is generated on a simulated data network that can be used to test and develop physical controllers. A developed low-level CPS simulation is a proof-of-concept solution that can be expanded and has a vital part to play with Taltech iseAuto 2.0 development.

4 Discussion

The theoretical aspects of this study are primarily concerned with the methodologies and equipment utilized for conducting additional research in both field and laboratory settings to allow other researchers to reproduce the core findings of the initial study. By implementing these strategies, researchers can accelerate the adoption of low-level control systems in future autonomous transportation systems.

At first, the methodology for creating a modular, flexible, open-source AMR called BoxBot was discussed. The robot was tested in a food and grocery production factory in Kulinaaria to show its ability to transport packages without human interference. Researchers can use this study to develop a modular architecture for different conditions. A significant benefit of Paper I is related to the changing environment and the interaction between humans and robots. The development included improved electronic design concepts and data bus concepts into the framework to enable better error detection and reconfigurability.

Second, the analysis of real-time fault monitoring and crash detection for distributed low-level CPS finds methods to improve the existing safety controller with wireless remote control switches capable of managing emergency shutdowns. After the platform's initial development, enhancements like crash sensors and battery voltage monitoring were added. Later, the new overall safety-related design concepts were included in the framework with the main idea of adding safety features to any ECUs.

Thirdly, the risks identified are universal and applicable to other low-speed automated vehicles, such as AVs and indoor AMRs (Paper II). This enables researchers to simplify decision-makers' judgments and handle the uncertainty caused by these judgments. The developed MCDM risk evaluation model allows researchers to assess safety systems. The results analyzed provide input for further developments and improvements of AVs. Automotive standards were added to the framework, which enables future ECUs to be designed with significantly greater reliability.

Fourthly, research continues to transform the existing self-driving shuttle into an open-source solution. This involved mapping the low-level architecture and identifying control computers and data connections. Logging and analyzing CAN messages allows for the extraction of essential data. Paper III contributes to future efforts to make the shuttle street-legal while ensuring compliance with safety standards and regulations. Requirements were taken into account during the development of the updated safety concept. Also, the framework was updated with fleet orchestration architecture concepts.

Previous work has shown a research gap in how to V&V low-level CPS. Paper IV primarily focuses on DTs in addressing V&V challenges associated with the principal components of AVs. The study shows that building TDs with a high predictive value for low-level CPS is possible. High-fidelity simulation can be extended with simulated low-level CPS, including simulated ECUs and data buses, generating simulated communication inside databuses, proper for HIL testing, development, and V&V. Researchers can use these low-level control validation methodologies for CPS components, which V&V usually require substantial labor and effort.

In conclusion, this study provides a comprehensive overview of the hypotheses and the corresponding analyses. The findings from these analyses contribute significantly to the ongoing efforts to enhance the practical design, validation, and verification framework for vehicle hardware and software components of AVs. The validation process ensures that the framework is robust and capable of handling the complexities of different AV and AMR platforms.

5 Conclusion and Future Research

The doctoral research topic targets an approach that focuses on CPS and algorithms. RH1 and RT3 stated that a more modular, user-friendly, and expandable low-level control system framework was developed, including electronic design, data buses, operational security, regulation, and standards. V&V methods are viewed as deploying DT of low-level CPS.

First, the modular framework provides developers with comprehensive methodologies and design principles to address weaknesses and issues in designing low-level control systems for AV projects, particularly self-driving platform Taltech iseAuto and small in-house AMRs. The accomplishment also shows how to integrate proven autonomous technologies into various older AV models, extending their useful lifetime and reducing costs, as demonstrated in case studies referenced in Chapters 3.1 and 3.4.

Second, the framework includes a risk analysis model referenced in RH3 with non-functional aspects, such as real-time responsiveness, sensor reliability, communication robustness, and environmental uncertainties. Beyond functional safety, risk prioritization for low-level and high-level control algorithms is expected to significantly enhance the overall safety of AVs across diverse use cases, as proven in the case study referenced in Chapter 3.2.

Third, the validation and verification challenges related to the key components of AVs will be tackled. By conducting an exhaustive review of existing methodologies, this study sheds light on the relationship between the process of creating DTs and the vital task of ensuring the reliability of safety-critical systems. The complexities of constructing a low-level control system model within a simulated vehicle, as referenced in RH4, encompass both high-level and low-level control systems and data buses. These efforts aim to refine modeling capabilities, improve predictive accuracy, and address the identified limitations, as validated in Chapter 3.4. By addressing these challenges, the goal is to create a robust and reliable simulation environment that can be used to test and validate different aspects of the vehicle's control systems. This not only enhances the accuracy of the simulations but also provides valuable insights that can be used to improve the design and implementation of the actual systems.

The developed framework provides a structured approach to CPS development, addressing the complexities and challenges of creating sophisticated mobile control systems. It includes detailed guidelines for module development. By adhering to established standards and best practices, the framework ensures that the developed CPSs are effective and compliant with regulatory requirements. This approach facilitates the development of advanced AVs and AMRs, contributing to the advancement of autonomous technology and its safe integration into real-world applications, as proved by practical case studies of implementing a CPS on different AVs. Using the developed framework, the effectiveness of the development has increased by an estimated 20%. With a team of just 10-20 master's students, support from private companies, and a total budget of 500,000 Euros, a fully functional AV shuttle prototype was successfully showcased. This is a fraction of the cost OEM automakers pay. The findings of this thesis have contributed to improving the safety and reliability of the TalTech iseAuto 2.0.

The modular framework is subject to ongoing refinement. This research will continue to improve the framework. Future research can explore and expand the methodologies introduced in this thesis:

- Modern automotive standard-based modular ECU architecture, including low-level control algorithms, novel methods for error checking, and configuration capabil-

ity, decreases the development time. Modularity can be achieved by developing a distinct set of types of ECUs, each capable of handling specific tasks.

- The proposed methods showed that creating DTs with a high predictive value for low-level CPS is possible. The current behavior of simulated low-level CPS, including simulated ECUs and data buses, can be expanded for more predictive XIL testing, development, and V&V.
- The research explores whether there are more effective methods for conducting low-level CPS simulations. For instance, simulating a microcontroller and executing the same compiled firmware gives a higher fidelity level. In such scenarios, determining how to address the input/output (I/O) challenges becomes one of the most significant questions.
- AI-based low-level driver assistance, like systems for the mission computer, allows fault detection, which can provide novel opportunities to improve security and reliability.
- Manufacturers, such as the Automotive Open System Architecture (AUTOSAR), use standards-based frameworks for ECU hardware and software development. Most automotive companies are relatively conservative about open source, restricting the availability of AUTOSAR to the general research and education community. Research on the topic of whether it would be possible to overcome it.
- One of the most challenging research topics is replacing the OEM's low-level vehicle platform with in-house development, providing new research opportunities as there would no longer be closed-source and undocumented systems.
- Completing TalTech's iseAuto v2.0 case study and achieving street-legal status for the new shuttle as fast as possible using the developed framework is a challenging task to ensure strict adherence to safety standards and regulations.

List of Figures

List of Tables

References

[1] D. Pojani and D. Stead, "Sustainable urban transport in the developing world: Beyond megacities," *Sustain. Sci. Pract. Policy*, vol. 7, pp. 7784–7805, June 2015.

[2] A. Broggi, *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*. World Scientific, 1999.

[3] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer, Nov. 2009.

[4] B. A. Guvenc, L. Guvenc, E. S. Ozturk, and T. Yigit, "Model regulator based individual wheel braking control," in *Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003.*, vol. 1, pp. 31–36 vol.1, IEEE, 2003.

[5] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, 2014.

[6] L. Masello, G. Castignani, B. Sheehan, F. Murphy, and K. McDonnell, "On the road safety benefits of advanced driver assistance systems in different driving contexts," *Transportation Research Interdisciplinary Perspectives*, vol. 15, p. 100670, Sept. 2022.

[7] A. S. Mueller, J. B. Cicchino, and D. S. Zuby, "What humanlike errors do autonomous vehicles need to avoid to maximize safety?," *J. Safety Res.*, vol. 75, pp. 310–318, Dec. 2020.

[8] M. Maurer, J. Christian Gerdes, B. Lenz, and H. Winner, *Autonomous Driving: Technical, Legal and Social Aspects*. Springer Berlin Heidelberg, May 2016.

[9] I. Nastjuk, B. Herrenkind, M. Marrone, A. B. Brendel, and L. M. Kolbe, "What drives the acceptance of autonomous driving? an investigation of acceptance factors from an end-user's perspective," *Technol. Forecast. Soc. Change*, vol. 161, p. 120319, Dec. 2020.

[10] H. Pikner and K. Karjust, "Multi-layer cyber-physical low-level control solution for mobile robots," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1140, p. 012048, May 2021.

[11] R. Baheti and H. Gill, "Cyber-physical systems," *The impact of control technology*, vol. 12, no. 1, pp. 161–166, 2011.

[12] Y. Guo, X. Hu, B. Hu, J. Cheng, M. Zhou, and R. Y. K. Kwok, "Mobile cyber physical systems: Current challenges and future networking applications," *IEEE Access*, vol. 6, pp. 12360–12368, 2018.

[13] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pp. 363–369, IEEE, May 2008.

[14] "ROS: Home." `https://www.ros.org/`. Accessed: 2024-5-14.

[15] E. Ackerman, "Lidar that will make self-driving cars affordable [news]," *IEEE Spectrum*, vol. 53, pp. 14–14, Oct. 2016.

[16] H.-P. Schoener, "Automotive applications of unconventional actuators," 2001.

[17] S. M. Castano, L. N. Barragan, C. C. Rodriguez, and A. Javier Maixé, "Design of a brushless DC motor for an automotive application: A comparative evaluation with a commercial model," in *2012 Workshop on Engineering Applications*, pp. 1–6, IEEE, May 2012.

[18] E. Kouicem, C. Raievsky, and M. Occello, "Artificial emotions for distributed cyber-physical systems resilience," in *Proceedings of the Cyber-Physical Systems PhD Workshop 2019*, 2019.

[19] A. Y. Zalozhnev and V. N. Ginz, "Industry 4.0: Underlying technologies. industry 5.0: Human-Computer interaction as a tech bridge from industry 4.0 to industry 5.0," in *2023 9th International Conference on Web Research (ICWR)*, pp. 232–236, IEEE, May 2023.

[20] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, Jan. 2015.

[21] T. T. Mezgebe, M. G. Gebreslassie, H. Sibhato, and S. T. Bahta, "Intelligent manufacturing eco-system: A post COVID-19 recovery and growth opportunity for manufacturing industry in Sub-Saharan countries," *Sci Afr*, vol. 19, p. e01547, Mar. 2023.

[22] A. Mathur, A. Dabas, and N. Sharma, "Evolution from industry 1.0 to industry 5.0," in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pp. 1390–1394, IEEE, Dec. 2022.

[23] D. P. F. Möller, H. Vakilzadian, and R. E. Haas, "From industry 4.0 towards industry 5.0," in *2022 IEEE International Conference on Electro Information Technology (eIT)*, pp. 61–68, IEEE, May 2022.

[24] R. Sell, A. Rassolkin, R. Wang, and T. Otto, "Integration of autonomous vehicles and industry 4.0," *Proc. Eston. Acad. Sci.*, vol. 68, no. 4, 2019.

[25] B. A. Talkhestani, N. Jazdi, W. Schlögl, and M. Weyrich, "A concept in synchronization of virtual production system with real factory based on anchor-point method," *Procedia CIRP*, vol. 67, pp. 13–17, Jan. 2018.

[26] Aec, "Q100 rev. H-Failure mechanism based stress test qualification for integrated circuits," 2014.

[27] Automotive Electronics Council, "Stress test qualification for passive components," Tech. Rep. AEC - Q200 - Rev E, Component Technical Committee, Luton, UK, 2023.

[28] USCAR, "Performance specification for automotive electrical connector systems," Tech. Rep. USCAR2-7, SAE International, Jan. 2020.

[29] R. Debouk, "Overview of the second edition of ISO 26262: Functional safety— road vehicles," *Hazard Prev.*, vol. 55, pp. 13–21, Mar. 2019.

[30] O. I. de Normalización, *ISO 26262-2: Road Vehicles – Functional Safety. Management of functional safety. Gestion de la sécurité fonctionnelle*. ISO, 2018.

[31] K.-L. Lu and Y.-Y. Chen, "ISO 26262 ASIL-Oriented hardware design framework for Safety-Critical automotive systems," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 1–6, IEEE, Nov. 2019.

[32] K. Radlak, M. Szczepankiewicz, T. Jones, and P. Serwa, "Organization of machine learning based product development as per ISO 26262 and ISO/PAS 21448," in *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 110–119, IEEE, Dec. 2020.

[33] T. Sae, "Taxonomy and definitions for terms related to driving automation systems for On-Road motor vehicles," Tech. Rep. J3016_202104, SAE MOBILUS, 2021.

[34] SAE International, "SAE international releases updated visual chart for its "levels of driving automation" standard for Self-Driving vehicles," tech. rep., WARRENDALE, PA., 2018.

[35] B. H. Cavazza, R. M. Gandia, F. Antonialli, A. L. Zambalde, I. Nicolaï, J. Y. Sugano, and A. D. M. Neto, "Management and business of autonomous vehicles: a systematic integrative bibliographic review," *Int. J. Automot. Technol. Manage.*, vol. 19, no. 1/2, p. 31, 2019.

[36] I. 22/sc, "Road vehicles safety of the intended functionality," Tech. Rep. ISO 21448:2022, International Organization for Standardization, June 2022.

[37] D. Ward and P. Wooderson, *Automotive Cybersecurity: An Introduction to ISO/SAE 21434*. SAE International, Dec. 2021.

[38] ISO/IEC/IEEE, "Systems and software engineering vocabulary," Tech. Rep. 24765, 2017.

[39] United Nations Economic Commission for Europe (UNECE), "UN regulation no. 157 - automated lane keeping systems (ALKS)," Tech. Rep. 3, https://unece.org/transport/documents/2021/03/standards/un-regulation-no-157-automated-lane-keeping-systems-alks, 2021.

[40] M. Bellone, A. Ismailogullari, J. Müür, O. Nissin, R. Sell, and R.-M. Soe, "Autonomous driving in the Real-World: The weather challenge in the sohjoa baltic project," in *Towards Connected and Autonomous Vehicle Highways: Technical, Security and Social Challenges* (U. Z. A. Hamid and F. Al-Turjman, eds.), pp. 229–255, Cham: Springer International Publishing, 2021.

[41] J. H. Holland, *Adaptation in natural and artificial systems an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, Mass: MIT Press, 1st mit press ed. ed., 1992.

[42] H. Yue, H. Medromi, H. Ding, and D. Bassir, "A novel hybrid drone for multi-propose aerial transportation and its conceptual optimization based on surrogate approach," *J. Phys. Conf. Ser.*, vol. 1972, p. 012103, July 2021.

[43] S. Guessasma and D. Bassir, "Neural network computation for the evaluation of process rendering: application to thermally sprayed coatings," *Int. J. Simul. Multi. Design Optim.*, vol. 8, p. A10, 2017.

[44] X. G. Tang, M. Rezoug, R. Hamzaoui, D. Bassir, R. El Meouche, J. F. Khreim, and Z. Q. Feng, "Multiobjective optimization on urban flooding using RSM and GA," *Adv. Mat. Res.*, vol. 255-260, pp. 1627–1631, May 2011.

[45] S. Guessasma and H. D. Bassir, "Comparing heuristic and deterministic approaches to optimise mechanical parameters of biopolymer composite materials," *Mech. Adv. Mater. Struct.*, vol. 16, pp. 293–299, May 2009.

[46] H. Herranen, J. Majak, P. Tsukrejev, K. Karjust, and O. Märtens, "Design and manufacturing of composite laminates with structural health monitoring capabilities," *Procedia CIRP*, vol. 72, pp. 647–652, Jan. 2018.

[47] H. Taherdoost and M. Madanchian, "Multi-Criteria decision making (MCDM) methods and concepts," *Encyclopedia*, vol. 3, pp. 77–87, Jan. 2023.

[48] T. L. Saaty, "The analytic hierarchy process: Decision making in complex environments," in *Quantitative Assessment in Arms Control: Mathematical Modeling and Simulation in the Analysis of Arms Control Problems* (R. Avenhaus and R. K. Huber, eds.), pp. 285–308, Boston, MA: Springer US, 1984.

[49] N. F. Mahad, N. Yusof, and N. F. Ismail, "The application of fuzzy analytic hierarchy process (FAHP) approach to solve multi-criteria decision making (MCDM) problems," *J. Phys. Conf. Ser.*, vol. 1358, p. 012081, Nov. 2019.

[50] R. M. Rodriguez, L. Martinez, and F. Herrera, "Hesitant fuzzy linguistic term sets for decision making," *IEEE Trans. Fuzzy Syst.*, vol. 20, pp. 109–119, Feb. 2012.

[51] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications : a State-of-the-art Survey*. Springer-Verlag, 1981.

[52] R. Rahim, A. P. U. Siahaan, R. F. Wijaya, H. Hantono, N. Aswan, S. Thamrin, D. A. P. Sari, S. Agustina, R. B. Santosa, W. M. Muttaqin, and Others, "Technique for order of preference by similarity to ideal solution (TOPSIS) method for decision support system in top management," *Pro Mark*, vol. 8, no. 2, 2018.

[53] G. Bakioglu and A. O. Atahan, "AHP integrated TOPSIS and VIKOR methods with pythagorean fuzzy sets to prioritize risks in self-driving vehicles," *Appl. Soft Comput.*, vol. 99, p. 106948, Feb. 2021.

[54] C. Ziyan and L. Shiguo, "China's self-driving car legislation study," 2021.

[55] M. Harrison, Z. Yang, T. T. Nguyen, S. Kavakeb, J. Wang, and S. Bonsall, "A TOPSIS method for vehicle route selection in seaports — a real case analysis of a container terminal in north west europe," in *2015 International Conference on Transportation Information and Safety (ICTIS)*, pp. 599–606, IEEE, June 2015.

[56] I. P. Gomes, D. R. Bruno, F. S. Osório, and D. F. Wolf, "Diagnostic analysis for an autonomous truck using multiple attribute decision making," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pp. 283–290, IEEE, Nov. 2018.

[57] S. Lu and Q. Jin, "Constructing ECU software architecture based on OSEK," in *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pp. 721–7214, IEEE, Oct. 2019.

[58] J. P. Trovao, "An overview of automotive electronics [automotive electronics]," *IEEE Veh. Technol. Mag.*, vol. 14, pp. 130–137, Sept. 2019.

[59] H. Zhang, Y. Pan, Z. Lu, J. Wang, and Z. Liu, "A cyber security evaluation framework for In-Vehicle electrical control units," *IEEE Access*, vol. 9, pp. 149690–149706, 2021.

[60] J. Bortolazzi, T. Hirth, and T. Raith, "Specification and design of electronic control units," in *Proceedings EURO-DAC '96. European Design Automation Conference with EURO-VHDL '96 and Exhibition*, pp. 36–41, IEEE, 1996.

[61] G. Reichart, "Home AUTOSAR." `https://www.autosar.org/`, Feb. 2024. Accessed: 2024-2-28.

[62] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, and W. Shi, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 8, pp. 6469–6486, Apr. 2021.

[63] K. C. Wang, *Embedded and Real-Time Operating Systems*. Springer Nature, Sept. 2023.

[64] S. C. Hpl, "Introduction to the controller area network (CAN)," *Application Report SLOA101*, pp. 1–17, 2002.

[65] K. Kalaiyarasu and C. Karthikeyan, "Design of an automotive safety system using controller area network," 2015.

[66] M. Ruff, "Evolution of local interconnect network (LIN) solutions," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, vol. 5, pp. 3382–3389 Vol.5, IEEE, 2003.

[67] H. C. Wense, "Introduction to local interconnect network," *SAE Trans. J. Mater. Manuf.*, vol. 109, pp. 87–91, Mar. 2000.

[68] P. Hank, S. Müller, O. Vermesan, and J. Van Den Keybus, "Automotive ethernet: In-vehicle networking and smart mobility," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1735–1739, IEEE, Mar. 2013.

[69] I. Park and M. Sunwoo, "FlexRay network parameter optimization method for automotive applications," *IEEE Trans. Ind. Electron.*, vol. 58, pp. 1449–1459, Apr. 2011.

[70] S.-H. Seo, S.-W. Lee, S.-H. Hwang, and J. W. Jeon, "Development of network gateway between CAN and FlexRay protocols for ECU embedded systems," in *2006 SICE-ICASE International Joint Conference*, pp. 2256–2261, IEEE, Oct. 2006.

[71] H.-Y. Hwang, S.-M. Oh, and J. Shin, "CAN gateway for fast vehicle to vehicle (V2V) communication," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 523–525, IEEE, Oct. 2015.

[72] W. He, H. Li, X. Zhi, X. Li, J. Zhang, Q. Hou, and Y. Li, "Overview of V2V and V2I wireless communication for cooperative vehicle infrastructure systems," in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 127–134, IEEE, Dec. 2019.

[73] K. Wang, Z. Gong, Y. Hou, M. Zhang, C. Liu, and R. Chen, "Model based design and procedure of flight control system for unmanned aerial vehicle," in *2020 3rd International Conference on Unmanned Systems (ICUS)*, pp. 763–768, IEEE, Nov. 2020.

[74] R. Hýl and R. Wagnerová, "Fast development of controllers with simulink coder," in *2017 18th International Carpathian Control Conference (ICCC)*, pp. 406–411, IEEE, May 2017.

[75] B. R. Mudhivarthi, V. Saini, A. Dodia, P. Shah, and R. Sekhar, "Model based design in automotive open system architecture," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1211–1216, IEEE, May 2023.

[76] M. Maddeh, S. Al-Otaibi, S. Alyahya, F. Hajjej, and S. Ayouni, "A comprehensive MCDM-Based approach for Object-Oriented metrics selection problems," *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 13, p. 3411, Mar. 2023.

[77] N. Kalra and S. M. Paddock, *Driving to Safety: How Many Miles of Driving Would it Take to Demonstrate Autonomous Vehicle Reliability?* RAND Corporation, 2016.

[78] E. Thorn, S. Kimmel, and M. Chaka, "A framework for automated driving system testable cases and scenarios," Tech. Rep. DOT HS 812 623, Virginia Tech Transportation Institute, Sept. 2018.

[79] J. A. Matute-Peaspan, A. Zubizarreta-Pico, and S. E. Diaz-Briceno, "A vehicle simulation model and automated driving features validation for Low-Speed high automation applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, pp. 7772–7781, Dec. 2021.

[80] W. Wachenfeld and H. Winner, "The release of autonomous vehicles," *Autonomous Driving: Technical, Legal and Social Aspects*, pp. 425–449, 2016.

[81] J. Tao, Y. Li, F. Wotawa, H. Felbinger, and M. Nica, "On the industrial application of combinatorial testing for autonomous driving functions," in *2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 234–240, IEEE, Apr. 2019.

[82] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, Sept. 2020.

[83] Autoware Foundation, "TIER IV AWSIM," 2022.

[84] "Rosbridg package summary." `https://wiki.ros.org/rosbridge_suite`. Accessed: 2024-5-14.

[85] M. Malayjerdi, Q. A. Goss, M. İ. Akbaş, R. Sell, and M. Bellone, "A Two-Layered approach for the validation of an operational autonomous shuttle," *IEEE Access*, vol. 11, pp. 89124–89137, 2023.

[86] A. Roberts, M. Malayjerdi, M. Bellone, O. Maennel, and E. Malayjerdi, "Analysing adversarial threats to rule-based local-planning algorithms for autonomous driving," in *Proceedings Inaugural International Symposium on Vehicle Security & Privacy*, (Reston, VA), Internet Society, 2023.

[87] M. Short and M. J. Pont, "Hardware in the loop simulation of embedded automotive control system," in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pp. 426–431, IEEE, 2005.

[88] D. Ekert, J. Dobaj, and A. Salamun, "Cybersecurity verification and validation testing in automotive," *jucs*, vol. 27, pp. 850–867, Aug. 2021.

[89] A. V. Tumasov, A. S. Vashurin, Y. P. Trusov, E. I. Toropov, P. S. Moshkov, V. S. Kryaskov, and A. S. Vasilyev, "The application of Hardware-in-the-Loop (HIL) simulation for evaluation of active safety of vehicles equipped with electronic stability control (ESC) systems," *Procedia Comput. Sci.*, vol. 150, pp. 309–315, Jan. 2019.

[90] "SAE MOBILUS." `https://saemobilus.sae.org/content/j3016_202104`. Accessed: 2024-2-13.

[91] M. della Cava and U. Today, "Tesla announces fully self-driving cars," *USA Today*, Oct. 2016.

[92] K. Korosec, "Ford postpones autonomous vehicle service until 2022," *TechCrunch*, Apr. 2020.

[93] A. Rassõlkin, T. Vaimann, A. Kallaste, and R. Sell, "Propulsion motor drive topology selection for further development of ISEAUTO Self-Driving car," in *2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pp. 1–5, IEEE, Nov. 2018.

[94] H. Pikner, R. Sell, J. Majak, and K. Karjust, "Safety system assessment case study of automated vehicle shuttle," *Electronics*, vol. 11, p. 1162, Apr. 2022.

[95] R. Sell, E. Malayjerdi, M. Malayjerdi, and B. C. Baykara, "Safety toolkit for automated vehicle shuttle -practical implementation of digital twin," in *2022 International Conference on Connected Vehicle and Expo (ICCVE)*, pp. 1–6, IEEE, Mar. 2022.

[96] "Self-driving shuttle for passenger transportation." `https://www.navya.tech/en/solutions/moving-people/self-driving-shuttle-for-passenger-transportation/`, June 2020. Accessed: 2024-5-14.

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

[98] A. Rassõlkin, R. Sell, and M. Leier, "Development case study of the first estonian self-driving car, iseauto," *Electrical, Control and Communication Engineering*, vol. 14, no. 1, pp. 81–88, 2018.

[99] R. Sell, E. Coatanéa, and F. Christophe, "Important aspects of early design in mechatronic," in *Proceedings of the 6th international conference of DAAAM Baltic industrial engineering* (R. Küttner, ed.), 2008.

[100] E. Väljaots and R. Sell, "Unmanned ground vehicle SysML navigation model conducted by energy efficiency," in *Advanced Materials Research*, vol. 905, pp. 443–447, 2014.

[101] R. Sell, M. Leier, A. Rassõlkin, and J.-P. Ernits, "Autonomous last mile shuttle ISEAUTO for education and research," *IJAIML*, vol. 10, pp. 18–30, Jan. 2020.

[102] D. Goswami, R. Schneider, A. Masrur, M. Lukasiewycz, S. Chakraborty, H. Voit, and A. Annaswamy, "Challenges in automotive cyber-physical systems design," in *2012 International Conference on Embedded Computer Systems (SAMOS)*, pp. 346–354, IEEE, July 2012.

[103] H. A. Al-Fedhly and W. ElMaraghy, "Design methodology framework for cyber-physical products," *International Journal of Industry and Sustainable Development*, vol. 1, no. 2, pp. 64–75, 2020.

[104] J. Friedman, "MATLAB/Simulink for automotive systems design," in *Proceedings of the Design Automation & Test in Europe Conference*, vol. 1, pp. 1–2, IEEE, 2006.

[105] S. Chakraborty, M. A. Al Faruque, W. Chang, D. Goswami, M. Wolf, and Q. Zhu, "Automotive Cyber–Physical systems: A tutorial introduction," *IEEE Des. Test Comput.*, vol. 33, pp. 92–108, Aug. 2016.

[106] L. Rosencrance, G. Lawton, and C. Moozakis, "User datagram protocol (UDP)." `https://www.techtarget.com/searchnetworking/definition/UDP-User-Datagram-Protocol`, Dec. 2023. Accessed: 2024-5-14.

[107] A. Chattopadhyay, K.-Y. Lam, and Y. Tavva, "Autonomous vehicle: Security by design," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, pp. 7015–7029, Nov. 2021.

[108] A. Nardi and A. Armato, "Functional safety methodologies for automotive applications," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 970–975, IEEE, Nov. 2017.

[109] Y. Chung and Y.-P. Yang, "Hardware-in-the-Loop simulation of Self-Driving electric vehicles by dynamic path planning and model predictive control," *Electronics*, vol. 10, p. 2447, Oct. 2021.

[110] Z. Bi, G. Xu, C. Wang, G. Xu, and S. Zhang, "A method for translating automotive Body-Related CAN messages based on labeled bits," *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 13, p. 1942, Feb. 2023.

[111] K. Karjust, J. Majak, H. Pikner, and R. Sell, "Multi-layer cyber-physical control method for mobile robot safety systems," *Proceedings of the Estonian Academy of Sciences*, vol. 70, no. 4, pp. 383–391, 2021.

[112] S. Vinodh, M. Prasanna, and N. Hari Prakash, "Integrated fuzzy AHP–TOPSIS for selecting the best plastic recycling method: A case study," *Appl. Math. Model.*, vol. 38, pp. 4662–4672, Oct. 2014.

[113] S. Kaganski, J. Majak, and K. Karjust, "Fuzzy AHP as a tool for prioritization of key performance indicators," *Procedia CIRP*, vol. 72, pp. 1227–1232, Jan. 2018.

[114] M. Paavel, K. Karjust, and J. Majak, "PLM maturity model development and implementation in SME," 2017.

[115] M. Paavel, K. Karjust, and J. Majak, "Development of a product lifecycle management model based on the fuzzy analytic hierarchy process," 2017.

[116] R. Sell, E. Väljaots, T. Pataraia, and E. Malayjerdi, "Modular smart control system architecture for the mobile robot platform," *Proc. Eston. Acad. Sci.*, vol. 68, no. 4, pp. 395–400, 2019.

[117] 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)*, pp. 287–296, Apr. 2018.

[118] T. Pivoňka, R. Sell, H. Pikner, and L. Přeučil, "Fiducial Marker-Based monocular localization for autonomous docking," *IFAC-PapersOnLine*, vol. 56, pp. 2957–2962, Jan. 2023.

[119] T. Raamets, J. Majak, K. Karjust, K. Mahmood, and A. Hermaste, "Autonomous mobile robots for production logistics: a process optimization model modification," *Proc. Eston. Acad. Sci.*, 2024.

[120] T. Raamets, J. Majak, K. Karjust, K. Mahmood, and A. Hermaste, "Development of process optimization model for autonomous mobile robot used in production logistics," 2024.

[121] R. Sell and A. Petritsenko, "Early design and simulation toolkit for mobile robot platforms," *International Journal of Product*, 2013.

[122] H. Pikner and M. Malayjerdi, "Cyber-physical universal safety and crash detection system for autonomous robot," *Robotic Systems and Applications*, vol. 1, no. 2, pp. 46–52, 2021.

[123] B. Carneiro, D. M. Batista, R. H. Junior, and K. Ullah, "A crash response system using LoRa-based V2X communications," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5, IEEE, May 2023.

[124] S. Abbaspour Asadollah, R. Inam, and H. Hansson, "A survey on testing for cyber physical system," in *Testing Software and Systems*, pp. 194–207, Springer International Publishing, 2015.

[125] R. Sell, R.-M. Soe, R. Wang, and A. Rassõlkin, "Autonomous vehicle shuttle in smart city testbed," in *Intelligent System Solutions for Auto Mobility and Beyond*, pp. 143–157, Springer International Publishing, 2021.

[126] M. Malayjerdi and R. Sell, "Scenario-based validation of safety and performance of an autonomous vehicle by a software in loop simulation method." `https://digikogu.taltech.ee/et/Item/5d3435ba-8ce1-4da6-8d16-4b279e88c861`. Accessed: 2024-5-14.

[127] E. Malayjerdi, *Advanced Autonomous Vehicle's Functions for Safety Improvements in Urban Mobility Context*. PhD thesis, Tallinn University of Technology, 2022.

[128] P. Trink, "Autonomous path following on a vehicle using an open source software autoware," Master's thesis, Tallinn University of Technology, june 2018.

[129] D. M. Albelo, R. D. Dias, R. Neves, and B. S. Paterlini, "Vehicle dynamic control using vehicle network toolbox from MATLAB/Simulink®," tech. rep., SAE Technical Paper, 2022.

[130] S. Mishra, K. P. Sunil, D. Prakash, K. M. Umar, V. Kshartiya, and Y. Gowda, "Design solutions for Off-Road electric skateboards," in *2021 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, pp. 55–60, IEEE, June 2021.

[131] "Kvaser website." https://www.kvaser.com/. Accessed: 2024-5-14.

[132] "National instruments website." https://www.ni.com/en.html. Accessed: 2024-5-14.

[133] PEAK-System Technik GmbH, "Home: PEAK-system." https://www.peak-system.com/Home.59.0.html?&L=1. Accessed: 2024-5-14.

[134] "About vector." https://www.vector.com/int/en/company/about-vector/. Accessed: 2024-5-14.

[135] M. Sojka, P. Píša, M. Petera, O. Špinka, and Z. Hanzálek, "A comparison of linux CAN drivers and their applications," in *International Symposium on Industrial Embedded System (SIES)*, pp. 18–27, IEEE, July 2010.

[136] "PCAN-USB interface." https://www.peak-system.com/PCAN-USB.199.0.html?&L=1. Accessed: 2024-5-14.

[137] H. Pikner, R. Sell, K. Karjust, E. Malayjerdi, and T. Velsker, "Cyber-physical control system for autonomous logistic robot," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pp. 699–704, IEEE, Apr. 2021.

[138] H. Pikner, R. Sell, and E. Malayjerdi, "Level 4 commercial autonomous vehicle control system transition to an open-source solution," *Proc. Eston. Acad. Sci.*, vol. 73, no. 2, pp. 124–133, 2024.

[139] H. Pikner, M. Malayjerdi, M. Bellone, B. Baykara, and R. Sell, "Autonomous driving validation and verification using digital twins," in *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, pp. 204–211, INSTICC, SciTePress, 2024.

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.

Abstract
Cyber-Physical Control System for Self-Driving Vehicles

Autonomous Vehicle (AV) technology advancements have generated significant demand for driverless transportation. The hope is to achieve better efficiency, increased traffic safety, and energy savings. However, the absence of a human driver presents a significant challenge for low-level Cyber-Physical Systems (CPSs) in these vehicles due to their increasing complexity. While a conventional vehicle's driver can assess its technical condition using vibrations, sounds, and visual information displayed by the car, an AV may not have a driver on board, or the driver may be located remotely. As a result, all issues must be detected through sensor data and feedback from various systems.

The thesis explores the low-level CPS of AVs and various aspects of low-level CPS development. These aspects include module development, validation of modules, simulation, and risk analysis. These stages must adhere to standards, legislation, and established practices. The thesis's primary goal is to create a comprehensive framework encompassing different methods for developing CPSs, which can be applied in AV and Autonomous Mobile Robot (AMR) development with levels 4 and 5 autonomy, as defined by SAE International. These levels represent varying degrees of automation, from driver assistance systems to fully autonomous vehicles.

The findings of this research are published in two international conference papers and three peer-reviewed international journals. The developed framework enables a safer, more cost-effective, and efficient development cycle. This is demonstrated through practical case studies that use the framework for developing AVs and AMR control systems. During the research, multiple versions of a logistics robot were built, and a method was developed to modernize the entire AV control system, aiming for autonomous operation. In addition, several safety system components and controllers were developed.

One of the main directions of the study focuses on creating a low-level CPS with high predictive value. This Digital Twin (DT) allows the simulation of the operation of low-level control systems at various levels. Simulations related to the autonomous part of the vehicle can be augmented with low-level CPS simulations, including simulating the basic functionality of Electronic Control Modules (ECUs) along with data interfaces. The traffic generated in these simulated data interfaces is identical to what exists inside the real-world vehicle's data network. This is useful for Hardware in the Loop (HIL) simulations, system testing, development, verification, and validation.

In summary, this study provides a comprehensive overview of the developed scientific methodologies, significantly contributing to ongoing efforts to enhance the practical design, validation, and verification of low-level hardware and software components for AVs. This ensures that the developed framework is compatible and capable of handling the complexity of various AV and AMR platforms, ultimately contributing to the realization of a safe and reliable autonomous transportation system.

Kokkuvõte
Isejuhtivate sõidukite küberfüüsikaline juhtsüsteem

Autonoomsete ehk isejuhtivate sõidukite tehnoloogia edusammud on tekitanud suurt nõudlust ilma juhita transpordi järele lootuses saavutada paremat tõhusust, suuremat liiklusohutust ja energiasäästu. Samas sõiduki juhi puudumine seab olulise väljakutse sõiduki madala taseme küberfüüsikalistele süsteemidele nende keerukuse tõusu tõttu. Kui tavasõidukil saab juht vibratsiooni, heli ja sõiduki poolt kuvatavat visuaalset infot kasutades aimu selle tehnilisest seisukorrast siis isejuhtival sõidukil ei pruugi juhti olla sõidukis või asub juht eemal. See tähendab, et kõik probleemid tuleb tuvastada sensoorika ja erinevatelt süsteemidelt pärineva tagasiside abil.

Käesolev väitekiri uurib autonoomsete sõidukite madala taseme küberfüüsikalisi juhtsüsteeme ja nende arendamise erinevaid aspekte. Juhtsüsteemide arendamise erinevad aspektid võivad olla nende moodulite väljatöötamine, valmis moodulite valideerimine, simuleerimine ja nendele riskianalüüsi tegemine. Need etapid peaks lähtuma standarditest, seadusandlusest ja väljakujunenud tavadest. Esmane eesmärk on luua universaalne küberfüüsikaliste süsteemide arendamise erinevaid meetodeid hõlmav raamistik, mida saab rakendada 4. ja 5. taseme autonoomsete sõidukite, ja robotite arenduses. Need tasemed on määratlenud SAE International poolt ja esindavad erinevat automatiseerituse taset alates juhiabisüsteemidest kuni täielikult autonoomsete sõidukiteni.

Uurimuse tulemused on avaldatud kahes rahvusvahelises konverentsi artiklis ja kolmes eelretsenseeritud rahvusvahelises ajakirjas. Väljatöötatud raamistik võimaldab turvalisemat, kuluefektiivsemat ja tõhusamat arendustsüklit. Seda demonstreeritakse läbi praktiliste juhtumiuuringute, mis hõlmavad raamistiku kasutamist autonoomsete sõidukite ja lisaks autonoomsetel robotite juhtsüsteemide väljatöötamisel. Töö käigus ehitati mitu versiooni logistikarobotist, töötati välja meetod kogu sõiduki juhtsüsteemi moderniseerimiseks eesmärgiga, et see saaks juba uue süsteemi pealt autonoomselt sõita. Lisaks arendati mitmete ohutust tagavate süsteemi komponente ja kontrollereid.

Uurimuse üks põhisuundi keskendub madala taseme juhtsüsteemist kõrge ennustusväärtusega digitaalse kaksiku loomisele. Digitaalse kaksiku kasutamine võimaldab simuleerida madala taseme juhtimissüsteemide tööd mitmetel tasemetel. Sõiduki isejuhtivat osa hõlmavatele simulatsioonidele saab juurde lisada madala taseme küberfüüsikaliste süsteemide simulatsioone, sealhulgas simuleerida elektrooniliste kontrollmoodulite põhilist funktsionaalsust koos andmdesiinidega. Nii genereeritud simuleeritud andmesideliidestes olev liiklus on identne sellega, mis on olemas reaalses sõidukis asuvas andmesidevõrgus. See on kasulik riistvara tsüklis simulatsioonide tegemiseks, süsteemide testimisel, arendamisel, verifitseerimisel ja valideerimisel.

Kokkuvõtteks annab käesolev uuring põhjaliku ülevaate arendatud teaduslikest metoodikatest, mis annavad olulise panuse käimasolevatesse jõupingutustesse täiustada isejuhtivate sõidukite madala taseme riist- ja tarkvarakomponentide praktilist disaini, valideerimist ja verifitseerimist. See tagab, et arendatud raamistik ühildub ja suudab toime tulla erinevate autonoomsete sõidukite ja autonoomsete robotite platvormide keerukusega, andes panuse ohutu ja usaldusväärse autonoomse transpordisüsteemi realiseerimisse.

Appendix 1

I

H. Pikner, R. Sell, K. Karjust, E. Malayjerdi, and T. Velsker, “Cyber-physical control system for autonomous logistic robot,” in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pp. 699–704, IEEE, Apr. 2021

Cyber-physical Control System for Autonomous Logistic Robot

Heiko Pikner
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
heiko.pikner@taltech.ee

Raivo Sell
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
raivo.sell@taltech.ee

Kristo Karjust
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
kristo.karjust@taltech.ee

Ehsan Malayjerdi
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
malayjerdi@gmail.com

Tarmo Velsker
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
tarmo.velsker@taltech.ee

2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC) | 978-1-7281-5660-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/PEMC48073.2021.9432526

***Abstract*—The rapid development of intelligent control technology has affected the logistics industry and led to the implementation of new concepts of autonomous indoor logistic robots. These robots are operating inside warehouses and factories to move goods and packages around in the rooms. There is a high motivation to reduce the cost of these mobile robots as well as to create flexible control systems. Tallinn University of Technology in cooperation with a Kulinaaria production plant has developed a new small scale logistic robot to move boxes in rather tight spaces. New technology, including AI and machine vision, is applied to achieve the flexibility and cost-effectiveness of the newly developed mobile robot solution.**

Keywords— cyber-physical system, logistic robot, low-level control system, transportation, industrial environment

I. Introduction

Indoor smaller-scale mobile robots have become an important part of the modern logistics system in the factories. It is usually a trivial task to carry goods or packages between working stations and warehouses and using the human workforce for this task is not the most effective use of resources. Furthermore, the task itself needs global planning and precise movements in order to ensure a smooth production process. The nature of indoor logistics fits much better for robots than humans. However, it is not trivial to implement robot-based logistics as there are many specific issues that need to be solved and addressed. For example, corridors and transportation space are in most cases designed for humans, not robots and during the transportation task, several fast modifications on the route and navigation must be executed. This is an easy task for humans but not for individual robots. One of the reasons is because the existing robots have predefined fixed paths in their control system and are not flexible to reconfigure if new tasks need to be assigned or unexpected situations occurred on the route.

In TalTech (Tallinn University of Technology) the integrated team consisting of students and researchers from the School of Engineering were started to address this problem. The reason was a real demand from the industry, but also the fact that information and communications technology (ICT) and engineering students have high dropout rates which can be improved when practical projects and industry-related real problems are integrated into the study process. The study [1] conducted in several universities in Estonia showed that integration of interdisciplinary knowledge to attract students from diverse disciplines will improve motivation and reduce the dropout rate.

As a result of an interdisciplinary team brainstorming a decision was made to develop a new robot. The knowledge and experience were gathered from previous self-driving vehicle projects, in particular self-driving platform Iseauto [2, 3] and research of unmanned ground vehicle development methodologies [4, 5]. The work involved the development of a low-level cyber-physical system architecture for an autonomous robot to get a more modular, user-friendly and expandable solution at a lower price point. The robot is designed to carry liquids in a container, powders, empty bottles, packages, cardboard boxes, products packed in boxes, etc. to optimize and automate transport logistics in a factory of the future. This allows components as well as semi-finished products and ready-to-use products to be transported from one manufacturing process to another. One of the first challenges was to redesign the existing Iseauto concept to fit into a smaller space and to be more modular to support 36 V main battery. Multi-master broadcast serial bus communication protocol for connecting controllers in lower-level control systems was applied. The system is compliant with automotive standards [6]. A differential-drive wheel controller and lifting mechanism controller modules were developed. The logistic robot was tested in a factory with a success rate of 80%, field tests confirmed the ability of the robot to pick products without human interference.

The main content of this paper is divided into three parts, the first part is an overview of similar robots, the second part is an overview of the new system architecture of the robot, the third part is an experiment analysis of the tests and the need for future development as to be a part of a new Kulinaaria production plant. As a result of the work, a universal easily expandable logistics robot prototype was developed and tested in an industrial environment.

II. State-of-Art

The warehouse and logistic robotics industry is developing rapidly to obtain price competitiveness through the reduction of logistics costs [7]. The number of small scale industrial logistics robots, similar to the developed solution exists. Robots, listed in Table 1 are selected based on the lifting mechanism which allows them to drive under the frame and lift the payload. These robots have good navigation

capabilities which are crucial in narrow corridors and cases where limited space is available. By comparison, the table shows a TalTech logistics robot, named BoxBot.

TABLE I. LOGISTICS ROBOTS COMPARISON

Developer (Robot name)	Max payload (kg)	Dimensions (mm)	Speed (m/s)	Operation time (hours per charge)
Taltech (BoxBot)	**25**	**length 700, width 340, height 220**	**2**	**12**
Amazon Robotics (Drive Unit)	1000	length 760, width 640, height 410	1.3	N/A
Omron (LD)	130	length 700, width 500, height 383	1.8	15
OTTO Motors (OTTO 100)	100	length 740, width 550, height 301	2	N/A
Fetch Robotics (Freight)	100-1500 (based on model)	length 559, width 508, height 356	2	9
Vecna Robotics (RC20 Conveyor)	20	N/A	N/A	N/A
InVia Robotics (Picker Robot)	18	length 664, width 622, height 650 - 2440	2.2	10
Robotnik Automation (RB-2 BASE)	200	length 990, width 623, height 390	1,7	10
MiR - Mobile Industrial Robots (MiR100)	100	length 890, width 580, height 352	1.5	10
Homag Group (Transbot)	1200	N/A, but visually estimated: length 1200, width 700, height 360	N/A	N/A

The robots listed in the table have quite different payload purposes. For example, Amazon Robotics and Homag Group robots [8] are meant for moving pallets. The load capacity of the robots is over 1000 kg. However, most robots are designed for smaller boxes and payload is limited to 100 kg.

The software, sensors and control system differ greatly. There is a trend to use open-source solutions for different kinds of mobile robot control software. A popular choice nowadays is a Robot Operating System (ROS) with specific software stack or add-on modules. There are few such robots, for example, Robotnik [9], MiR and Freight [10]. Others are using some sort of proprietary system. The high-quality sensors for example 3D lidar (light detection and ranging) are not very common. Most robots use 2D lidars, cameras and ultrasound sensors. The Control system is responsible for guiding the robot, detecting obstacles, selecting paths and lifting. Robots are controlled usually by some smaller computer that may be specifically designed for the particular robot.

III. PROJECT SCOPE

The experiments for logistic robot BoxBot were implemented in cooperation with food and grocery production factory Kulinaaria. Kulinaaria has a classical production floor setup where most of the goods and packages transport between the workstations and warehouse is done by human resources. This is time and resources consuming and also produces a lot of mess and noneffective workflow. In parallel to experimenting with mobile robots, the company is building a completely new production facility. In this new facility, robotic solutions are already taken into account when planning the ground floor.

Experiments with BoxBot in Kulinaaria were carried out in the frame of H2020 project acceleration program for small and medium-sized European manufacturing enterprises and technology suppliers (L4MS) [11]. Special tracks were selected and simulations to find out key performance indicators (KPI-s) were conducted [12]. Based on simulated data, similar routes were experimented with robots to validate simulation results. Mobile robots were replacing humans by carrying raw materials to the production units and transporting empty boxes between washing stations and production units. Detail experiment analysis and KPI calculations are still work in process, but it was clearly seen that applying mobile robots to food production, and in particular Kulinaaria factory set-up, several parameters were improved. For example, the corridors were clean and in good order after robots were applied and human transport workers removed.

IV. TECHNICAL SOLUTION

A. Construction and technical parameters

Before the design process, the initial requirements for autonomous robots were determined and in light of these requirements, a first prototype version was developed. The chassis of a prototype autonomous robot shown in Fig. 1. is made from aluminium to keep the robot's net weight as low as possible and increase the load-bearing capacity. In addition to good strength properties, aluminium also benefits from its good availability, economic feasibility and it is also highly ductile and machinable. A developed autonomous robot has two electrically separately controlled hoverboard motor-driven wheels with diameter 140 mm, width 30 mm and 6 mm thick rubber coating. Both driven wheels are mounted on tension spring adjustable swings. The tension spring-adjustable swings ensure that the robot has enough grip between the rubber-coated wheels and the ground, both under load and unloaded, to efficiently move the robot.

Fig. 1. Universal easily expandable logistics robot prototype “BoxBot”.

The payload for the autonomous robot is carried for better manoeuvrability by four double omni wheels with 125 mm diameter. To ensure that all four load-carrying wheels are in contact with the ground, the front side omni wheels of the robot are mounted on the front axle, swinging perpendicularly according to the robot’s longitudinal axis.

The most important mechanism of the robot is the lifting mechanism, which must be able to lift and carry the prescribed load. As the focus of this project is on robot autonomy, then the first prototype version had a requirement for a maximum payload of 30 kg. There are different possibilities that can be used for the purpose of lifting – driving pulley and belt, lifting jack, lead screw, ball screw, spiral lift, etc. Taking into consideration all the advantages and disadvantages of different available mechanisms for lifting and all the requirements for the robot project, the electric linear actuator was decided to use. The design of a lifting mechanism allows for maximum lift height 30 mm, which is enough to safely lift and carry a payload in the warehouse.

Aluminium made four-pronged lifting leg is connected with the nut of the lead screw of the linear actuator, shown in Fig. 2. Four cylindrical guide bearings made from POM are mounted on one plane to the robot chassis and the cylindrical rods inside the guide bearings rest on the edges of the four-pronged lifting leg. If the led screw is rotating the four-pronged lifting leg connected with a nut is shifting cylindrical rods by the openings inside the bearing guides. The lifting mechanism is stopped by the limit switches. The payload is carried from cylindrical rods by the four-pronged lifting leg to the lead screw. The upper part of the lead screw is fixed with the chassis to guarantee the stability of the hoisting drive. The analyses for the autonomous robot chassis, tension spring adjustable swings and four-pronged lifting leg was performed in the *SolidWorks Simulation* environment.

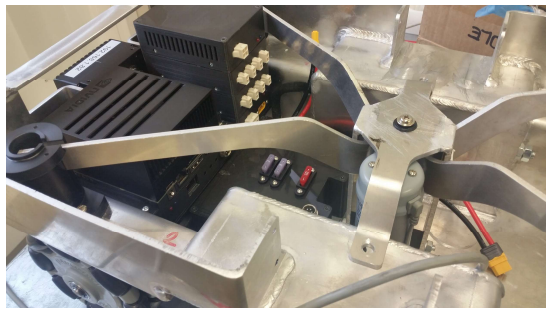

Fig. 2. The lifting mechanism.

B. Control architecture

The control architecture of the logistic robot is similar to Iseauto and is divided into three layers as described in Fig. 3. The upper layer provides input to the ROS high-level control system. All four solid-state lidars are connected to the central network switch. The USB interface is used for the ZED camera and Xsens inertial sensor module. AI & drive algorithm layer is based on the NVIDIA Jetson AGX Xavier developer kit, suitable for creating and deploying end-to-end AI robotics applications for manufacturing, delivery, retail, agriculture, and more. The first and second layers are described in more detail in the next section.

The logistic robot commands are sent to the low-level control layer that has a mission-critical functionality to take care of the robot's movement control. The central unit for this

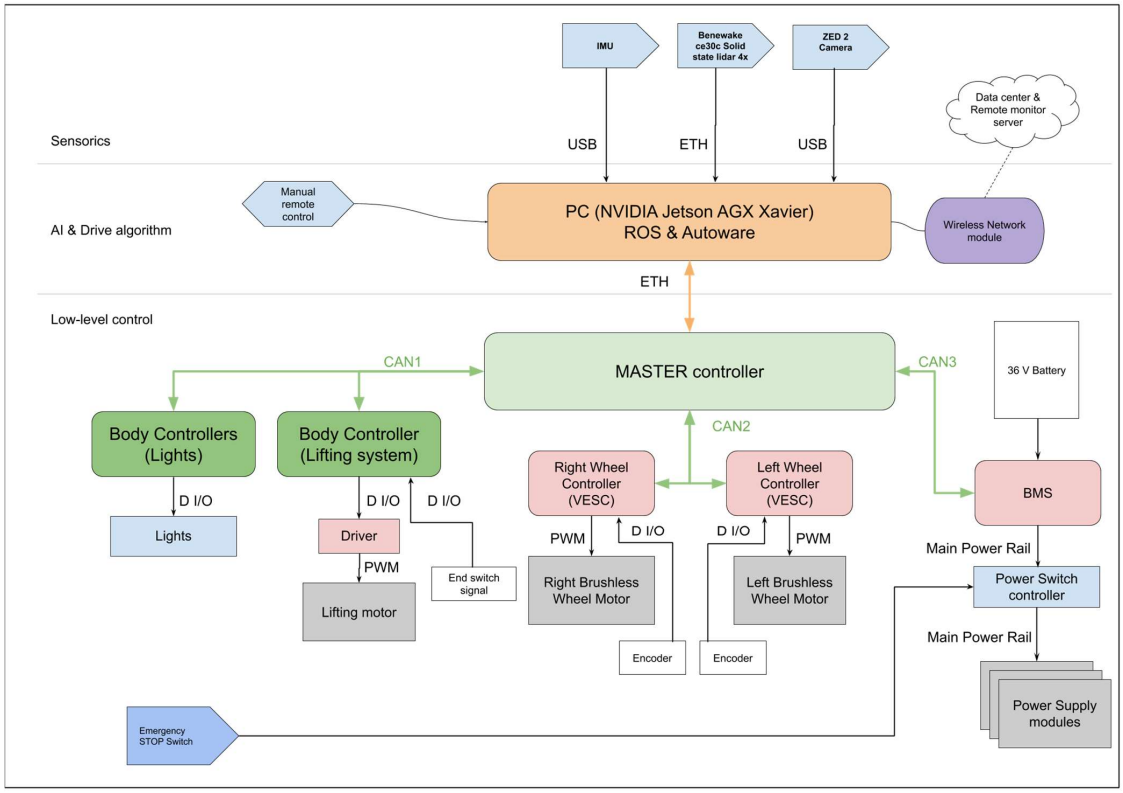

Fig. 3. The low and high-level architecture of the logistic robot control system.

layer is the master controller, initially developed for Iseauto. The main task of the master controller is to forward information from and to the main computer, the NVIDIA Xavier. It handles communication with ROS and communicates with the other controllers over different CAN networks. The CAN1 network is dedicated to the lifting mechanism and other body controllers. The CAN2 network is for the propulsion system. The number of slave controllers may vary depending on the progress of the development. The controller's hardware is based on the STM32 ARM.

The propulsion system consists of two Mini FSESC4.20 open source brushless motor controllers. These motor controllers are capable of driving current over 50 A and are widely configurable. For the communication with motor controllers, inside the master controller firmware, a CAN2 network is configured differently and has a specially developed driver to support VESC custom protocol. A CAN bus frame can contain a maximum of 8 bytes of data. Each VESC has a unique CONTROLLER_ID given from the configuration interface. CAN messages based on an extended ID (EID) 29-bit identifier. These identifiers contain COMMAND and CONTROLLER_ID data fields [13].

Two electric wheel hub motors are connected to the VESC motor controllers. The power consumption of each 300 W motor is 8 A. Compared to the max current 50 A, there is a quite large margin to use more powerful motors when needed. The motors have built-in hall sensors. Hall sensors are used by the motor controller for the PID regulator. In addition, speed information alongside other information will be sent over the CAN bus to the master controller.

Lifting mechanism has its own controller which is connected to CAN1. The position information of the mechanism is obtained from the limit switches and status sent to the CAN bus. This allows the computer to decide if the cargo is suddenly too heavy. According to the command, the lifting mechanism is moved up or down. Inside the lifting mechanism is a body controller, motor controller, interface board and power supply to lower the voltage to be suitable for a 12 V DC motor.

The logistic robot power system is based on a 36 V rechargeable Li-ion battery. The battery was chosen which has a nominal capacity of 25 Ah. That's enough to power the robot over 12 hours, which is comparable to other similar robots. Power switching goes over the solid-state switch. Fuses, a solid-state switch and both VESC motor controllers are assembled into a compact driving module.

In addition, modular power supply was designed as shown in Fig. 4. for the robot to get different voltages for a computer (19 V), a network switch (9 V), a master controller (12 V) and solid state lidars (12 V). The power supply module provides a simple structure to add or remove sections to get new power rails.

C. High-level software architecture

The high-level software solution is based on ROS and modular architecture proposed by [14]. The reasons behind this decision were open-source drivers for multiple sensors and ease of integration of third-party software like Autoware [15] and multiple device drivers. The main sensor is 3D lidar used for localization, obstacle detection, safety functions and path following. Control outputs are twist command, brake and linear velocity which are sent to the low-level controllers over UDP messages. The main decision-making software for current architecture is ROS/Autoware which is an open-source library for self-driving cars and thus has many advanced software capabilities like lane following, obstacle avoidance, lane detection, etc.

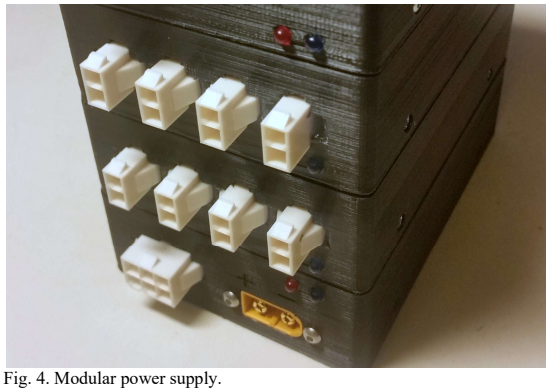

Fig. 4. Modular power supply.

The ROS platform itself is based on the 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 the multiple platforms. The ROS is a middleware and operates well on multiple cross platforms. To merge ROS and lower-level controllers, a software bridge was built which converts custom ROS messages to UDP messages. The modularity of this particular high-level software architecture is mainly a result of key principles of ROS and its approach to implementing multiple software libraries as building blocks of the primary product.

The mapping, localization, and navigation on this robot are based on the Normal Distributions Transforms algorithm. For creating a 3D map, four solid-state lidars (Beneware Ce30-C) are used in the robot. After applying the sensor fusion four solid-state lidars are converted to a 360-degree lidar output (Fig. 5).

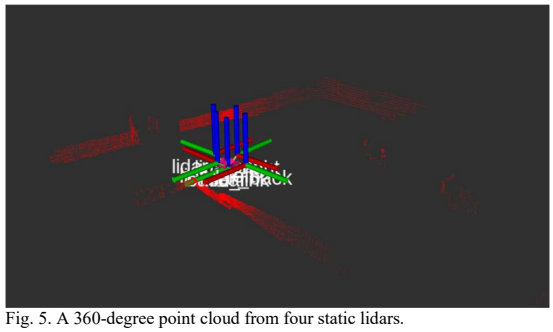

Fig. 5. A 360-degree point cloud from four static lidars.

This point-cloud is used for creating a 3D map (Fig. 6), which is used for localization and navigation. For navigation, the pure pursuit algorithm is used. Pure pursuit is a path following algorithm. It calculates the angular velocity command that moves the robot from its current position to

reach some look-ahead point in front of the robot shown in Fig. 7.

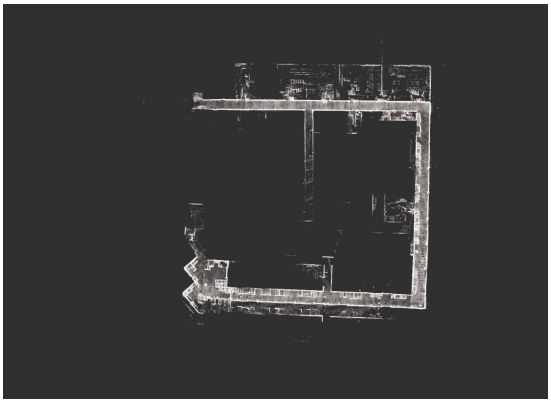

Fig. 6. The 3D point cloud map.

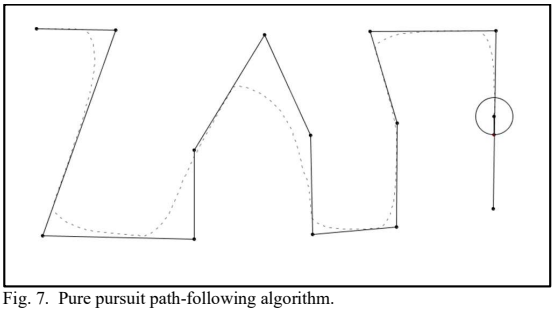

Fig. 7. Pure pursuit path-following algorithm.

The linear velocity is assumed constant, hence you can change the linear velocity of the robot at any point. The algorithm then moves the look-ahead point on the path based on the current position of the robot until the last point of the path.

V. Results

The developed logistic robot experimented in the Kulinaaria factory in Tallinn. Different sections where the transportation need was highest were identified and selected, see Fig.8. All paths were initially simulated and then validated with the real robot.

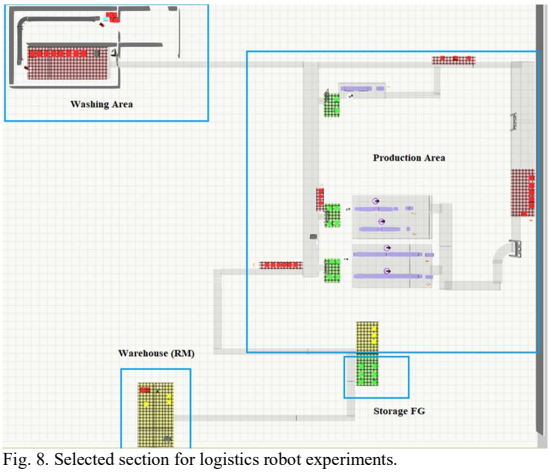

Fig. 8. Selected section for logistics robot experiments.

Before the experiments, all selected areas were mapped with 3D lidar. The created map for one path is shown in Fig. 9.

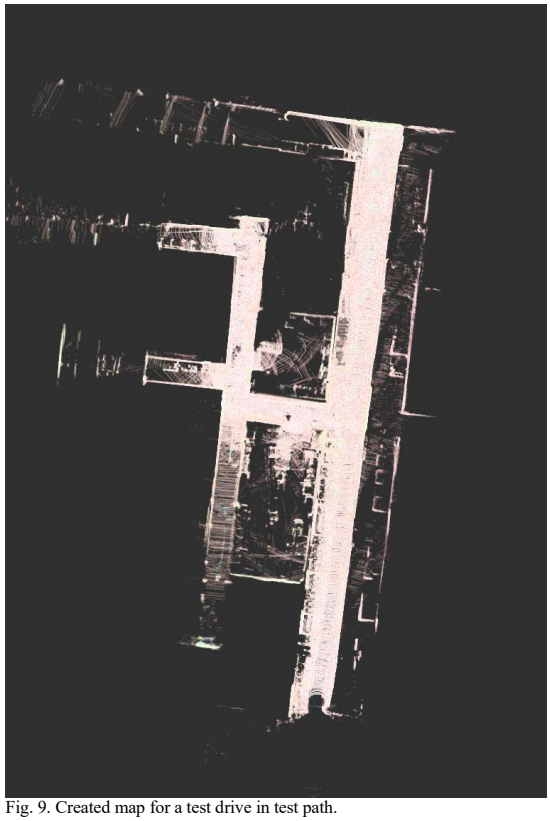

Fig. 9. Created map for a test drive in test path.

After creating a waypoint, the logistic robot moved autonomously on this path. As a result of the first experiments, several issues were identified. For example, there were too many spare boxes and random objects in the corridor. As these objects are not static (they were constantly relocated) it generated errors between the pre-recorded and real-time maps resulting in the loss of localization of the robot. As a conclusion of this issue, it is very complicated to implement human-robot co-existence in the indoor transportation solution. Humans tend to work randomly and have no big trouble rearranging their transport path when the environment is changed while robots expect much higher order in their path and are more precise and predictable. Findings out of this experiment helped to design a better environment for the future factory of Kulinaaria - to be more robot friendly and optimized effective indoor logistics with the help of robots.

VI. Future Work

Several improvements and updates are scheduled for the next prototype. An updated machine vision algorithm for object detection in the factory area will be implemented to get better results of other objects around. The navigation algorithm needs to be optimized. Although the pure pursuit algorithm is generally a good solution it has some problems in narrow corridors, so there is a need for a better path-following algorithm for precise navigation. The maximum detection range on the Ce30-c lidar is 4 meters, again in narrow

corridors the beam is too narrow and long-range lidar works better as it can localize the robot not from walls but ceiling corners. From a mechanical point of view, the aim of the next version is to increase the lifting capacity of the lifting mechanism and make it more rigid than it is on the current version. It is also necessary to deal with the smoother start and braking of the robot movement to move taller stacked plastic containers.

VII. Conclusion

The paper discusses mobile robots in indoor industrial logistics and is focusing on the new development of flexible, open-source based robot BoxBot. A cost-effective robot was built based on competence and know-how about autonomous vehicles which was obtained from the developing previous self-driving platform Iseauto. The logistic robot was tested in a food and grocery production factory Kulinaaria. Field tests confirmed the ability of the robot to transport packages without human interference. Experiments brought up weaknesses and issues which need to be addressed in the next version. Some of them are specific to this particular factory set-up but some of them are more general and are related to the changing environment due to the fact that humans and robots are working together. If the placement of goods in storage areas changes too much, then maps will be too different and the robot lost its localization. This will be taken into account when developing the next improved version of a logistic robot. A better vision algorithm for object detection in the factory area with odometry data should solve the problem.

Acknowledgment

The project is supported by the H2020 project L4MS (VE19030) and the INTERREG project INFORM (VIR19004).

References

[1] Siiman LA, Pedaste M, Tõnisson E, et al. A review of interventions to recruit and retain ICT students. *International Journal of Modern Education and Computer Science* 2014; 6: 45.

[2] Sell R, Leier M, Rassolkin A, et al. Self-driving car ISEAUTO for research and education. In: *Proceedings of the 2018 19th International Conference on Research and Education in Mechatronics*

[3] Rassõlkin A, Sell R, Leier M. Development case study of the first estonian self-driving car, iseauto. *Electrical, Control and Communication Engineering* 2018; 14: 81–88.

[4] Sell R, Coatanéa E, Christophe F. Important Aspects of Early Design in Mechatronic. In: Küttner R (ed) *Proceedings of the 6th international conference of DAAAM Baltic industrial engineering*. Tallinn University of Technology, (2008).

[5] Väljaots E, Sell R. Unmanned ground vehicle SysML navigation model conducted by energy efficiency. In: *Advanced Materials Research*. Trans Tech Publ, 2014, pp. 443–447.

[6] Johansson KH, Törngren M, Nielsen L. Vehicle Applications of Controller Area Network. In: Hristu-Varsakelis D, Levine WS (eds) *Handbook of Networked and Embedded Control Systems*. Boston, MA: Birkhäuser Boston, 2005, pp. 741–765.

[7] Keow MAKS, Nee AYH. Robotics in Supply Chain. *Emerging Technologies for Supply Chain Management* 2018; 25.

[8] HOMAG. Automated guided vehicle system TRANSBOT | HOMAG. *Homag Group AG*, https://www.homag.com/en/product-detail/automated-guided-vehicle-system-transbot (2019, accessed 10 March 2020).

[9] Guzmán R, Navarro R, Cantero M, et al. Robotnik—Professional Service Robotics Applications with ROS (2). *Studies in Computational Intelligence* 2017; 419–447.

[10] Estolatan E, Geuna A, Guerzoni M, et al. Mapping the Evolution of the Robotics Industry: A cross country comparison. *White Paper Series*, https://munkschool.utoronto.ca/ipl/files/2018/07/robots-final-Jul11.pdf (2018).

[11] L4MS project Home | L4MS, http://www.l4ms.eu/ (accessed 10 March 2020).

[12] Mahmood K, Karaulova T, Otto T, et al. Development of cyber-physical production systems based on modelling technologies. *Proc Eston Acad Sci*; 68, (2019).

[13] VESC Canbus Communication — Triforce 0.1.0 documentation, https://triforce-docs.readthedocs.io/en/latest/canbus/canbus.html (accessed 10 March 2020).

[14] Sell R, Väljaots E, Pataraia T, et al. Modular smart control system architecture for the mobile robot platform. *Proc Eston Acad Sci* 2019; 68: 395–400.

[15] Kato S, Tokunaga S, Maruyama Y, et al. Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems. In: *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. 2018, pp. 287–296.

Appendix 2

II

H. Pikner, R. Sell, J. Majak, and K. Karjust, “Safety system assessment case study of automated vehicle shuttle,” *Electronics*, vol. 11, p. 1162, Apr. 2022

electronics

MDPI

Article

Safety System Assessment Case Study of Automated Vehicle Shuttle

Heiko Pikner [1,2,*], Raivo Sell [1,2], Jüri Majak [1,2] and Kristo Karjust [1]

[1] Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia; raivo.sell@taltech.ee (R.S.); juri.majak@taltech.ee (J.M.); kristo.karjust@taltech.ee (K.K.)
[2] FinEst Centre for Smart Cities, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia
* Correspondence: heiko.pikner@taltech.ee

Abstract: Automated vehicle (AV) minibuses, i.e., AV shuttles, are gaining popularity in the testing of new types of transportation services in real traffic conditions. AV shuttles have moved from closed test areas to low-traffic public sites such as local residential areas, technology parks, university campuses, etc. These types of vehicles are usually low-speed and rely on a lidar-camera sensor set and a self-driving software stack. These new use cases are increasing these systems' safety demands. In addition to functional safety, many other aspects need to be considered. In this study, a risk analysis model is developed, combining the fuzzy analytical hierarchy process and the Technique for Order of Preference by Similarity to Ideal Solution method. The proposed model is utilized to prioritize risks corresponding to the particular case study, based on real AV shuttle bus development, and focuses on the low-level hardware/software safety issues and improvements.

Keywords: safety architecture of the AV shuttle; automotive electronics key standards; risk evaluation model development; automotive communication networks

Citation: Pikner, H.; Sell, R.; Majak, J.; Karjust, K. Safety System Assessment Case Study of Automated Vehicle Shuttle. *Electronics* **2022**, *11*, 1162. https://doi.org/10.3390/electronics11071162

Academic Editor: Shinichi Yamagiwa

Received: 21 February 2022
Accepted: 29 March 2022
Published: 6 April 2022

1. Introduction

Automated driving technology development is under active investigation in many different industrial sectors, such as the automotive industry, mining, machinery, etc. The automotive industry is constantly developing new autonomous driving aid system features and functionalities. The general target is to reach fully autonomous driving by the end of this decade. Many car manufacturers, such as Tesla, Ford, etc., have declared in recent years that they will reach fully autonomous driving cars very soon [1] but have had to postpone their announced deadlines many times [2]. At the same time, several IT giants are trying to develop autonomous driving, with Waymo from Google and Apple's self-driving car project being the most well-known, but the challenges involved have been higher than initially predicted, and because of this their deadlines have been prolonged. Companies in the manufacturing industry and warehouse logistics have tested and applied automated mobile robots to make industrial processes more efficient and flexible. The Industry 4.0 and 5.0 philosophies rely heavily on connected and automated systems with seamless connectivity. Several studies have focused on the integration of AV shuttles into industrial processes as part of the Industry 4.0 concept [3]. All these efforts related to automated driving and vehicle developments face rather similar challenges. Functional safety and cybersecurity are often the main concerns when implementing and deploying automated vehicles.

Automated vehicle (AV) shuttles are a new type of transportation, targeted at solving the last-mile public transport gap. AV shuttles are mostly low-speed 6–12 seat minibuses with SAE level 4 [4] autonomy. This means that the vehicles are fully automated, without having any on-board human control devices, but are operating in a defined operational domain. The operational design domain (ODD) sets the limits in which the conditions of the vehicle are designed to operate, in terms of geographical area, weather and road

conditions, speeds and traffic density, etc. Safety is the main concern and is kept in mind as the number one priority throughout the whole development process, starting from the design and development stage and ending with the deployment and services stage.

In this study, a safety assessment case study is carried out based on the AV shuttle prototype designed and developed at TalTech by the autonomous vehicles research group in cooperation with industrial partners [5,6]. The shuttle was designed modularly, and safety issues were addressed in many layers. In fact, one of the industrial partners, ABB, was responsible for designing a low-level safety system to ensure safe vehicle operation and signal-level monitoring of anomalies. Safety was included at the very beginning of the design process, and was supported by the early design methodology for the mechatronic system, proposed in the earlier collaborative work of the Aalto and TalTech research groups [7,8].

Industry 4.0 requires high levels of digitalization in order to process all the information that is generated in virtual representations or cyber versions of the physical world. The modular cyber-physical system (CPS) is a critical part of the integration between these two worlds. Modules interacting with the physical world can be divided mainly into sensors, actuators, and computational units [9]. Mobile modular CPS is typically designed as a network to create some global behavior [10], and it has significant computational resources to maintain localization, obstacle detection, safety functions, and path following. Computational resources can be divided into two categories: artificial intelligence (AI) based on high-level decision-making and lower-level control logic. AI and high-level decision-making are based on the use of special computers to run robotic operating systems (ROS). The low-level control logic is implemented near or inside the actuator or sensor modules. It handles the regulation of actuators and performs the first information processing of information received from sensors. It also controls and forwards information between the modules.

Despite intensive developments in autonomous driving, fully automated driving systems (without human supervision) are not yet allowed onto public streets together with urban traffic [11]. Safety is a key concern of any fully or partially autonomous driving system, due to the need to consider/understand several complex factors such as the environment, traffic, hardware and software systems' reliability, information availability, cyber security, etc. For example, twelve principles have been identified by authors from different car manufacturers, which highlight the safety and security-relevant aspects [12].

The problem considered includes multiple criteria and a number of impact factors. In engineering design, evolutionary optimization techniques are most commonly utilized for handing mixed-integer variables and to provide convergence to a global optimum [13–17]. To reduce computing time, artificial-intelligence-based meta modeling techniques have been implemented (ANN) for the modeling of objective and constraint functions [14,15,17]. Another approach for simplifying complex engineering design problems is to decompose the initial optimization problem into simpler subproblems. In [18], a nondestructive testing method was presented for determining the elastic constants of orthotropic composites using Lamb wave propagation measurements in plates and fitting the dispersion curves by means of a simple genetic algorithm. The results obtained in [18] were extended in [19], in which the micro genetic algorithm (μGA) and two-stage Nelder–Mead simplex optimization procedure were developed. It was shown in [19] that the two-stage algorithm outperforms GA and μGA by reduced computing time. In [20], GA and a modified two-stage simplex optimization algorithm were employed to solve laminate stiffness parameter identification inverse problems. The two-stage simplex optimization algorithm appears to be less time consuming. In [21], multicriteria parametric optimization of composite sandwich plywood plates with skin layers of birch plywood and a core of straight and waved plywood cell-type ribs was performed to reduce the computing time of the the response modeling, as applied to both objective and constraint functions. The optimal design of the load-bearing capacity of high-performance concrete columns subjected to compression and flexure loads was studied in [22]. It was observed that the use of high-performance steel fiber concrete

as a column material was especially effective for columns, with additional longitudinal reinforcement, and the load-bearing capacity was up to 15%.

However, the problem considered in the current study has some specific features. The evaluations (judgments) provided by decision makers include uncertainty. The evolutionary multicriteria optimization methods described in the previous section have been applied with success in solving a wide class of engineering design problems [13–22]. However, despite their stochastic nature, these evolutionary algorithms are not well suited for handling judgements involving uncertainty. For this reason, in the following, multicriteria decision-making (MCDM) methods are utilized.

Firstly, for the prioritization of the criteria, the fuzzy analytic hierarchy process (FAHP) is applied. The Fuzzy AHP was introduced as a combination of fuzzy sets and AHP [23]. The FAHP has an obvious advantage over AHP; it simplifies decision makers' evaluations by replacing fixed-value judgments with interval judgments.

Secondly, for the prioritization of risks, the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is applied. According to TOPSIS, the most preferred alternatives should have the shortest distance from the positive ideal solution (PIS) and the farthest distance from the negative ideal solution (NIS) [24]. The TOPSIS method has found wide use in transportation and intelligent vehicle systems [11,25,26]. In [26], a hybrid approach was employed, combining the TOPSIS and AHP methods.

Other popular MCDM methods include Elimination and Choice Translating Reality (ELECTRE), Vlsekriterijumska optimizacija IKompromisno Resenje (VIKOR), Preference-Ranking Organization Method for Enrichment Evaluations (PROMETHEE), the weighted sum model (WSM) and weighted product model, etc. The ELECTRE method is used to develop a solution based on an outranking relationship between two alternatives [27]. The implementation of the ELECTRE algorithms is estimated to be rather complex. The VIKOR method determines the optimal solution based on estimating the closeness of alternatives to an ideal alternative [27]. This method may become challenging in the case of conflicting scenarios. The PROMETHEE method belongs to the class of outranking methods and it is based on the comparison of the amplitude of the deviations between the evaluations of the alternatives within each criterion [27]. In the case of this method, an extra tool is needed for the evaluation of the weights of the criteria. According to the weighted sum model (WSM) the optimal solution is determined as the one with the best value of the weighted sum. In the case of the weighted product model (WPM) the summation is replaced by multiplication [27].

The reasons for the selection of the TOPSIS method the current study can be outlined as follows.

- TOPSIS is simple to implement;
- TOPSIS provide robust solutions, it tends to provide a positive ideal solution, but avoid a negative ideal solution; and
- TOPSIS has been utilized with success in the study of intelligent vehicle systems.

In the following, the fuzzy AHP and TOPSIS approaches are combined for the prioritization of the criteria and risks, respectively. The proposed fuzzy sets-based approach allows us to apply linguistic assessments corresponding to the natural representation of the judgment [23,24].

This paper focuses on providing a practical approach to the implementation of a cyber-physical system on autonomous vehicles, focusing on the AV shuttle in particular. The safety issues are studied in the context of considered problems. The risks and their evaluation criteria are developed for a particular class of problems.

2. Background of Key Automotive Standards

Technological innovations and progress in the automotive industry, especially with the introduction of driver-assist and automated driving systems, have brought about a need for standards that define functional safety and functions that contribute to the prevention of accidents in emergency situations. Functional safety is a method of reducing risks to

an acceptable level to ensure safety by devising functions. Among many other standards, not limited to the automotive field, ISO 26262 is a functional de facto safety standard for electrical and electronic systems in road vehicles, based on IEC 61508. ISO 26262—A, B, C, and D define ASIL as a risk classification system. A represents the lowest degree, and D represents the highest degree of automotive hazard. It is mainly used as a basis to perform hazard analysis and risk assessment for vehicle electronic control units (ECUs). It is possible to measure severity, exposure, and controllability and provide classifications. Each classification is broken down into sub-classes. These classifications and sub-classes are analyzed and combined to determine the required ASIL [28,29].

Manufacturers must meet a list of specific industry standards throughout the component manufacturing and testing process in order for the automotive to qualify. The IATF 16949/ISO 9001 international standard defines the requirements for a quality management system for organizations in the automotive industry, including automotive production, service, and accessory parts organizations [30].

The durability standards of automotive electronic components are defined by the component type. AEC-Q100 is a failure mechanism-based stress test qualification for packaged integrated circuits. An AEC-Q100-qualified device means that the device has passed the specified stress tests and guarantees a certain level of quality/reliability [31]. AEC-Q200 is a global stress resistance standard set for all passive electronic components. Five temperature ranges are defined. Parts are deemed to be AEC-Q200-qualified if they have passed the stringent suite of stress tests [32]. SAE USCAR2 is a standard that covers the performance testing of road vehicle electrical terminals and connectors [33].

3. Risk Evaluation Model Development

Safety is one of the most critical issues in the development of mobile robots and self-driving vehicles, since a high price can be paid for shortcomings in this area, depending on the safety topics involved. The risk analysis presented here provides an overview of the current situation and forms a basis for safety improvements in future solutions. The proposed risk evaluation model includes three main modules:

- Formulation of criteria and risks [34];
- Prioritization of criteria (fuzzy AHP);
- Prioritization of risks (fuzzy TOPSIS).

The first module covers the formulation of the criteria and risks for considered mobile robot types. It was introduced by authors in [34] and is described as follows.

Mission computer and AI performance (C1): This criterion refers to the reliability of the mission computer and AI system. Situations in which the AV vehicle is unable to perform the tasks assigned to it may lead to the cessation of production or interruption of the transportation of passengers and goods.

Cybersecurity (C2): This criterion refers to all sorts of hacking of automated systems. Remote-control attacks are one of the prioritized security threats. Autonomous passenger transport carries the risk of the passenger gaining access to the vehicle's internal network or computer viruses finding their way into the system.

Malfunction of AV mechanical component (C3): The mechanical components of an autonomous vehicle may fail, which creates the risk of accidents and further damage.

The sensor system (C4): This criterion refers to the reliability of the sensors. The sensors may stop working due to mechanical breakdown or electrical failure. The operation of the sensors can maliciously interfere with lasers, radio jammers, and other devices.

The communication link (C5): This criterion refers to the reliability of the communication links. The components of the communication link may fail due to hardware or software issues and hacking. A loss of communication may lead to accidents.

Weather factors (C6): This criterion refers to the driving environment factors, including weather conditions and other factors that are essential for prioritizing the risk in a driverless vehicle.

Low-level cyber-physical system performance (C7): This criterion refers to low-level cyber-physical system performance and failure, which also creates the risk of accidents and further damage.

Mechanical failure risk (A1): This risk category refers to the failure of the mechanical components due to normal wear and tear, manufacturing or design errors, corrosion, vandalism, mishandling, or an accident.

Electrical failure (A2): This risk category refers to the failure of the electrical components. Electrical components can be divided roughly into ECUs, wiring harness, batteries, sensors, and mechanical actuators. Failure may occur due to manufacturing or design errors, corrosion, short circuit, overheating, software failure, or hacking. Mechanical damage is also possible. These types of faults can lead to greater damage, such as fire or accident.

Information shortage (A3): This risk category refers to the failure relating to the loss of communication. As the vehicle or robot should operate autonomously, this type of error does not directly cause major damage. However, if an attempt is made to stop or drive the vehicle due to a previous malfunction, an information shortage may result in an accident.

Autonomous driving software failure (A4): This risk category refers to the failure of autonomous driving software. This is one of the most prioritized security threats, which could lead to an accident. This type of failure is difficult to detect and correct from the lower side and requires urgent intervention by the remote-control center.

Low-level software failure (A5): This risk category refers to a low-level software failure, mainly due to programming or design errors. This risk is controllable by making the right design choices in the cyber-physical architecture. However, the occurrence of these failures is dangerous, as the actuators can move unpredictably, and the vehicle may undergo high acceleration, causing a crash. The actuators and the electrical system may be damaged due to overload or due to signals occurs in the wrong order.

Communication bandwidth shortage (A6): As the vehicle should operate autonomously, this type of error does not directly cause major damage. However, if an attempt is made to stop or drive the vehicle due to a previous malfunction, a communication bandwidth shortage may result in an accident. This risk category refers to the fact that the remote-control center may lose access to the vehicle overview information and the remote-control option.

Cyber-hacking (A7): This risk category is involved with the deliberate exploitation of automated vehicle systems by unauthorized entities. The target of the attack can vary, ranging from an attack on software to managing the system. Remote-control attacks are one of the highly prioritized security threats, and could be considered the most dangerous type of attack.

Interruption of uplink (A8): As the vehicle should operate autonomously, this type of error does not directly cause major damage, but the remote-control center may lose access to the vehicle overview information and the remote-control option.

A drastic change of environment (A9): A drastic change in the environment may pose a risk. For example, snow may accumulate on the sensor's surfaces, and heavy rain or snowing may disturb the operation of the sensors. An inside environment may contain dust, food, and other substances which may cover sensors or block mechanical actuators. An accident may occur if dire circumstances coincide. A significant drop in temperature may cause an electrical system failure.

Loss of localization (A10): In this case, the vehicle does not know where it is located. An accident may occur if the vehicle tries to move. With appropriate design choices for autonomous driving software, this risk should be minimized. In addition, if the vehicle is unable to restore its localization, the remote-control center should take control.

Based on the above-defined criteria and risks, a decision hierarchy tree for the considered mobile autonomous systems can be established, as shown in Figure 1.

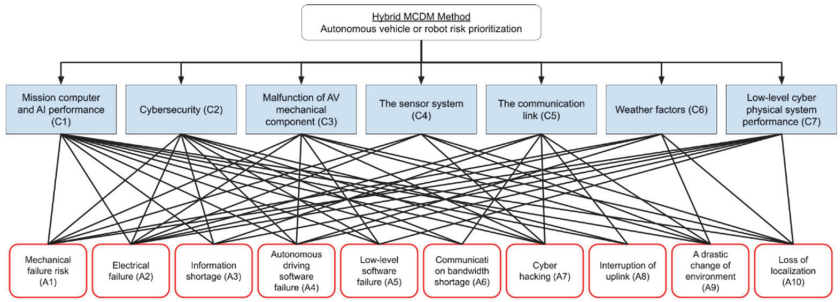

Figure 1. Criteria and risks decision hierarchy.

In the following section, the last two modules of the risk evaluation model are described.

3.1. Criteria Prioritization Using Fuzzy AHP

In the following, the fuzzy AHP approach, based on triangular fuzzy numbers (TFN), is applied to prioritize the criteria introduced above.

Step 1. The criteria were evaluated in terms of linguistic variables. First, the linguistic variables were introduced, as shown in Table 1, to simplify the evaluation process of the importance of criteria [35].

Table 1. Linguistic variables for the importance of the criteria (based on [35]).

The Relative Importance in Terms of Linguistic Variables	Crisp AHP Scale	Fuzzy Triangular	Reciprocal Fuzzy
Equally Preferred (EqP)	1	1, 1, 1	1, 1, 1
Equally to Moderately Preferred (Eq-MP)	2	1, 2, 3	1/3, 1/2, 1
Moderately Preferred (MP)	3	2, 3, 4	1/4, 1/3,1/2
Moderately to Strongly Preferred (M-SP)	4	3, 4, 5	1/5, 1/4, 1/3
Strongly Preferred (SP)	5	4, 5, 6	1/6, 1/5, 1/4
Strongly to Very Strongly Preferred (S-VSP)	6	5, 6, 7	1/7, 1/6, 1/5
Very Strongly Preferred (VSP)	7	6, 7, 8	1/8, 1/7, 1/6
Very Strongly to Extremely Preferred (VS-ExP)	8	7, 8, 9	1/9, 1/8, 1/7
Extremely Preferred (ExP)	9	8, 9, 9	1/9, 1/9, 1/8

Next, the expert group of decision-makers filled the pairwise comparison matrix criteria vs. criteria in terms of linguistic variables. Table 2 presents the linguistic "grades" given by one expert as an example.

Table 2. Pairwise comparison matrix of main criteria.

	C1	C2	C3	C4	C5	C6	C7
Mission (C1)	EqP						
Cybersecurity (C2)	Eq-MP	EqP					
Malfunction of AV mech. component (C3)	EqP	Eq-MP	EqP				
Sensor system (C4)	S-VSP	MP	EqP	EqP			
Communication link Reliability (C5)	1/MP	1/MP	1/M-SP	1/MP	EqP		
Weather factors (C6)	EqP	1/MP	1/SP	1/M-SP	MP	EqP	
Low-level cyber-physical system (C7)	EqP	MP	EqP-MP	EqP	S-VSP	SP	EqP

Step 2. The linguistic scales were transferred to triangular fuzzy numbers (TFN) based on Table 1. These individual tables are omitted herein for the sake of brevity.

Step 3. The aggregated evaluation matrix, presented in Table 3, was computed by applying a fuzzy geometric mean

$$r_{ij} = \left(\prod_{n=1}^{N} c_{ijn}\right)^{1/N}. \quad (1)$$

In Equation (1), c_{ijn} stands for the fuzzy comparison value in terms of the TFN of criteria i to criteria j given by the n-th expert and N is the total number of decision-makers involved. The computed values of the pairwise comparison matrix r_{ij} are given in Table 3. Here $r_{ij} = (l_{ij}, m_{ij}, u_{ij})$ are triangular Fuzzy numbers, where l, m, and u stand for lower, medium, and upper values, respectively.

Table 3. Aggregated pairwise comparison matrix.

	C1	C2	C3	C4	C5	C6	C7
C1	(1.00; 1.00; 1.00)	(0.34; 0.43; 0.60)	(0.33; 0.38; 0.47)	(0.15; 0.18; 0.23)	(1.20; 1.77; 2.33)	(1.00; 1.00; 1.00)	(0.46; 0.53; 0.63)
C2	(1.67; 2.33; 2.94)	(1.00; 1.00; 1.00)	(0.44; 0.54; 0.73)	(0.37; 0.45; 0.59)	(1.78; 2.47; 3.24)	(2.14; 2.61; 3.03)	(0.35; 0.44; 0.63)
C3	(2.14; 2.61; 3.03)	(1.36; 1.85; 2.29)	(1.00; 1.00; 1.00)	(0.31; 0.35; 0.42)	(1.35; 1.70; 2.12)	(2.00; 2.53; 3.24)	(0.34; 0.47; 0.71)
C4	(4.44; 5.52; 6.46)	(1.70; 2.24; 2.70)	(2.40; 2.85; 3.20)	(1.00; 1.00; 1.00)	(1.76; 2.22; 2.74)	(2.29; 2.74; 3.14)	(0.93; 1.07; 1.26)
C5	(0.43; 0.56; 0.83)	(0.31; 0.41; 0.56)	(0.47; 0.59; 0.74)	(0.37; 0.45; 0.57)	(1.00; 1.00; 1.00)	(0.37; 0.45; 0.59)	(0.37; 0.40; 0.43)
C6	(1.00; 1.00; 1.00)	(0.33; 0.38; 0.47)	(0.31; 0.40; 0.50)	(0.32; 0.37; 0.44)	(1.70; 2.24; 2.70)	(1.00; 1.00; 1.00)	(0.30; 0.34; 0.40)
C7	(1.59; 1.89; 2.18)	(1.59; 2.25; 2.85)	(1.40; 2.14; 2.93)	(0.79; 0.93; 1.07)	(2.31; 2.51; 2.71)	(2.49; 2.93; 3.32)	(1.00; 1.00; 1.00)

Step 4. Next, the aggregation was applied with respect to each row of the aggregated comparison matrix given in Table 3. As a result, the fuzzy comparison values $r_i = (l_i, m_i, u_i)$ can be evaluated as:

$$r_i = \left(\prod_{j=1}^{Ncrit} r_{ij}\right)^{1/Ncrit}. \quad (2)$$

In Equation (2) $Ncrit$ stands for the number of criteria used.

Step 5. The triangular fuzzy weight w_i of criteria i is determined as the normalized value of the r_i.

$$w_i = (l_i, m_i, u_i) = r_i \otimes (r_1 \oplus r_2 \oplus \ldots \oplus r_{Ncrit})^{-1}, \ldots i = 1, \ldots, Ncrit. \quad (3)$$

Step 6. Finally, the crisp weights can be obtained by applying defuzzification for fuzzy weights as (different approaches for defuzzification can be found in [36]).

$$w_i^{Crisp} = l_i + [(u_i - l_i) + (m_i - l_i)]/3. \quad (4)$$

In Table 4 are presented the fuzzy and crisp weights, as well as the final ranks of the criteria.

Table 4. Fuzzy and crisp weights of the criteria, and final ranks.

	Aggregated Fuzzy Comp. Val.	Fuzzy Weights	Crisp Weights	Normalized Crisp Weights	Rank
C1	(0.51; 0.60; 0.71)	(0.05; 0.07; 0.11)	0.079	0.076	6
C2	(0.86; 1.07; 1.34)	(0.09; 0.13; 0.20)	0.142	0.137	4
C3	(0.98; 1.19; 1.46)	(0.10; 0.15; 0.22)	0.157	0.151	3
C4	(1.83; 2.17; 2.50)	(0.19; 0.27; 0.37)	0.280	0.268	1
C5	(0.44; 0.52; 0.65)	(0.05; 0.07; 0.10)	0.070	0.067	7
C6	(0.56; 0.64; 0.73)	(0.06; 0.08; 0.11)	0.083	0.079	5
C7	(1.49; 1.81; 2.09)	(0.16; 0.23; 0.31)	0.232	0.223	2

Step 7. The criteria were prioritized based on normalized crisp weights given in column 5 of Table 4. The consistency ratio (CR) of the defuzzified matrix was calculated and validated (should be <0.1).

The normalized crisp weights and ranks of criteria can be considered as final results of the fuzzy AHP implemented above.

3.2. Risk Prioritization Using Fuzzy TOPSIS

In the following, the risk evaluation was performed by taking into account the results of the applied fuzzy AHP and utilizing the fuzzy TOPSIS approach.

Step 1. The pairwise comparison risk vs. criteria analysis was performed by the same expert group who performed the evaluation of the criteria. Similarly to above, the triangular fuzzy numbers and the linguistic variables were employed [37]. The linguistic variables for the evaluation of the importance of the risks with respect to criteria are presented in Table 5.

Table 5. Linguistic variables for the importance of the risks-s with respect to criteria.

The Relative Importance of the Risks with Respect to Criteria in Terms of Linguistic Variables	Crisp AHP Scale	Fuzzy Triangular	Reciprocal Fuzzy
Very Weak (VW)	1	1, 1, 1	1, 1, 1
Very Weak to Weak (VW-W)	2	1, 2, 3	1/3, 1/2, 1
Weak (W)	3	2, 3, 4	1/4, 1/3, 1/2
Weak to Average (W-A)	4	3, 4, 5	1/5, 1/4, 1/3
Average (A)	5	4, 5, 6	1/6, 1/5, 1/4
Average to Strong (A-S)	6	5, 6, 7	1/7, 1/6, 1/5
Strong (S)	7	6, 7, 8	1/8, 1/7, 1/6
Strong to Very Strong (S-VS)	8	7, 8, 9	1/9, 1/8, 1/7
Very Strong (VS)	9	8, 9, 9	1/9, 1/9, 1/8

Step 2. The risk evaluation with respect to criteria was performed. The sample results of one decision-maker are shown in Table 6.

Table 6. Risk vs. criteria evaluation.

	C1	C2	C3	C4	C5	C6	C7
A1	VS	A	VS	S	S	S	S
A2	VS	A	VS	VS	W	W	VS
A3	VS	S	S	A	W	W	VS
A4	VS	S	VS	W	W	W	S
A5	VS	A	VS	S	S	W	VS
A6	A	S	S	A	S	W	W
A7	S	S	VS	A	S	W	S
A8	A	S	A	A	S	W	W
A9	S	W	VS	S	S	S	A
A10	VS	S	VS	S	S	A	A

Step 3. The linguistic "grades" given by decision-makers (see Table 6) were transferred to triangular fuzzy numbers (TFN) based on the relations given in Table 5.

The aggregation of the decision-makers' evaluation matrices was performed by applying the fuzzy arithmetic mean (in the case of Fuzzy AHP was applied geometric mean) as:

$$x_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_{ijn}, \quad (5)$$

where N is the number of decision-makers and x_{ijn} stands for the rating of risk i to criterion j given by the n-th decision-maker. The computed fuzzy triangular numbers $x_{ij} = (l_{ij}, m_{ij}, u_{ij})$ are presented in Table 7.

Table 7. Aggregated pairwise comparison matrix.

	C1	C2	C3	C4	C5	C6	C7
A1	(7.67; 8.67; 8.83)	(4.67; 5.67; 6.50)	(8.00; 9.00; 9.00)	(6.67; 7.67; 8.33)	(6.00; 7.00; 7.67)	(5.33; 6.33; 7.17)	(5.67; 6.67; 7.50)
A2	(7.67; 8.67; 8.83)	(3.50; 4.33; 5.17)	(7.00; 8.00; 8.50)	(7.67; 8.67; 8.83)	(4.17; 5.00; 5.83)	(3.17; 4.00; 4.83)	(6.67; 7.67; 8.17)
A3	(5.83; 6.67; 7.00)	(3.83; 4.67; 5.50)	(4.33; 5.33; 6.33)	(4.00; 5.00; 5.83)	(4.67; 5.67; 6.33)	(2.67; 3.67; 4.67)	(5.00; 6.00; 6.67)
A4	(7.67; 8.67; 8.83)	(5.00; 6.00; 6.83)	(6.33; 7.33; 7.83)	(4.00; 5.00; 5.83)	(3.33; 4.33; 5.33)	(2.33; 3.17; 4.00)	(6.00; 7.00; 7.67)
A5	(6.67; 7.67; 8.00)	(5.67; 6.67; 7.33)	(7.00; 8.00; 8.33)	(6.00; 7.00; 8.00)	(4.50; 5.33; 6.17)	(3.17; 4.17; 5.00)	(7.67; 8.67; 8.83)
A6	(3.50; 4.33; 5.00)	(3.83; 4.50; 5.33)	(4.00; 5.00; 6.00)	(3.67; 4.67; 5.67)	(6.00; 7.00; 7.83)	(3.67; 4.67; 5.50)	(4.17; 5.17; 6.00)
A7	(5.33; 6.33; 7.17)	(7.00; 7.83; 8.33)	(6.67; 7.67; 8.17)	(6.00; 7.00; 7.50)	(7.33; 8.33; 8.67)	(3.67; 4.67; 5.67)	(6.67; 7.67; 8.17)
A8	(4.33; 5.33; 6.33)	(5.17; 6.00; 6.50)	(4.50; 5.33; 6.00)	(3.83; 4.67; 5.50)	(7.33; 8.33; 8.67)	(4.67; 5.67; 6.33)	(2.67; 3.67; 4.67)
A9	(5.00; 6.00; 6.83)	(2.67; 3.50; 4.33)	(5.17; 6.00; 6.50)	(4.83; 5.67; 6.50)	(4.50; 5.33; 6.17)	(6.67; 7.67; 8.33)	(3.17; 4.00; 4.83)
A10	(7.00; 8.00; 8.33)	(4.67; 5.67; 6.50)	(6.67; 7.33; 7.83)	(5.83; 6.83; 7.50)	(4.17; 5.00; 5.67)	(4.67; 5.67; 6.67)	(2.83; 3.67; 4.50)

Step 4. The aggregated fuzzy decision matrix was normalized. The fuzzy weights of the criteria obtained by applying fuzzy AHP (see Table 4) were utilized to compute the weighted normalized decision matrix given in Table 8.

Table 8. Weighted normalized fuzzy decision matrix.

	C1	C2	C3	C4	C5	C6	C7
A1	(0.05; 0.07; 0.10)	(0.05; 0.08; 0.15)	(0.03; 0.04; 0.06)	(0.14; 0.23; 0.35)	(0.03; 0.05; 0.08)	(0.03; 0.06; 0.09)	(0.10; 0.17; 0.26)
A2	(0.05; 0.07; 0.10)	(0.04; 0.06; 0.12)	(0.03; 0.13; 0.21)	(0.16; 0.26; 0.37)	(0.02; 0.04; 0.06)	(0.02; 0.04; 0.06)	(0.12; 0.19; 0.28)
A3	(0.04; 0.06; 0.08)	(0.04; 0.07; 0.12)	(0.05; 0.09; 0.15)	(0.09; 0.15; 0.24)	(0.02; 0.04; 0.07)	(0.02; 0.03; 0.06)	(0.09; 0.15; 0.23)
A4	(0.05; 0.07; 0.10)	(0.05; 0.09; 0.15)	(0.07; 0.12; 0.19)	(0.09; 0.15; 0.24)	(0.02; 0.03; 0.06)	(0.02; 0.03; 0.05)	(0.10; 0.18; 0.27)
A5	(0.04; 0.06; 0.09)	(0.06; 0.10; 0.16)	(0.08; 0.13; 0.20)	(0.13; 0.21; 0.33)	(0.02; 0.04; 0.07)	(0.02; 0.04; 0.06)	(0.13; 0.22; 0.31)
A6	(0.02; 0.04; 0.06)	(0.04; 0.07; 0.12)	(0.05; 0.08; 0.15)	(0.08; 0.14; 0.24)	(0.03; 0.05; 0.09)	(0.02; 0.04; 0.07)	(0.07; 0.13; 0.21)
A7	(0.03; 0.05; 0.09)	(0.07; 0.12; 0.19)	(0.08; 0.13; 0.20)	(0.13; 0.21; 0.31)	(0.04; 0.06; 0.09)	(0.02; 0.04; 0.07)	(0.12; 0.19; 0.28)
A8	(0.03; 0.04; 0.08)	(0.05; 0.09; 0.15)	(0.05; 0.09; 0.15)	(0.08; 0.14; 0.23)	(0.04; 0.06; 0.09)	(0.03; 0.05; 0.08)	(0.05; 0.09; 0.16)
A9	(0.03; 0.05; 0.08)	(0.03; 0.05; 0.10)	(0.06; 0.10; 0.16)	(0.10; 0.17; 0.27)	(0.02; 0.04; 0.07)	(0.04; 0.07; 0.10)	(0.06; 0.10; 0.17)
A10	(0.04; 0.07; 0.10)	(0.05; 0.08; 0.15)	(0.08; 0.12; 0.19)	(0.12; 0.21; 0.31)	(0.02; 0.04; 0.06)	(0.03; 0.05; 0.08)	(0.05; 0.09; 0.16)

Step 5. The distances of each risk to positive and negative ideal solutions were computed as

$$d_i^+ = \sum_{j=1}^{n} d\left(v_{ij}, v_j^+\right),\ i = 1, \ldots, m,\ d_i^- = \sum_{j=1}^{n} d\left(v_{ij}, v_j^-\right),\ i = 1, \ldots, m, \tag{6}$$

where

$$v_j^+ = (1,1,1),\ v_j^- = (0,0,0),\ j = 1,2,\ldots,n \tag{7}$$

and

$$d(x,y) = \sqrt{\left(\frac{1}{3}\right).\left[\left(l_x - l_y\right)^2 + \left(m_x - m_y\right)^2 + \left(u_x - u_y\right)^2\right]}. \quad (8)$$

Step 6. Based on the positive and negative ideal solution, the similarities were calculated as

$$C_i = \frac{d_i^-}{d_i^+ + d_i^-},\ i = 1, \ldots, m. \quad (9)$$

The risks were ranked based on the values of the similarities. Table 9 presents the positive and negative ideal solutions, the similarities, and the final ranking of the risks.

Table 9. Final ranking of the risks.

		d_i^+	d_i^-	C_i	**Rank**
A1	Mechanical failure	6.269	0.789	0.1118	4
A2	Electrical failure	6.204	0.872	0.1232	3
A3	Information shortage	6.378	0.679	0.0963	7
A4	Autonomous driving software failure	6.300	0.761	0.1078	5
A5	Low-level software failure	6.173	0.893	0.1263	2
A6	Communication bandwidth shortage	6.413	0.645	0.0914	10
A7	Cyber-hacking	6.171	0.893	0.1264	1
A8	Interruption of uplink	6.399	0.656	0.0930	9
A9	A drastic change in the environment	6.385	0.669	0.0949	8
A10	Loss of localization	6.309	0.749	0.1061	6

The estimation of a number of different types of risks and the evaluation of multiple criteria is a challenging task in the development of AV systems. The fuzzy AHP-TOPSIS-based risk analysis approach proposed here provides estimates of the ranks of criteria and risks. Cyber hacking, low-level software failure, and electrical failure appear to be the most critical risks in the current case study. The weights of criteria and similarity values of the risks are another valuable piece of information for the further improvement of AV systems.

As the results point out, low-level software failures are one of the highest risk factors and thus require a high level of attention during the system design stage and implementation stage. The following case study covers low-level system safety improvements for the TalTech iseAuto AV shuttle, which was designed and manufactured for research and educational purposes by the Autonomous Vehicles lab at Tallinn University of Technology.

4. Low-Level Communication and Safety Architecture for the AV Shuttle Based on the Risk Evaluation Model

The iseAuto AV shuttle was designed to be a minibus, with the aim of operating primarily on the territory of the university campus. Therefore, the speed of the minibus was limited to 20 km/h. The architecture of the vehicle CPS was first explained in [34], and it is divided into layers as described in Figure 2. The AI and high-level decision-making layer make autonomous driving decisions based on the sensor's input layer. The various controlling commands are sent to the actuator layer, which has a mission-critical functionality to take care of the robot's actual control.

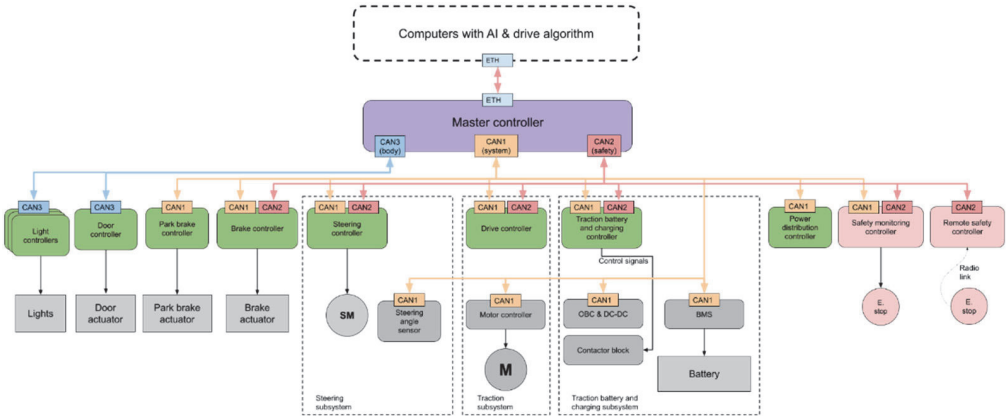

Figure 2. Low-level control solution for TalTech iseAuto v2.0.

The shuttle's control logic is divided into two layers—the master controller layer and the function-based controller layer. The main task of the master controller is to act as a central gateway between all the nodes. Function-based controllers are classified as critical or non-critical. Critical controllers are involved in the direct control of the vehicle or the control of the traction battery and its charging. For safety reasons, separate safety controllers have been added to stop the vehicle when a fault is detected. The communication is shared between three CAN buses:

- CAN 1 for all system controllers;
- CAN 2 for safety-related controllers and for duplicating critical system messages; and
- CAN 3 for vehicle body-related and other low-priority controllers.

The correct design of critical CAN networks is important. First, is it essential to choose the correct package IDs for CAN bus data frames. The data frames have an ID that can be used to separate data frames, and data frames are ranked in order of importance using this ID. Data frames with a lower ID are preferred [38]. An extra checksum and counter value can be added into critical data frames. The controller using the data frames will only do so if the checksum is correct. A possible reason for this is hacking because the CAN network is not encrypted. A 15-bit CRC checksum is added to every CAN message via a hardware layer anyway, but it is harder to inject the messages into the network if there is an extra checksum. Counter values are used to check if some data frame loss has occurred. For faster system diagnostics and error detection, a diagnostic data frame should be sent out by the ECU. For example, if the expected data frame does not arrive at the correct time interval, if the supply voltage limit is exceeded, or something else happens, the flag is set. Every diagnostic data frame on the CAN bus can carry 8 bytes of data or 64 flags. The safety controller monitors these flags and can decide to trigger a safety logic process. A similar logic is used in Tesla vehicles [39].

ECU components should comply with international automotive application standards. The previously used STM32 family microcontroller is not certified for automotive use. A good replacement for the STM32 is the general-purpose STMicroelectronics SPC5 family automotive microcontrollers, which qualify according to the AEC-Q100 standard and have a wide range of automotive interfaces. The chosen specialized hardware should allow the achievement of safety goals [40]. Passive components qualifying to AEC-Q200 and automotive connectors are used in the design of new ECUs. Automotive connectors should be crimp-type connectors in order to establish better connections and save time. For example, the WireLock low-mating-force automotive-grade connector system is a good option and is USCAR-2 V2-compatible.

It is good practice to design the ECU internal electronics as a fortress. This means that over- or undervoltages (provided that they remain within the selected limits), electrical

noise, and short circuits applied to power inputs, digital IO, or data interfaces, cannot interrupt the operation of the microcontroller. If the ECU has a power source for the sensors, and if this source is shorted or something draws too much current, microcontroller power should not be affected. Any such errors should be logged, and flags should be set and sent out by the diagnostic data frame on the CAN bus.

Authentication and secret key establishment, providing confidentiality and integrity to the in-vehicle network, makes it possible to design a process that does not violate the real-time constraints of automotive CPS applications even in the presence of errors in computation and transmission [41]. Furthermore, it is possible to integrate both security and dependability principles simultaneously in the design of ECUs with a negligible performance, energy, and resource overhead [42]. The ISO 26262 standard requires that at least one critical fault must be tolerated by the automotive applications to maintain intended functionality or achieve or maintain a safe state [28], and the ASIL, risk classification system, must be used to mitigate the risks when designing every ECU.

The power system can be built using regular automotive fuses. Today's state-of-the-art cars use electronic protection circuits for replacing fuse and relay boxes [43]. Electronic protection circuits are not only faster but also allow faults to be logged as soon as they occur. In addition to feeding the critical controllers, two separately protected supply lines can be added. For example, the steering controller, when power electronics and their controlling circuits are duplicated, is a good candidate. In this case, if one power line is faulty or short-circuited, the other will continue to work.

If something unexpected happens, then the safety logic is triggered, as shown in Figure 3. It is divided into three stages:

1. Normal braking is usually triggered by a high-level computer or safety lidar. When there is free room regenerative braking can be used, followed by normal braking if needed;
2. The emergency brake is triggered when the emergency STOP switch is pressed, the front safety lidar sees something that is too close, or when the safety monitoring controller is triggered by some fatal error;
3. An emergency shutdown may be followed by emergency braking when the emergency STOP switch is pressed (for example, a risk of fire because there is smoke in the cabin), the crash detection system is triggered, or some serious error is detected. Emergency shutdown disables the high-voltage traction battery.

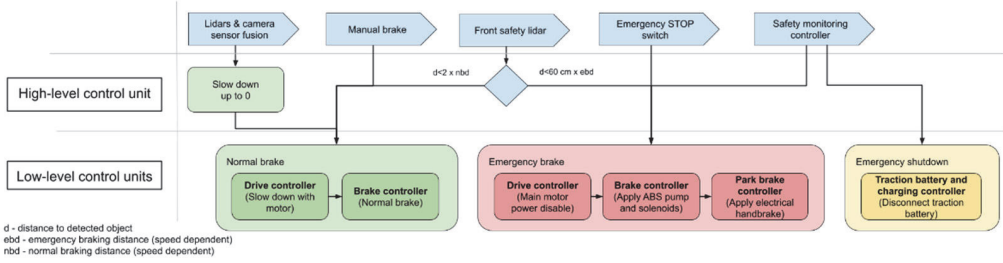

Figure 3. Safety triggering logic.

Normal and emergency braking is based on brake-by-wire (BBW) technology, which should cooperate with the regenerative braking system controlled by the drive controller ECU. The hydraulic brake system is made controllable by replacing the master cylinder with a gear pump. The intensity of the braking depends on the pressure of the brake fluid. The speed of the pump is controlled according to the feedback from the brake fluid pressure sensor and the required braking force sent by a high-level control system. The valve must be opened to release the brake. One of the biggest disadvantages of this system is that it is difficult to release the brake precisely and smoothly. The solution is to develop a distributed brake-by-wire system, as proposed in [44], which has a hydraulic actuator for

every wheel. This provides flexible and precise braking force control with shorter or no brake pipes. A disadvantage of this system is the lack of freely available brake components. Bosch developed a brake booster system called iBooster, which is used in Tesla and other cars capable of automatic driving. The brake pressurization rate of the iBooster is three times that of the conventional braking system, and it was meant to replace vacuum brake boosters [45]. Bosch iBooster is available as a spare part, but further research and testing are required to control it over the CAN bus. iBooster is compatible with the classic hydraulic braking system. In addition to normal brakes, a parking brake is also available in the iseAuto AV shuttle, controlled by an electric drive. This is intended primarily to prevent the vehicle from moving on its own but can be used in an emergency when the main brake is not working.

Self-driving vehicles do not have a driver who can detect problems directly. One of the most likely problems is a low tire pressure or flat tire. Tire pressure plays an important role in safety and energy consumption. If the AI and high-level decision-making layer of the self-driving vehicle are not alerted to this issue, a dangerous situation can arise. Today's vehicles use a tire-pressure monitoring system (TPMS). The TPMS measures the air pressure inside the pneumatic tires. Inside the stem of every wheel, an electronic unit is located that contains a pressure sensor, microcontroller, radio link, and battery. The TPMS control ECU has a radio receiver that reads pressure information. Methods to implement TPMS systems have been described [46], but in most cases, such systems are intended to warn the driver. The new iseAuto AV shuttle should be equipped with some sort of TPMS system to make it more secure. As a further development of the TPMS, it is possible to measure dangerous impacts on tires (to measure pressure pikes) when a vehicle accidentally drives against a road curb or against some objects on the road. If TPMS is triggered, the vehicle should probably park safely so as not to obstruct traffic and to call for help.

5. Conclusions

The final results of the study can be outlined as follows.

- An MCDM risk evaluation model was developed for safety system assessment;
- A list of prioritized risks was developed, as presented in Table 9;
- The most critical risks were determined to be cyber hacking, low-level software failure, and electrical failure.

First, the criteria and risks were defined in a previous study by the authors. Drawing on the results of that study, the seven criteria and ten risks were formulated and described.

Next, the criteria were prioritized by applying the fuzzy analytical hierarchy process. As a result, the sensor system (reliability of the sensors), the performance of low-level cyber-physical systems, and the malfunctioning of AV mechanical components were identified as the most important criteria for decision-making.

Finally, the risks were prioritized by utilizing the Technique for Order of Preference by Similarity to Ideal Solution method. As a result, cyber hacking, low-level software failure, and electrical failure were found to be the most critical risks for the current case study.

Based on the analysis of the highest risk affecting full system safety, low-level system safety criteria were selected in this research as an improvement option. The main ideas for testing of the improved solution for the low-level system architecture were proposed and briefly analyzed in the context of a particular AV shuttle—the TalTech iseAuto.

The information provided on the ranking of the criteria and risks consists only of positions, as a rule, without providing detailed information on how far are values from each other, etc. The crisp weights of the criteria and the similarity values of the risks provide more detailed and valuable information for the further improvement of mobile robot systems.

The approach proposed here may be used to simplify decision-maker's judgments and to handle uncertainty caused by these judgments. The risks identified here are rather universal, applicable not only to a specific autonomous shuttle design, but also to similar outdoor mobile robots and other low-speed automated vehicles. The risk evaluation results

can provide an input for further developments and improvements of AVs and, in particular, for the TalTech iseAuto version 2, which is under development.

Author Contributions: Conceptualization, H.P. and J.M.; methodology, H.P. and J.M.; model development, J.M.; validation, R.S. and K.K.; formal analysis, H.P.; investigation, H.P.; resources, R.S.; data curation, J.M.; writing—original draft preparation, H.P.; writing—review and editing, H.P., J.M., K.K. and R.S.; visualization, H.P.; supervision, R.S.; project administration, R.S.; funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: 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.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The financial support from the Estonian Ministry of Education and Research and the Horizon 2020 Research and Innovation Programme is gratefully acknowledged.

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

References

1. Della Cava, M. Tesla Announces Fully Self-Driving Cars. USA Today, 2016. Available online: https://eu.usatoday.com/story/tech/news/2016/10/19/tesla-announces-fully-self-driving-fleet/92430638/ (accessed on 13 March 2022).
2. Korosec, K. Ford Postpones Autonomous Vehicle Service until 2022. *TechCrunch*, 28 April 2020. Available online: https://techcrunch.com/2020/04/28/ford-postpones-autonomous-vehicle-service-until-2022/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAADBFTUMYSsgWbXuqaxjPCxHsMVa-3xDxahKGV33qvhPjg0sPUdDXuypt_zViyxxg-nZe8HSlMZWfgvGWu9ch1uB0Sa4fnxRslcxGyh5xfICKKji9dPOz4JLHXH9U-QLnno5a3WN5YnJ9F9o4qt-7C76fa9ULO6mkuCGMXLNRns2x (accessed on 21 December 2021).
3. Sell, R.; Rassolkin, A.; Wang, R.; Otto, T. Integration of Autonomous Vehicles and Industry 4.0. *Proc. Eston. Acad. Sci.* **2019**, *68*, 389. [CrossRef]
4. Shuttleworth, J. SAE Standard News: J3016 Automated-Driving Graphic Update, 2019. Available online: https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic (accessed on 20 December 2021).
5. Sell, R.; Leier, M.; Rassõlkin, A.; 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, The Netherlands, 7–8 June 2018; pp. 111–116. [CrossRef]
6. Rassõlkin, A.; Sell, R.; Leier, M. Development Case Study of the First Estonian Self-Driving Car, Iseauto. *Electr. Control Commun. Eng.* **2018**, *14*, 81–88. [CrossRef]
7. Sell, R.; Coatanéa, E.; Christophe, F. Important Aspects of Early Design in Mechatronic. In Proceedings of the 6th International DAAAM Baltic Conference, Tallinn, Estonia, 24–26 April 2008.
8. Sell, R.; Petritsenko, A. Early Design and Simulation Toolkit for Mobile Robot Platforms. *Int. J. Prod. Dev.* **2013**, *18*, 168. [CrossRef]
9. Mahmood, K.; Karjust, K.; Raamets, T. Production Intralogistics Automation Based on 3D Simulation Analysis. *J. Mach. Eng.* **2021**, *21*, 101–115. [CrossRef]
10. Pikner, H.; Karjust, K. Multi-Layer Cyber-Physical Low-Level Control Solution for Mobile Robots. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1140*, 012048. [CrossRef]
11. Ziyan, C.; Shiguo, L. China's Self-Driving Car Legislation Study. *Comput. Law Secur. Rev.* **2021**, *41*, 105555. [CrossRef]
12. Safety First for Automated Driving. Available online: https://newsroom.intel.com/wp-content/uploads/sites/11/2019/07/Intel-Safety-First-for-Automated-Driving.pdf (accessed on 24 December 2021).
13. Yue, H.; Medromi, H.; Ding, H.; Bassir, D. A novel hybrid drone for multi-propose aerial transportation and its conceptual optimization based on surrogate approach. *J. Phys. Conf. Ser.* **2021**, *1972*, 12103. [CrossRef]
14. Guessasma, S.; Bassir, D. Neural network computation for the evaluation of process rendering: Application to thermally sprayed coatings. *Int. J. Simul. Multisci. Des. Optim.* **2017**, *8*, A1. [CrossRef]
15. Tang, X.G.; Rezoug, M.; Hamzaoui, R.; Bassir, D.; El Meouche, R.; Hreim, J.F.; Feng, Z.Q. Multiobjective optimization on urban flooding using RSM and GA. *Adv. Mater. Res. Adv. Civ. Eng.* **2011**, *255–260*, 1627–1631. [CrossRef]

16. Guessasma, S.; Bassir, D. Comparing heuristic and deterministic approaches to optimize mechanical parameters of biopolymer composite materials. *Mech. Adv. Mater. Struct.* **2009**, *16*, 293–299. [CrossRef]
17. Herranen, H.; Majak, J.; Tsukrejev, P.; Karjust, K.; Märtens, O. Design and Manufacturing of composite laminates with structural health monitoring capabilities. *Procedia CIRP* **2018**, *72*, 647–652. [CrossRef]
18. Lasn, K.; Klauson, A.; Chati, F.; Décultot, D. Experimental determination of elastic constants of an orthotropic composite plate by using Lamb waves. *Mech. Compos. Mater.* **2011**, *47*, 435–446. [CrossRef]
19. Lasn, K.; Klauson, A. Non-destructive identification of elastic constants by vibration measurements and optimization. In Proceedings of the OAS 2011: International Conference on Optimization and Analysis of Structures, Tartu, Estonia, 25–27 August 2011.
20. Lasn, K.; Echtermeyer, A.T.; Klauson, A.; Chati, F.; Décultot, D. Comparison of laminate stiffness as measured by three experimental methods. *Polym. Test.* **2015**, *44*, 143–152. [CrossRef]
21. Frolovs, G.; Rocens, K.; Sliseris, J. Optimal design of plates with cell type hollow core. *IOP Conf. Ser. Mater. Sci. Eng.* **2017**, *251*, 12075. [CrossRef]
22. Sliseris, J.; Buka-Vaivade, K. Numerical Modelling of High Strength Fibre-Concrete's columns in Multi-Storey Building. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *660*, 012062. [CrossRef]
23. Vinodh, S.; Prasanna, M.; Hari Prakash, N. Integrated Fuzzy AHP-TOPSIS for selecting the best plastic recycling method: A case study. *Appl. Math. Model.* **2014**, *38*, 4662–4672. [CrossRef]
24. Bakioglu, G.; Atahan, A.O. AHP integrated TOPSIS and VIKOR methods with Pythagorean fuzzy sets to prioritize risks in self-driving vehicles. *Appl. Soft Comput.* **2020**, *99*, 106948. [CrossRef]
25. Harrison, M.; Yang, Z.; Nguyen, T.T.; Kavakeb, S.; Wang, J.; Bonsall, S. A TOPSIS method for vehicle route selection in seaports—A real case analysis of a container terminal in North West Europe. In Proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS), Wuhan, China, 25–28 June 2015; pp. 599–606. [CrossRef]
26. Pachêco Gomes, I.; Renan Bruno, D.; Santos Osório, F.; Fernando Wolf, D. Diagnostic Analysis for an Autonomous Truck Using Multiple Attribute Decision Making. In Proceedings of the 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), Pessoa, Brazil, 6–10 November 2018; pp. 283–290. [CrossRef]
27. Emovon, I.; Oghenenyerovwho, O.S. Application of MCDM method in material selection for optimal design: A review. *Results Mater.* **2020**, *7*, 100115. [CrossRef]
28. Debouk, R. *Overview of the 2nd Edition of ISO 26262: Functional Safety-Road Vehicles*; General Motors Company: Warren, MI, USA, 2018. [CrossRef]
29. *ISO 26262*; Road Vehicles—Functional Safety—Part 2: Management of Functional Safety. International Organization for Standardization: Geneva, Switzerland, 2018.
30. *IATF 16949*; Quality Management System Requirements for Automotive Production and Relevant Service Parts Organisations. Automotive Industry Action Group: Southfield, MI, USA, 2016; ISBN 9781605343471.
31. Automotive Electronics Council. *Failure Mechanism Based Stress Test Qualification for Integrated Circuits*; AEC Q100 Rev. H; Automotive Electronics Council: Luton, UK, 2014.
32. Automotive Electronics Council. *Stress Test Qualification for Passive Components*; AEC Q200 Rev. D; Automotive Electronics Council: Luton, UK, 2010.
33. SAE MOBILUS. Available online: https://saemobilus.sae.org/content/uscar2-7 (accessed on 12 November 2021).
34. Karjust, K.; Majak, J.; Pikner, H.; Sell, R. Multi-Layer Cyber-Physical Control Method for Mobile Robot Safety Systems. *Proc. Est. Acad. Sci.* **2021**, *70*, 383. [CrossRef]
35. Kaganski, S.; Majak, J.; Karjust, K. Fuzzy AHP as a Tool for Prioritization of Key Performance Indicators. *Procedia CIRP* **2018**, *72*, 1227–1232. [CrossRef]
36. Paavel, M.; Karjust, K.; Majak, J. PLM Maturity Model Development and Implementation in SME. *Procedia CIRP* **2017**, *63*, 651–657. [CrossRef]
37. Paavel, M.; Karjust, K.; Majak, J. Development of a Product Lifecycle Management Model Based on the Fuzzy Analytic Hierarchy Process. *Proc. Est. Acad. Sci.* **2017**, *66*, 279. [CrossRef]
38. Davis, R.I.; Burns, A.; Bril, R.J.; Lukkien, J.J. Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised. *Real Time Syst.* **2007**, *35*, 239–272. [CrossRef]
39. Lab, T.K.S. *Experimental Security Research of Tesla Autopilot*; Tencent Keen Security Lab: Shenzhen, China, 2019.
40. SPC5 32-Bit Microcontroller Series Featuring Power Architecture, 2016. Available online: https://www.st.com/content/ccc/resource/sales_and_marketing/presentation/product_presentation/81/61/89/8b/77/1b/42/5f/SPC5_Family_Overview.pdf/files/SPC5_Family_Overview.pdf/jcr:content/translations/en.SPC5_Family_Overview.pdf (accessed on 25 December 2021).
41. Giri, N.; Munir, A.; Kong, J. An Integrated Safe and Secure Approach for Authentication and Secret Key Establishment in Automotive Cyber-Physical Systems. In *Intelligent Computing. SAI 2020*; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2020. [CrossRef]
42. Poudel, B.; Munir, A. Design and Evaluation of a Reconfigurable ECU Architecture for Secure and Dependable Automotive CPS. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 235–252. [CrossRef]

43. Gysen, L.; Ayeb, M.; Brabetz, L. Cable Bundle Protection and Cross-Section Reduction by Using a Centralized Smart Fusing Strategy. In Proceedings of the 2018 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC), Nottingham, UK, 7–9 November 2018; pp. 1–5. [CrossRef]
44. Wang, Z.; Yu, L.; You, C.; Wang, Y.; Song, J. Fail-Safe Control Allocation for a Distributed Brake-by-Wire System Considering the Driver's Behaviour. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2014**, *228*, 1547–1567. [CrossRef]
45. Liu, H.; Deng, W.; He, R.; Qian, L.; Yang, S.; Wu, J. Power Assisted Braking Control Based on a Novel Mechatronic Booster. *SAE Int. J. Passeng. Cars Mech. Syst.* **2016**, *9*, 885–891. [CrossRef]
46. Hasan, N.N.; Arif, A.; Hassam, M.; Ul Husnain, S.S.; Pervez, U. Implementation of Tire Pressure Monitoring System with Wireless Communication. In Proceedings of the 2011 International Conference on Communications, Computing and Control Applications (CCCA), Hammamet, Tunisia, 3–5 March 2011; pp. 1–4. [CrossRef]

Appendix 3

III

H. Pikner, R. Sell, and E. Malayjerdi, “Level 4 commercial autonomous vehicle control system transition to an open-source solution,” *Proc. Eston. Acad. Sci.*, vol. 73, no. 2, pp. 124–133, 2024

Proceedings of the Estonian Academy of Sciences,
2024, **73**, 2, 124–133
https://doi.org/10.3176/proc.2024.2.05
Available online at www.eap.ee/proceedings

AUTONOMOUS VEHICLES

Level 4 commercial autonomous vehicle control system transition to an open-source solution

Heiko Pikner[a*], Raivo Sell[a,b] and Ehsan Malayjerdi[a]

[a] Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia
[b] FinEst Centre for Smart Cities, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia

Received 8 September 2023, accepted 31 October 2023, available online 26 March 2024

Abstract. This paper introduces a novel approach for transferring the entire set of low-level control systems from one robot bus, i.e., an autonomous vehicle (AV) shuttle, to another with distinct electronics and mechanical specifications. The research entails a series of experiments to assess the reliability and safety of the AV shuttle after integrating the critical control systems responsible for steering, accelerating, and braking into the target shuttle. The ultimate goal is to meet the necessary requirements for registering the target AV shuttle as a legal vehicle on the roads in Estonia. Consequently, several crucial tests of the shuttle's low-level control system were conducted, e.g., intentionally disconnecting different subsystems to simulate sudden failures and evaluate whether the shuttle responds in accordance with the appropriate protocols. As a case study, the upgraded autonomous shuttle was tested on the streets of Tallinn. The most relevant findings are introduced in the second part of this paper. The outcomes of the study demonstrate the feasibility of seamlessly transferring low-level control systems between various models of autonomous shuttles, eliminating the risk of encountering safety or reliability issues.

Keywords: cyber-physical system, autonomous vehicle, robot bus, low-level control, open-source software.

1. INTRODUCTION

The advancement of automated vehicles (AVs) has recently sparked hopes for a future with fully driverless transportation, boasting improved efficiency, enhanced traffic safety, and energy conservation. The concept of AVs has a rich history, with one of the earliest notable examples dating back to the 1950s when General Motors pioneered an automation system embedded alongside roads, as the technology then did not permit integration within the vehicles. Nevertheless, this marked a significant step towards envisioning autonomous driving [1].

The actual realization of AVs began to materialize in the new millennium. In 1998, the ARGO vehicle achieved a remarkable feat by successfully completing a driving test spanning over 2000 km on an Italian highway, signaling the dawn of driverless vehicle history [2]. The contemporary definition of AVs revolves around their reliance on sensors to perceive their surroundings and computer technologies to make informed decisions. This definition was first practically demonstrated during the DARPA Grand and Urban Challenges competition held between 2005 and 2007 [3].

Following the 2010s, there was a surge in the development of AVs, with numerous companies and research groups investing substantial resources into creating commercialized technologies and experimental platforms [4,5]. The blossoming interest in AVs has set the stage for their potential widespread adoption and integration in various industries in the near future.

Among commercialized technologies, the advanced driver assistance system (ADAS) stands out as one of the most successful and widely adopted technologies in commercial vehicles. Its primary function is to offer basic

* Corresponding author, heiko.pikner@taltech.ee

Fig. 1. The TalTech iseAuto (left) and Navya Evo (right) autonomous shuttles. Photo by Heiko Pikner.

assisted features, such as distance control, lane keeping, and collision warning. ADAS represents a significant research direction focused on object perception to enable intelligent decision-making. Over the years, advancements in the sensor industry and computing power have driven remarkable progress in corresponding techniques.

Two key technologies that have contributed to the success of ADAS are light detection and ranging (LiDAR) sensors and computer vision. These technologies allow vehicles to overcome weather limitations and achieve precise detection and classification of objects, enhancing overall safety and performance.

While the industry has mostly relied on mature technologies, the research community has shown considerable interest in experimental autonomous driving platforms. Examples like [6] and [7] involve testing autonomous driving algorithms in vehicles for civilian usage. Developing low-speed AV shuttles, also known as robot buses, further seeks to explore the practical potential of autonomous vehicles in real-world scenarios. The deployment of such real-traffic AV shuttles could potentially reshape human transportation habits. One notable AV shuttle is the iseAuto shuttle (depicted as the left one in Fig. 1), designed and developed by the autonomous vehicles research group at TalTech, Estonia [8,9]. The iseAuto shuttle represents a significant step forward in the autonomous driving domain, and its success could pave the way for further advancements in the field.

The emphasis on reliability and safety has been paramount in developing autonomous vehicles since their conception in research communities. Unlike human drivers who rely on their sentient brains as sensors and computers to perceive the environment and make decisions, AVs require cutting-edge technology to replicate these functions. In traditional driving, the physical control of the steering wheel, brakes, and throttle by human hands and feet ensures the vehicle's safety. However, for autonomous vehicles, extensive research has centered around the perception-decision aspect, aiming to attain a comprehensive understanding of the environment and flawless decision-making capabilities. Nevertheless, some perspectives argue that the low-level control systems hold greater significance for AV safety than the perception-decision stage. The precise and fail-safe execution of critical steering, speed, and brake controls in AVs leaves little room for mistakes. Therefore, it is imperative to subject the AV's low-level control system to rigorous failure-proof and accuracy tests before deploying these vehicles into real traffic.

By prioritizing safety at both the perception-decision stage and the low-level control systems, researchers and developers endeavor to instill the highest levels of confidence in AV technology, ensuring its seamless integration into real-world transportation scenarios.

An often chosen platform for testing autonomous driving is commercial vehicles due to their well-tested

suspension, car frame, and other mechanical components. However, adapting these vehicles for autonomy requires significant electronic modifications to enable computer operation. For instance, in their research, Wei et al. [10] integrated multiple actuation/electronic control modules into a Cadillac SRX to achieve full autonomous capabilities for brake, throttle, steering, and transmission shifting systems.

Regarding low-speed AV shuttles, the controlling systems differ as they lack traditional components, such as steering wheels and brake/throttle paddles. Instead, manual control relies heavily on joystick controllers, while telecontrol utilizes simulated steering wheels. Consequently, the entire low-level control system must be extensively customized for each vehicle.

A significant contribution of their work is the successful transfer of the low-level control system from the TalTech iseAuto AV shuttle [11] to the Navya Evo AV shuttle (depicted as the right one in Fig. 1). The Navya AV shuttle, a mature French-made self-driving product in the market, had previous piloting experience on Estonian roads [12]. However, by the end of the pilot, the vehicle's software was outdated, and the contract with the manufacturer had concluded. Despite these challenges, reliability and performance testing of the Navya AV shuttle demonstrated the feasibility of migrating iseAuto's low-level control system to another type of AV shuttle with different hardware specifications. This achievement opens up possibilities for using proven autonomous technologies in various AV models, enhancing their safety and efficiency.

2. TRANSITION OF THE LOW-LEVEL CONTROL SYSTEM

In our research and development efforts, we have successfully constructed two autonomous vehicles, namely a full-scale AV shuttle – iseAuto [13] – and a warehouse logistic robot – BoxBot [14]. As we continue to progress, our team now focuses on transferring our advanced low-level control system to an open-source platform, ensuring its adaptability to various types of autonomous vehicles. This step aims to foster collaboration and innovation within the autonomous vehicle community, as a universal and modular low-level control solution can greatly facilitate the promotion and widespread deployment of autonomous technologies.

By making our low-level control system open-source, we enable other researchers, developers, and manufacturers to leverage our expertise and integrate our proven technology into their AV projects. The versatility of this solution ensures seamless integration into different vehicle models, reducing the development time and resources required for implementing autonomous functionalities. Such accessibility can accelerate the overall advancement of autonomous technology and pave the way for a safer and more efficient future of transportation.

To validate the effectiveness and compatibility of our low-level control system in different vehicles, we conducted a series of comprehensive tests and experiments on the Navya shuttle. The Navya shuttle serves as an excellent testbed for evaluating the adaptability and robustness of our control system. Through these rigorous assessments, we ensure that the transferred solution meets the highest standards of safety, reliability, and performance, laying the groundwork for its real-world implementation.

Our vision is to contribute significantly to the growth of the autonomous vehicle ecosystem by fostering collaboration and knowledge-sharing across the industry. By making our low-level control system openly accessible, we aspire to catalyze advancements in autonomous technology, fueling its widespread adoption and transforming the way we experience transportation in the modern world. Prior to any modifications, the self-driving shuttle Navya had the capability to autonomously traverse a predefined route. However, this required an expensive and time-consuming analysis and assessment process. The shuttle's supplier was responsible for recording and editing the 3D LiDAR map and driving path using their own proprietary models and software [15]. Consequently, implementing the shuttle on a new route or making changes to existing routes necessitated the presence of a specialized team from the vehicle manufacturer.

The process of converting the existing self-driving shuttle into an open-source solution comprises several design stages. As of now, the vehicle manufacturer has not disclosed any details about the performance and technical solutions of the vehicle. The original shuttle, which was operated through a joystick, lacked the capability for autonomous driving.

The initial phase entails examining and charting the current low-level architecture. The primary focus is on identifying the original control computers and their data connections to the vehicle's low-level systems. Communication with the low-level vehicle system is facilitated through the use of the controller area network (CAN), a well-established multi-master broadcast serial bus communication protocol employed for linking electronic control units (ECUs) in automotive applications [16]. Moreover, the vehicle is equipped with an ethernet network that allows the two control computers to communicate not only with each other but also with higher-level sensors like LiDAR sensors.

In the second stage, the focus shifts to logging the CAN messages from all three identified networks. Each message possesses a distinctive CAN ID for easy identification. To determine the CAN network speeds, various

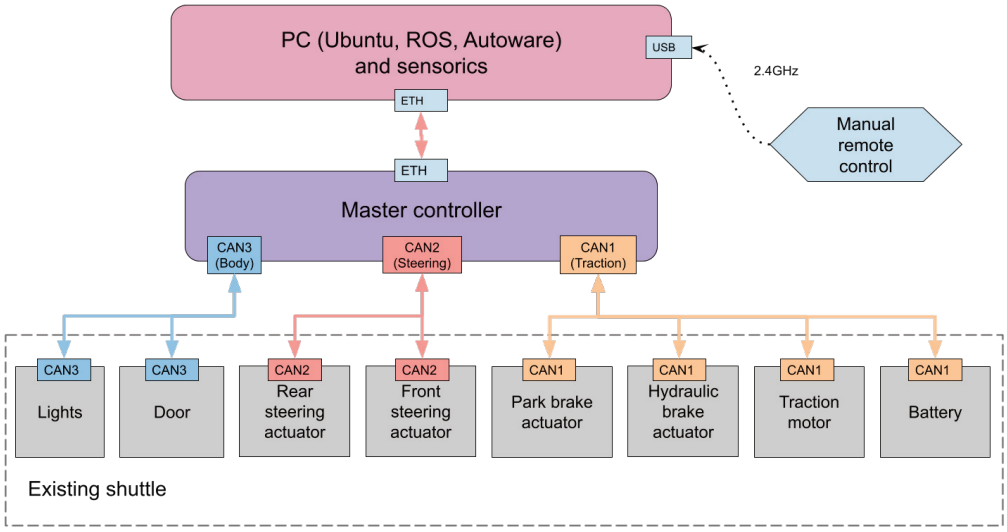

Fig. 2. Updated hardware architecture for the shuttle.

standard settings were experimented with. A custom-built gateway controller was employed separately to discern the direction of packets. For instance, the mapping of packets sent by the original control computer for each CAN network was accomplished through this process.

In the third phase, the main objective is to identify the data within the data field of the CAN packets. The SavvyCAN DBC files serve as repositories for definitions of how the data are formatted on the bus. By processing the raw data, it becomes possible to extract various parameters, such as RPM, odometer readings, and more [17]. To determine important parameters, adjustments were made using the existing joystick or touchscreen, while monitoring the changes in the CAN packets transmitted by the original control computer. The existing data, including specified ranges like the minimum and maximum steering angle, speed, and other signals, were thoroughly documented.

Moreover, it is essential to find feedback for each crucial signal, enabling the utilization of a regulator such as the proportional integral derivative controller (PID), which facilitates monitoring the execution of commands. This ensures that the control system can function effectively by providing necessary feedback and verification.

In the final step, the process involves establishing bidirectional communication for all the necessary messages required to control the shuttle, as illustrated in Fig. 2. To accomplish this, an existing in-house developed master controller [13] is utilized as the central control unit. To facilitate the integration of new vehicle-specific messages, additional software layers are added to the master controller.

Three distinct CAN buses are identified and connected to the master controller: CAN1 for traction and battery, CAN2 for steering, and CAN3 for body-related systems. Furthermore, a new control computer equipped with open-source software (Ubuntu, ROS, Autoware) is introduced into the system. To interconnect this new control computer with the existing lidars, cameras, and a mobile internet access point, a novel ethernet network is established.

Once these modifications are implemented, the self-driving shuttle becomes operational and capable of driving. The subsequent focus lies in fine-tuning and testing the vehicle to achieve autonomous driving capabilities.

3. EXPERIMENTS

Creating a safe and dependable solution necessitates conducting numerous experiments. The initial step involves integrating steering system control signals into the master controller. During this stage, it is imperative to determine the range of control and feedback signals. To facilitate further analysis, data logging is carried out to capture relevant information. For instance, Fig. 3 illustrates how the control signal sent by the original control computer considers the movement speed of the steering system. The vehicle possesses both front and rear axles,

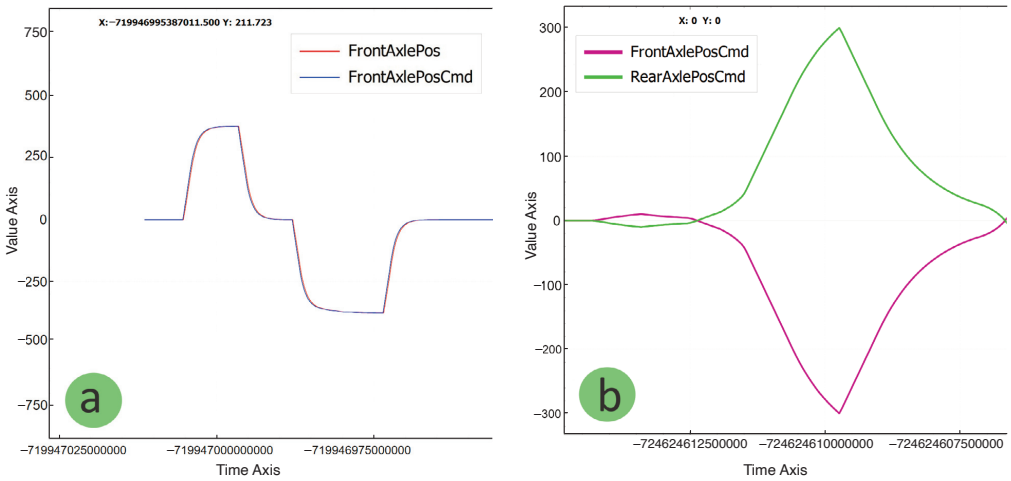

Fig. 3. Steering signals with the original control system: (a) relationship between position and feedback signals, (b) relationship between front and rear axle signals.

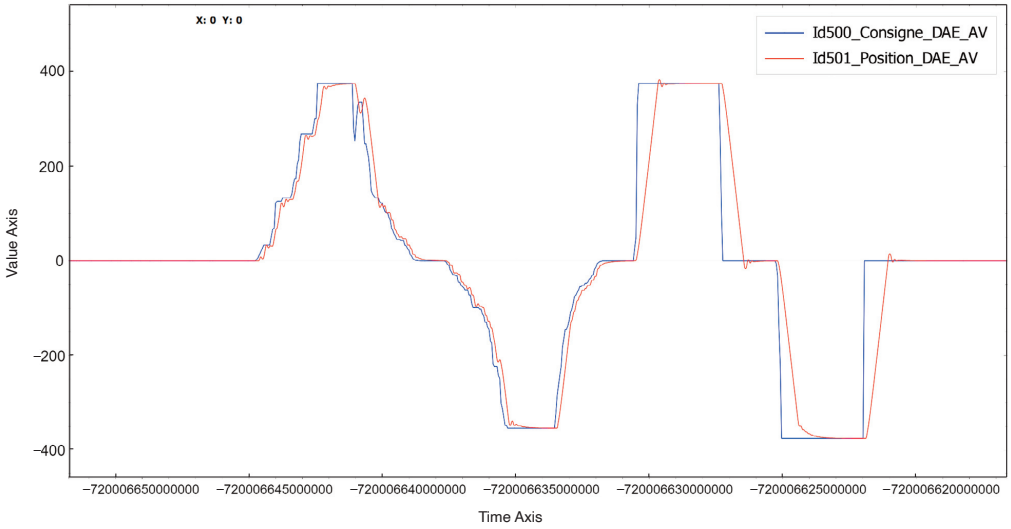

Fig. 4. A custom master controller sends out steering signals. Slow and high-speed movements are requested.

and their turning capabilities are taken into account. Notably, the values of the rear and front axle control signals differ in sign.

Moreover, an innovative approach is devised to enable independent control of the front and rear axles, a feature expected to be beneficial in various future experiments. Figure 4 showcases the steering signals transmitted by the master controller. Experiments were conducted to measure the speed at which the axle could mimic changes in the control signal. As anticipated, the maximum axis

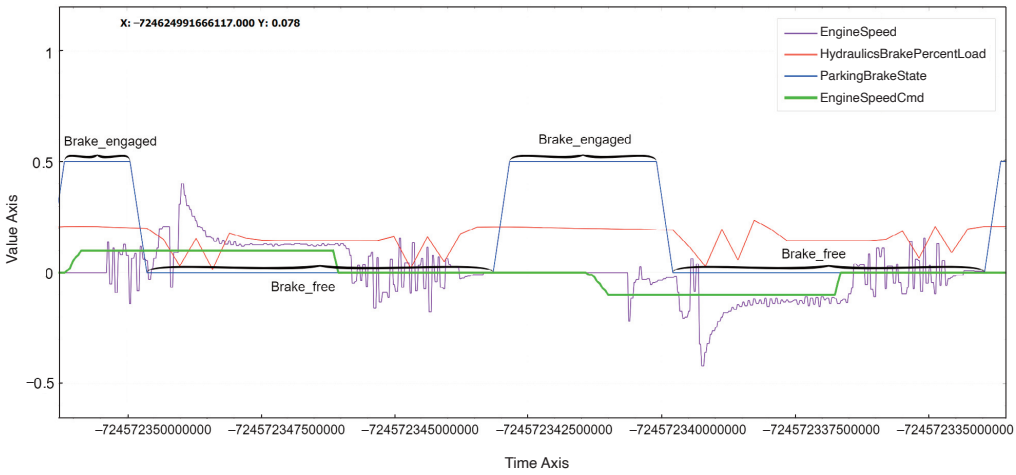

Fig. 5. Engine speed request sent out by the original control system activates the hydraulic brake and handbrake.

movement speed remains constant, and when reached, the actual position of the axle lags behind the required position.

The steering signal processing in the master controller follows a straightforward approach. The desired steering angle is conveyed through a user datagram protocol (UDP) packet from ROS, where it is converted from radians to degrees. Subsequently, a vehicle-specific CAN packet is generated based on this information. Moreover, the position signals of both axles are forwarded to ROS as feedback, completing the steering signal processing loop.

Moving on to the next step, the focus is on implementing the traction motor speed and control signals. The traction motor ECU awaits a status signal, which can either be in "use" or "standby" mode. However, managing the speed of the traction motor and braking presents a more intricate challenge. The vehicle features both a hydraulic brake and an electric handbrake that engages when the shuttle comes to a stop, as depicted in Fig. 5.

Upon scrutinizing the packets and analyzing the logs, it becomes evident that the corresponding ECU governs the brakes by utilizing the engine speed signal. As a result, inverting the engine speed signal causes the vehicle to move in reverse. When the engine speed signal reaches zero, the hydraulic brakes are engaged first, and as the shuttle comes to a stop, the handbrake is also applied to ensure a complete halt.

The speed signal processing within the master controller is relatively simpler compared to the steering signal. The desired speed signal is encapsulated in a UDP packet sent by ROS, which is then used to form a vehicle-specific CAN packet. Similarly, the traction motor control signal, represented by a one-byte flag, is processed in a manner similar to the iseAuto gear signal.

The engine speed request is transmitted by the master controller. As anticipated, the traction motor's speed adheres to the input signal. Additionally, the master controller forwards the speed feedback signals to ROS, enabling basic telemetry and speed regulation functionalities.

The autonomous shuttle underwent an extensive two-month testing phase within a specific district of the city, following a prescribed 1.1-kilometer route illustrated in Fig. 6. Subsequently, we harnessed a PostgreSQL database to meticulously record crucial data from the autonomous shuttle, which is structurally depicted in Fig. 8. These data empower us to conduct a thorough analysis of the shuttle's performance and behavior during the testing phase.

PostgreSQL, recognized as the world's leading open-source database management system (DBMS), provides comprehensive support for an array of structured query language (SQL) transactions, concurrent control mechanisms, and contemporary features. These include intricate query capabilities, trigger functionalities, view creation, transactional reliability, and the flexibility to integrate data type extensions, functions, operators, and procedural languages [18]. The database is organized into two distinct sections: the first segment houses higher-level data, encompassing sensor data, localization parameters, trajectory planning, and tracking parameters. In contrast, the second section manages vehicle status data at a lower

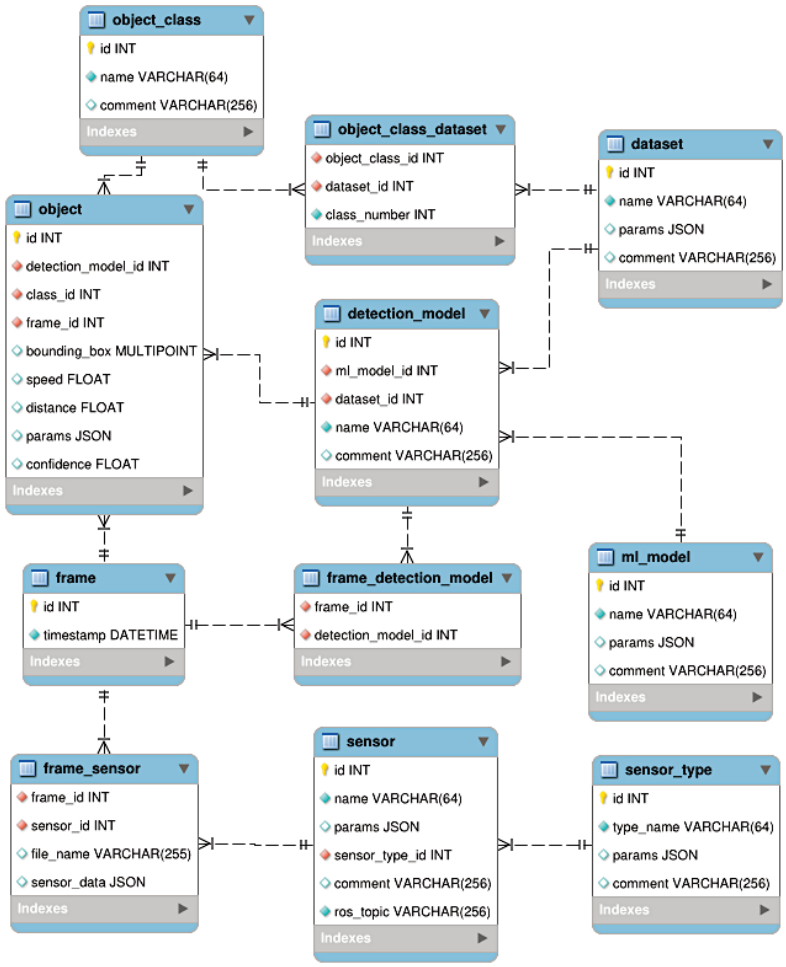

Fig. 6. System logging SQL database schema.

level, covering variables such as velocity, steering angle, door and light status, as well as brake and emergency brake status. As depicted in Fig. 8, the blue line represents GNSS (global navigation satellite system) data retrieved from the database, illustrating the trajectory path followed during the experiments on the designated route.

During our experimental evaluation of the master controller, we aimed to thoroughly assess its performance within the context of the autonomous shuttle. To do so, we conducted a comparative analysis using two different software systems: the original software, which came with the shuttle, and a custom software solution specifically designed for this study. These tests were carried out along a defined section of the shuttle's route, and throughout the experiment, we diligently recorded the steering data, as exemplified in Fig. 7.

The results, as depicted in Fig. 7, tell an intriguing story. They reveal that the steering angle achieved with our custom master controller consistently outperforms the steering provided by the original software. This enhanced performance is characterized by a smoother trajectory, implying greater precision and control over the shuttle's movements.

In conclusion, our evaluation strongly suggests that the custom master controller has the potential to sig-

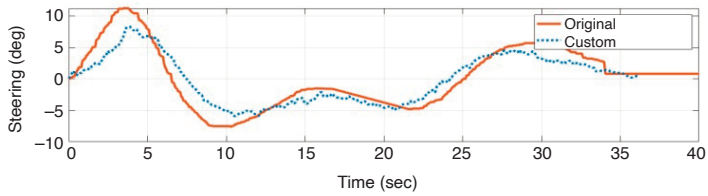

Fig. 7. Steering angle of the original software and with custom controller.

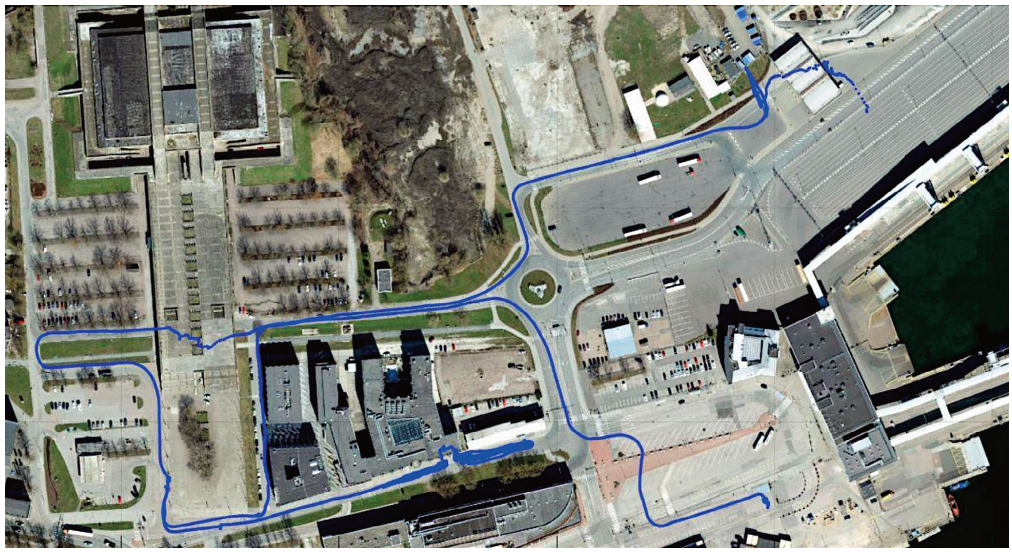

Fig. 8. Designated route that extended for a distance of 1.1 kilometers recorded in the database.

nificantly enhance the autonomous shuttle's steering performance compared to its original software counterpart. This finding highlights the importance of software optimization in achieving smoother and more reliable autonomous vehicle operations, which ultimately contribute to the advancement of autonomous transportation technologies. Further research and testing could provide valuable insights for refining and fine-tuning the custom software for even greater improvements in steering and overall autonomous vehicle performance.

4. RESULTS AND DISCUSSION

The new control system successfully completed all initial tests, conducted meticulously to ensure the shuttle's safety. The testing process adhered to a well-structured plan. First, the electronic control modules were tested individually on a testing bench while closely monitoring their performance. The analyzed results proved beneficial in fine-tuning data and refining the modules' operations. Next, the ECU was mounted on the shuttle, and further tests were conducted while the vehicle was lifted from the ground. This step allowed for additional scrutiny to verify the system's functionality under practical conditions.

Lastly, the driving tests of the shuttle were carried out on an empty street to identify any critical bugs and enhance the software's performance. This real-world testing enabled the team to rectify any issues and make necessary improvements to ensure the system's optimal functioning.

During the testing and debugging phase, a safety-critical bug was detected and promptly addressed in the

master controller software. The issue arose from the incorrect processing of speed command packets, leading to a sudden application of the shuttle's brakes. This behavior posed a significant safety risk to the passengers. Fortunately, the bug was swiftly rectified, ensuring that such abrupt braking incidents no longer occur, thereby enhancing the overall safety and reliability of the shuttle's control system.

To register the autonomous shuttle as a legal vehicle in Estonia, specific tests are mandated, which include verifying the reliability of the shuttle's control system by temporarily disconnecting certain system components. These requirements were taken into account during the development of the updated safety concept. Each module within the system serves a safety-related function.

For instance, the AI computer plays a crucial role in processing lidar and camera data, allowing it to execute smooth braking maneuvers when deemed safe and with sufficient distance. The master controller, in this safety concept, primarily acts as a gateway. It possesses the capability to deactivate all control packets transmitted across the three CAN networks if there is a loss of databus connection. This precautionary measure enables the low-level vehicle hardware to detect the issue and promptly execute emergency brakes, which involve shutting down the traction motor power, thereby ensuring a secure and controlled braking procedure.

In addition to initiating regular brakes and applying the handbrake, the AI computer is programmed to detect the absence of feedback packets. In such a situation, it immediately halts active driving actions to ensure safety and prevent any potential risks or hazards. The new master controller has three CAN connections and an ethernet connection linked to the main computer. Through testing, it was discovered that each connection plays a critical role in the safe operation of the shuttle. If the traction CAN1 is disconnected, the vehicle immediately engages in emergency braking and initiates a shutdown of the high-voltage system. Disconnection of the steering CAN2 causes the steering mechanism to cease functioning, leading to the loss of position feedback packets. Subsequently, the system shuts down as a precautionary measure.

When the body CAN3 is disconnected, both interior and exterior lights are deactivated, and the automatic doors cease to operate. However, a dedicated switch is available to cut off the power, allowing for manual opening of the doors. Lastly, if the ethernet connection between the master controller and the new control computer is severed, all control packets vanish from the three CAN networks. As a safety measure, the shuttle engages emergency brakes and comes to a complete stop. These safety protocols ensure that any potential disruptions or disconnections are promptly detected and managed, safeguarding the passengers and the vehicle's overall operation.

The functionality of the emergency stop buttons was individually tested using the control signal generated by the new master controller. The outcome demonstrated that the vehicle came to an immediate halt, as expected, aligning with the safety concept outlined previously. Nevertheless, further tests are necessary to assess the braking force and ensure it meets the stipulated requirements. These additional tests will provide a comprehensive evaluation of the shuttle's braking capabilities and validate its compliance with safety standards.

5. CONCLUSIONS

The process of transforming the existing self-driving shuttle into an open-source solution entails several crucial design steps. Firstly, the team maps the existing low-level architecture, identifying control computers and data connections to low-level vehicle systems, particularly using the controller area network (CAN) protocol.

Secondly, they log and analyze the CAN messages to extract essential data, creating a comprehensive understanding of the shuttle's functioning. Based on these data, a new solution is developed and implemented.

To ensure the safety and reliability of the new solution, rigorous experiments and tests are conducted on both low-level and high-level components. Critical tests involve deliberately disconnecting various system components to verify the system's resilience and ability to respond to faults appropriately.

In low-level tests, the focus is on assessing whether life-critical actuators precisely follow the intended movement patterns and if the selected action plan is triggered accurately when artificial faults are introduced.

The successful outcome of these experiments and tests culminated in creation of the new parallel-built shuttle, TalTech iseAuto 2.0. The insights and knowledge gained from this work contribute to the future efforts of making the shuttle street-legal in the shortest possible time, while ensuring its compliance with safety standards and regulations. The new shuttle is based on the Estonian first self-driving shuttle ISEAUTO, but has an updated low-level control system as well as a higher-level autonomous driving software stack – Autoware.Universe.

ACKNOWLEDGMENTS

This research has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under grant agreement No. 856602. The publication costs of this article were partially covered by the Estonian Academy of Sciences.

REFERENCES

1. Wetmore, J. Driving the dream. The history and motivations behind 60 years of automated highway systems in America. *Automot. Hist. Rev.*, 2003, **7**, 4–19.
2. Broggi, A., Bertozzi, M., Fascioli, A. and Conte, G. *Automatic Vehicle Guidance: the Experience of the ARGO Autonomous Vehicle.* World Scientific, 1999.
3. Buehler, M., Iagnemma, K. and Singh, S. (eds). *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic.* Springer, Berlin, Heidelberg, 2009.
4. Bertozzi, M., Bombini, L., Broggi, A., Buzzoni, M., Cardarelli, E., Cattani, S. et al. Viac: an out of ordinary experiment. In *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011*. IEEE, 2011, 175–180.
5. Broggi, A., Buzzoni, M., Debattisti, S., Grisleri, P., Laghi, M. C., Medici, P. et al. Extensive tests of autonomous driving technologies. *IEEE Trans. Intell. Transp. Syst.*, 2013, **14**(3), 1403–1415.
6. Zhang, J. and Singh, S. Laser–visual–inertial odometry and mapping with high robustness and low drift. *J. Field Robot.*, 2018, **35**(8), 1242–1264.
7. Gao, H., Cheng, B., Wang, J., Li, K., Zhao, J. and Li, D. Object classification using CNN-based fusion of vision and LiDAR in autonomous vehicle environment. *IEEE Trans. Industr. Inform.*, 2018, **14**(9), 4224–4231.
8. Rassõlkin, A., Vaimann, T., Kallaste, A. and Sell, R. Propulsion motor drive topology selection for further development of ISEAUTO self-driving car. In *Proceedings of the 2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), Riga, Latvia, 12–13 November 2018.* IEEE, 2018, 1–5.
9. Pikner, H., Sell, R., Majak, J. and Karjust, K. Safety system assessment case study of automated vehicle shuttle. *Electronics*, 2022, **11**(7). https://doi.org/10.3390/electronics 11071162, https://www.mdpi.com/2079-9292/11/7/1162
10. Wei, J., Snider, J. M., Kim, J., Dolan, J. M., Rajkumar, R. and Litkouhi, B. Towards a viable autonomous driving research platform. In *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 23–26 June 2013*. IEEE, 2013, 763–770.
11. Pikner, H. and Karjust, K. Multi-layer cyber-physical low-level control solution for mobile robots. *IOP Conf. Ser.: Mater. Sci. Eng.*, 2021, **1140**, 012048.
12. Bellone, M., Ismailogullari, A., Müür, J., Nissin, O., Sell, R. and Soe, R.-M. Autonomous driving in the real-world: the weather challenge in the Sohjoa Baltic project. In *Towards Connected and Autonomous Vehicle Highways* (Hamid, U. Z. A. and Al-Turjman, F., eds). Springer, Cham, 2021, 229–255.
13. Sell, R., Soe, R.-M., Wang, R. and Rassõlkin, A. Autonomous vehicle shuttle in smart city testbed. In *Intelligent System Solutions for Auto Mobility and Beyond. AMAA 2020* (Zachäus, C. and Meyer, G., eds). Springer, Cham, 2021, 143–157.
14. Pikner, H., Sell, R., Karjust, K., Malayjerdi, E. and Velsker, T. Cyber-physical control system for autonomous logistic robot. In *Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 25–29 April 2021*. IEEE, 2021, 699–704.
15. Rehrl, K. and Zankl, C. Digibus©: results from the first self-driving shuttle trial on a public road in Austria. *Eur. Transp. Res. Rev.*, 2018, **10**(2), 1–11.
16. Texas Instruments. Introduction to the controller area network (CAN). *Application Report SLOA101B, 2002.* https://www.ti.com/lit/an/sloa101b/sloa101b.pdf
17. Chi, H., Liu, J., Xu, W., Peng, M. and deGoicoechea, J. Design hands-on lab exercises for cyber-physical systems security education. *CISSE*, 2022, **9**(1), 1–8.
18. Viloria, A., Acuña, G. C., Franco, D. J. A., Hernández-Palma, H., Fuentes, J. P. and Rambal, E. P. Integration of data mining techniques to PostgreSQL database manager system. *Procedia Comput. Sci.*, 2019, **155**, 575–580.

Tase 4 autonoomse sõiduki juhtsüsteemi ümberkujundamine vabavaralisele lahendusele

Heiko Pikner, Raivo Sell ja Ehsan Malayjerdi

Uurimistöö tutvustab uudset lähenemisviisi avatud lähtekoodiga madala taseme juhtsüsteemi ülekandmiseks ühelt erinevate parameetritega autonoomselt sõidukilt teisele. Töö jagunes mitmeks etapiks. Esiteks kaardistati olemasolev lahendus ja leiti andmesiinid. Andmesiinidel liikuvad paketid salvestati ja neist eraldati olulised juhtsignaalid. Peale seda oli võimalik need signaalid taasluua, kasutades vabavaralist lahendust. Töökindluse ja ohutuse hindamiseks korraldati mitu katset erinevate alamsüsteemide rikete simuleerimiseks ja tulemuste mõõtmiseks. Pilootprojekti raames testiti modifitseeritud autonoomset sõidukit Tallinna tänavatel. Uuringu tulemused näitasid, et madala taseme juhtsüsteemide ülekandmine erinevate autonoomsete sõidukite vahel on teostatav. Tulemusi kasutati TalTechi uue iseAuto v2.0 arenduseks, kus võeti arvesse eksperimendi tulemusi ja saadud kogemusi madala taseme süsteemide testimisel linnatänavatel.

Appendix 4

IV

H. Pikner, M. Malayjerdi, M. Bellone, B. Baykara, and R. Sell, “Autonomous driving validation and verification using digital twins,” in *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, pp. 204–211, INSTICC, SciTePress, 2024

Autonomous Driving Validation and Verification Using Digital Twins

Heiko Pikner[1][a], Mohsen Malayjerdi[1][b], Mauro Bellone[2][c], Barış Cem Baykara[1], and Raivo Sell[1][d]
[2]*FinEst Smart City Centre of Excellence, Tallinn University of Technology, Tallinn, 19086 Estonia*
[1]*Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Tallinn, 19086 Estonia*
{*heiko.pikner, mohsen.malayjerdi, mauro.bellone, baris.baykara, raivo.sell*}*@taltech.ee*

Keywords: Autonomous Vehicles, Validation and Verification, Modeling and Simulation, Artificial Intelligence, Open Source

Abstract: With the introduction of autonomous vehicles, there is an increasing requirement for reliable methods to validate and verify artificial intelligence components that are part of safety-critical systems. Validation and verification (V&V) in real-world physical environments is costly and unsafe. Thus, the focus is moving towards using simulation environments to perform the bulk of the V&V task through virtualization. However, the viability and usefulness of simulation is very dependent on its predictive value. This predictive value is a function of the modeling capabilities of the simulator and the ability to replicate real-world environments. This process is commonly known as building the digital twin. Digital twin construction is non-trivial because it inherently involves abstracting particular aspects from the multi-dimensional real world to build a virtual model that can have useful predictive properties in the context of the model-of-computation of the simulator. With a focus on the V&V task, this paper will review methodologies available today for the digital twinning process, and its connection to the validation and verification process with an assessment of strengths/weaknesses and opportunities for future research. Furthermore, a case study involving our automated driving platforms will be discussed to show the current capabilities of digital twins, connected to their physical counterparts and their operating environment.

1 INTRODUCTION

The Autonomous Vehicle (AV) industry aims to ensure system safety before mass deployment. Real-world testing would take decades to accumulate over tens of billion accident-free miles which alone is not a reliable safety indicator (Kalra and Paddock, 2016). Among all testing methods, high-detail simulations show better performance considering cost and time (Thorn et al., 2018) (Matute-Peaspan et al., 2020). Leveraging physics engines and digital twins of real-world environments can significantly reduce testing time and cost, and test any upcoming potential feature in varying operational design domains (ODDs), such as weather conditions or traffic patterns. While AI-based AV controllers are effective in real-world conditions, they may disregard physical rules, resulting in atypical decisions. As a result, the significance and complexity of validation and verification V&V of autonomous driving functionalities increases (Wachenfeld and Winner, 2016; Tao et al., 2019).

Verification and validation (V&V) is defined in the ISO-IEC-IEEE 24765 (ISO-IEC-IEEE, 2017), as the

> *"process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements (...), and the final system or component complies with specified requirements".*

It is clear from the definition in the standard that the V&V process is aimed to verify specific predefined requirements, typically described in a technical specification. However, the ISO also says that while the process of verification ensures that *the system has been built right*, the validation addresses the question of whether *the right system has been built* for the specific task.

In autonomous driving, V&V of systems with both deterministic and stochastic components poses a challenge. Deterministic systems have predictable behavior with known inputs and outputs, such as vehicle hardware and electronics. In contrast, stochas-

[a] https://orcid.org/0000-0002-5360-4321
[b] https://orcid.org/0000-0001-6976-2095
[c] https://orcid.org/0000-0003-3692-0688
[d] https://orcid.org/0000-0003-1409-0206

tic processes, like object detection, have probabilistic outputs. In consideration of these aspects, the V&V process has to be carried out at the elementary level, in which each component is validated individually, and at an integration level, in which the V&V process is carried out to all components working together.

From a V&V standpoint, validating a stochastic process means verifying its entire probability distribution. Take dice rolling as an example; you'd need to roll the dice thousands of times to ensure each face appears equally. However, for complex systems like AVs, there are countless scenarios, making it impractical to physically test all outcomes. This is where digital twinning technology shines, allowing the computation of thousands of scenarios to predict system behavior. The precision of the digital twin directly impacts V&V fidelity. This paper explores recent digital twinning techniques in AVs and their distinctions from our custom platform.

2 RELATED WORK

Any industrial product, including AVs, starts its embryonic life from a Computer Aided Design (CAD) model with the goal of representing the idea, and continues to the Computer Aided Engineering (CAE) process that aims to optimize and test initial functionalities. Such product eventually goes to the production stage in which Computer Aided Manufacturing (CAM) comes into the game optimizing the manufacturing process. The industrial world very often confuses such processes as the digital twining process that, instead, has a fundamental difference: it represents a product as built, operating in the real world, and receiving data from it. These three characteristics are intrinsic and fundamental to defining a digital twin resembling a real product in its operational environment. CAD models represent a model as it could be, whereas digital twins represent the model as it is.

Literature in the field often refers to digital twins as an asset that improves products along their life cycle (Löcklin et al., 2020) (Ashtari Talkhestani et al., 2019). From this point of view, it is clear that CAD-CAM models and digital twins are very different objects, but CAD models are elements of digital twins.

The definition of digital twins was introduced by NASA 2012 (Shafto et al., 2012) with the necessity of modeling as accurately as possible flight conditions for astronauts in space or other environments, then shifted to other domains including industrial engineering and robotics (Negri et al., 2017). NASA defines a digital twin (Shafto et al., 2012) as

> *"an integrated multiphysics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin"*.

While the initial NASA's definition includes all the main components of digital twins, it lacks generality and new functionalities. For this reason, the definition has been updated and generalized referring to a digital replica of a physical system able to mirror all its static and dynamic characteristics (Talkhestani et al., 2018). However, it is really when digital twins start receiving data from their physical counterparts that becomes powerful exploiting computational capabilities to predict failures and drive update strategies. One can also see the digital twin as the feedback loop of a physical system, receiving data and thus correcting possible unexpected outcomes.

In this approach, also AVs and their testing environments can be connected to their digital twins in the simulated space. Nowadays, a commercial car has an expected lifespan of about 10-15 years, these vehicles, autonomous or not, have already many software functionalities that could be improved and updated over time keeping the same hardware components. Digital twinning allows manufacturers to continuously simulate each vehicle's behavior and receive data from their physical counterparts to verify and validate products and components, detecting possible faults in advance and releasing a fix via software update. AV simulations, for example in CARLA (Dosovitskiy et al., 2017) and Autoware (Kato et al., 2018), mainly use the concept of the digital twin to validate and verify the safety and performance of those vehicles. Autoware is an open-source software project for autonomous driving, while CARLA focuses on game-engine based simulation and providing assets to build environments (urban details, road users, etc.).

The advancements in computer graphics have opened a plethora of techniques to efficiently and more easily represent 3D environments with physics simulations and realistic lighting. Two of the most popular tools to take advantage of these methods are Unity and Unreal Engine. Although they are designed as game development tools, they can also be utilized as simulators thanks to their ability to simulate physics. AWSIM (Autoware Foundation, 2022) and CARLA are simulators that were built on top of these game engines with a specific focus on automated driving. On the other side of the ocean, Baidu is also driving the sector with the Apollo [1] open-source simulation and verification platform focusing

[1] Apollo, 2022: https://github.com/ApolloAuto/apollo

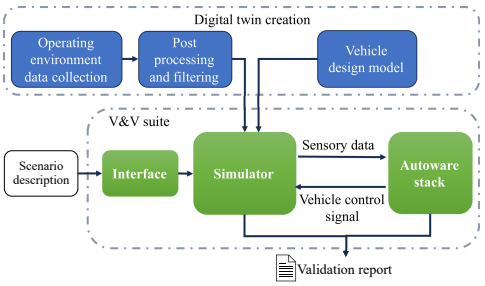

Figure 1: V&V suite workflow with digital twin, including environment and vehicles, as input to the V&V suite to provide a validation report.

on autonomous driving with several iterations of development. A testing case of this framework can be found in (Li et al., 2023).

An example of V&V platform, the PolyVerif[2] framework is very well detailed in (Razdan et al., 2023) and (Alnaser et al., 2019). Since the verification of physical objects is costly, not scalable, and has obvious safety concerns, this platform has been developed based on simulation methods. With any form of simulation, one must directly address the nature of model abstraction, and this is typically aligned with the operational abstraction of the Device Under Test (DUT), the AV stack in our case. Overlaid on the simulation framework is the design-of-experiment (DOE) unit consisting of a variety of scenarios (environment, dynamic actors) and some definition of correctness (pass/fail). The general workflow of a V&V platform is shown in Fig. 1. The framework defines an interface where the scenario definitions can be fed into the simulator. The digital twin including a vehicle under the test and its operating environment is a direct input to the simulator as an external loadable. It defines the environment domain and its properties such as buildings, vegetation, road definitions, etc. The simulator runs alongside the Autoware stack to aggregate the scenario definitions within that digital twin environment, and based on the outcome, it produces validation reports. The scenario description includes the specific use case of the vehicle in the environment to be validated.

3 DOE VALIDATION FLOWS

For a serious V&V task, one must build a Design of Experiment (DOE) infrastructure which is programmatic in nature. Key elements of the DOE flow mimic the process for any sophisticated large software project with elements. In summary, five concrete methods are provided to validate various parts of the AV stack. These flows provide researchers with a good initial understanding of the framework and encourage them to build derivatives that extend the paradigm in interesting directions.

In terms of modeling abstraction, the Autoware AV stack (Kato et al., 2018) (or any AV stack) is operating in a conventional Newtonian physics universe. To be useful, any simulation environment must model key concepts such as momentum, graphic processing, sound dynamics, and more. These concepts can be modeled at various levels of fidelity with a trade-off between accuracy and simulation performance (Malayjerdi et al., 2023b). At a component level, the internal useful abstractions of the major pieces of the Autoware AV stack are:

- **Detection**: This stage accepts sensor inputs, and the outputs are abstract objects in 3-D space. Thus, it is possible to test Detection functionality independent of the rest of the stack in simulation under a variety of conditions.
- **Control**: The control stage accepts data from a simulated CAN bus, the mission planning, and the detection stage to verify any risky maneuver that might generate impacts of discomfort.
- **Localization**: Localization takes sensory input (GPS, IMU) as well as the results of Detection to generate an abstract positioning of the DUT in a global map. Models of noise can be introduced to test the robustness of the localization engines.
- **Mission Planning**: Path Planning consumes abstract objects from detection and Localization to build an actuation function and a predicted future path. Again, simulation data can be used to independently test the path planning function.
- **Low level:** This stage has the goal of testing ECU-level functionalities that might be safety critical for autonomous vehicles. Low-level hardware will be simulated or implemented in the validation platform to ensure that they behave safely and perform correctly.

3.1 Detection Validation

The V&V framework constructs detection validation by introducing stubs in the simulator with the goal of capturing errors between the ground truth data and Autoware stack detection log. This data logging is done on a per-frame basis, and the complete dataset

[2]The Source code repository of polyVerif is available online and maintained at https://github.com/MaheshM99/PolyVerif

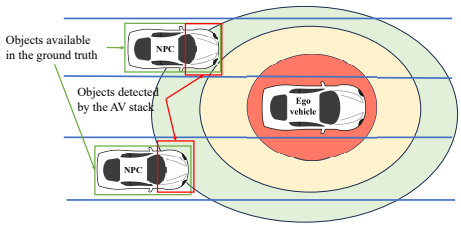

Figure 2: Detection validation example. The Ground truth of the detectable vehicle is indicated using green boxes while the detection is marked using red boxes. The success/failure ranges are indicated using circles around the ego vehicle.

is recorded in separate files for each of the test cases executed. Further, the framework automatically generates a figure of merit for the AV detection module performance. While generating results for object detection, below details can be considered (but not limited to):

- Frame by frame validation
- Report on objects detected by AV stack success and failure per object per frame.
- Distance based accuracy report generation, as lesser distant objects are important for control. E.g. Detection success/failure rate in the range 0-10 meters, 10-20 meters, etc.

Figure 2 explains about object comparison. Green boxes are shown for objects captured by ground truth while Red boxes are shown for objects detected by AV stack. Threshold based rules are designed to compare the objects. It is expected to provide specific indicators of detectable vehicles in different ranges for safety and danger areas.

3.2 Control Validation

In Control Validation, the framework checks the impact of detection on the AV stacks control mechanism. This validation enables safety testing of controls like automatic braking mechanisms by computing response time and braking distance parameters. The objects ground truth is captured from the simulator while perception results are captured from the AV stack with CANBUS data, to know the control instructions from the AV stack to the CARLA simulator. V&V algorithms are written to compare data and validate the AV stacks algorithms' efficiency and accuracy. Computed Information is as below:

- Time-To-Collision (TTC) Calculations as in Eq. 1
- Response time in the simulator after obstacle detection

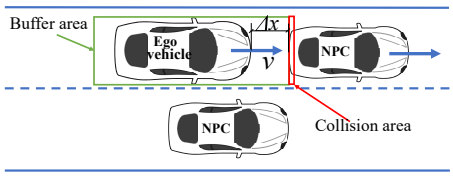

Figure 3: Time to Collisions Calculations and Collision Scenario.

- Response time in the AV stack after obstacle detection
- Delay in response due to perception/detection

$$TTC_i = \frac{\Delta x}{v_{rel}} \tag{1}$$

where v and x are the relative speed and distance between two vehicles. Figure 3 shows this concept in further detail, showing an ego-vehicle driving on the lane with other vehicles (NPCs), the time to collision is calculated using the simplest possible kinematic model using the relative speed between two vehicles.

Sufficient response time for AVs helps in the possibility of returning to a safer position without an imminent collision and by engaging the required braking force. Delay in response may cause collision and failure of AV systems.Computed parameters help in knowing the role of perception, their role in control initiation, and systems success/failure.

Current implementation rules are written considering highways and front/rear collisions from NPCs. Also, future plans are to consider all types of road infrastructures/junctions and static/pedestrian collisions from all directions.

3.3 Localization Validation

Vehicle localization failure leads to collision or accidents as shown in Fig. 4. Every AV stack has many inbuilt algorithms to ensure the accurate positioning of the vehicle. These algorithms use multiple sensors e.g. GPS/IMU for absolute position computation and other sensors like LIDAR/Camera/RADAR for relative position computation.

Under this validation, the V&V framework validates AV stacks localization algorithms and tests the capabilities of these algorithms in the case of GPS signal loss for a short period of time. This validation also helps in testing the localization mechanism by introducing different levels of noise into GPS/IMU sensor readings. The GPS and IMU noise can be modeled as per user requirements, and modified data can be published from the simulator to the AV stack to verify the behavior of the AV. The current validation

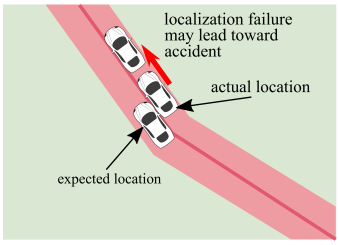

Figure 4: Localization validation, in some cases difference between expected location and actual location may lead to accidents.

method performs one-to-one mapping from the expected location vs. the actual location. Per frame, the vehicle position deviation value is computed and captured in the validation report. Later parameters like min/max/mean deviations are calculated from the same report.

In the validation procedure is also possible to modify the simulator to embed a mechanism to add noise in GPS/IMU data and provided the APIs to the end user. Through Python APIs, parameters can be passed to the simulator. The API internally models the noise and introduces the modified data in the simulation.

3.4 Mission Planning Validation

Each AV mission requires the capture of information from every possible sensor and the use of algorithms to move the vehicle safely to its destination based on that information. The success of the planned mission depends on the accuracy of these algorithms and the detection/perception of captured data by the sensors. Mission planning validation considers the start and goal position for the AV to navigate. Once these are set the AV generates a global trajectory based on the current location and the given destination. As shown in Fig. 5, the proposed platform validates that the trajectory is safely followed till the goal position. The validation report provides information on the trajectory following errors, collisions that have occurred, and whether the AV has reached its destination.

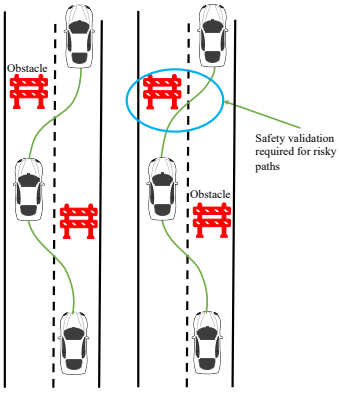

Figure 5: Trajectory validation example.

3.5 Low-level control validation

Low-level control systems involve electronic control modules (ECUs), data networks, and mechanical actuators. In modern vehicles, there may be over 80 ECUs in some cases, therefore, validating a low-level control system requires substantial labor and effort.

Classic solutions involve recording vehicle data bus traffic for post-processing or playback. Often, data packages in networks include checksum and other security elements. Manipulating pre-stored logs and altering specific signals is only possible by recalculating the checksum for each modified data package. These packages also contain counters, so simply deleting them would result in corrupted counter values.

Building a network of physical controllers can address the package generation challenge, but creating and validating such a network is labor-intensive. Additionally, testing vehicle subsystems in this simplified manner may yield undesirable results.

The next objective is to create a simulated low-level control system model inside a digital twin. One such tool is MATLAB and Simulink software. Simulink software allows the generation of a simulated low-level architecture for vehicles, including ECUs, and data buses as shown in Fig. 6. The autonomous software in ROS can generate navigation signals based on the virtual sensor data provided by the simulator. All navigation signals pass through the simulated low-level control system model and enter as

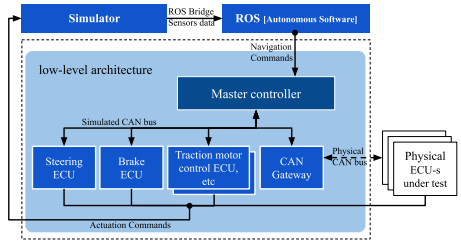

Figure 6: Low-level control system HIL simulation experimental structure. All of the vehicle's controllers are simulated, and while the simulation is running, traffic is generated on a simulated data network that can be used to test and develop physical controllers.

actuation commands into the simulator. So, for example, the consequence of turning off the steering system model would be that the control signal from the ROS computer would no longer turn the simulated vehicle wheels.

The gateway module facilitates the connection between physical and simulated data flow. This setup enables testing stand-alone ECUs or vehicle subsystems in a hardware-in-loop (HIL) environment when a vehicle self-drives inside a simulation and simultaneously generates all the traffic on the data network.

Such a test system facilitates easy and rapid validation for developing control modules and simulating system operation. Different designed situations and disturbances allow for performing various tests. It also provides testing scenarios that would be too hazardous to conduct in real traffic scenarios. Stability and durability can be evaluated by running tests for an extended period. Furthermore, the parameters of an actual vehicle can be compared against the model, and any discrepancies between the vehicle and the digital twin in response to the same input might indicate a possible fault.

4 CASE STUDY: TESTING AN AUTONOMOUS SHUTTLE

To decrease the entry barrier for researcher engagement, we provide a fully characterized AV-focused case study as a part of the V&V platform. We provide test cases by implementing an autonomous shuttle, iseAuto, in the simulated and real-world environment with the interesting premise that improvements in Autoware or V&V can be tested in cooperation with other research groups. The iseAuto is an autonomous shuttle of Tallinn University of Technologys (TalTech) AV research group operating on the campus for experimental and study purposes (Sell et al., 2022). The iseAuto projects objective was to build an open-source AV shuttle and establish a smart city testbed on the TalTech campus. It provides an ideal environment for researchers to make different types of projects on future urban mobility. The AV shuttle and its related operating environment are connected to its digital twin, enabling running all developments first in a simulation. The simulation environments, interfaces, and concepts are described in detail in (Sell et al., 2022) and (Malayjerdi et al., 2021).

The iseAuto high-level software architecture is based on the Robot Operating System (ROS). Perception, detection, and planning are performed by Autoware.ai driving (AD) stack. The vehicle is equipped with two Velodyne LiDARs at the top front (VLP-32) and top back (VLP-16) of the vehicle and two front sides Robosense RS-Bpearl to decrease the blind zone around the car. Sensors configuration and position are well detailed in (Malayjerdi et al., 2023a). Processes such as calibration, filtering, and concatenation were performed on the LiDARs point cloud to be optimized for perception purposes. The shuttle AD software is running on ROS, and its customized software architecture is described in (Sell et al., 2022).

4.1 Digital Twin of the iseAuto shuttle

The initial design model of the iseAuto shuttle was used and constantly updated, to deploy its digital twin, which is used as a DUT in any desired environment designed for testing and validation (Malayjerdi et al., 2021). The DUT digital twin contains the same sensor configuration as the real device as well as the 3D graphical model. The virtual environment also represents similar features to the actual test area; features such as urban details and vegetation are simulated within the environment. LGSVL (Rong et al., 2020) is deployed in the proposed platform as a vehicle simulator powered by the Unity game engine. This enables the creation of any desired virtual environment and the target vehicle to provide more flexibility and compliance in performing various tests. The simulator also benefits from a Python API toolkit to create different test scenarios based on pre-built features. It is also possible to import scenarios from a different platform (Malayjerdi et al., 2023a), e.g. SUMO (Behrisch et al., 2011).

To create a more complex test plan, multiple events can be included in one scenario. After running a simulation, the simulator provides virtual sensor inputs to the control algorithms provided by Autoware.ai. The raw data is received by the perception algorithms and then processed by various units. Finally, the software decides on the required actuation command and sends it back to the simulator environment. This communication is handled through a ROS bridge. Based on each study objective, various safety and performance KPIs are defined and the corresponding data is recorded during the runs. We then analyze and observe these criteria to find the vulnerabilities and corner cases where the DUT violated the metrics (Malayjerdi et al., 2023a; Roberts et al., 2023).

The data collection used in iseAuto is an end-to-end general-purpose AV data collection framework featuring algorithms for sensor calibration, information fusion, and data space to collect hours of robotics-related application that can generate data-driven models (Gu et al., 2023). The novelty of this

dataset collection framework is that it covers the aspects from sensor hardware to the developed dataset that can be easily accessed and used for other AD-related research. The framework has backend data processing algorithms to fuse the camera, LiDAR, and radar sensing modalities together. Detailed hardware specifications and the procedures to build the data acquisition and processing systems can be found in (Gu et al., 2023). Data collection and update is a crucial part of the digital twin creation process involving several resource demanding steps. However, it is worth mentioning that the digital map of an area can be reused in the digital twining process of several AVs or other types of robotic units as well.

The digital twin of the shuttle without its operational environment remains just a CAD model, to accurately represent the real environment in which the AV operates in a digital world (i.e. the workspace in which the AVs operate), aerial images of the environment must be collected. This can be done in various ways and with various sensors (LIDAR, RGB Camera, etc.).

Figure 7: Flight path example of a drone mission in the Taltech campus area.

In the case study proposed here, a drone with an RGB camera has been used in a grid flight path at a constant altitude to take sequential images of the environment (see Fig. 7). These images have been collected from three different angles to ensure the best possible coverage of the environments details. The images are georeferenced with a coordinate stamp by the drone acquisition system itself, the georeferencing process was supported by an RTK base station and ground markers to increase the accuracy of georeferencing. This makes it possible to photogrammetrically process them to obtain a point cloud of the environment. A small misalignment of the georeferenced images or unexpected glares on the lens of the camera could degrade the quality of the point cloud. Once the data has been collected, it goes through a photogrammetric alignment, point-cloud creation, and outlier removal process. This part is completely handled using commercially available software. This step makes it significantly easier to select and classify the point cloud and to clean it up from unwanted noise (see Fig. 8). The previously generated point clouds are then re-imported into Agisoft Metashape for classification and cleanup. It is also worth mentioning that after these processes are completed, one could easily generate buildings from this data directly in Metashape in any desired format.

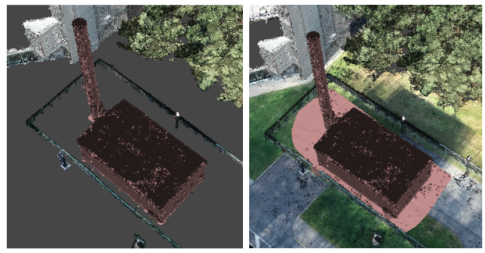

Figure 8: Comparison between point selection in segmented point-cloud and non-segmented point-cloud

4.2 TalTech iseAuto V&V

All of the steps required for the V&V process including the creation of the digital twin, scenario generation, and simulation are integrated into the simulation platform. As a primary step, an openDRIVE network map (xodr) of the target environment is needed. Figure 9 shows an example of a xodr map over the operating 3D virtual environment. In the next step, this map is used by the Scenic (Fremont et al., 2022) to generate distributed test cases all over the area. Scenic utilizes M-SDL, a human-readable, high-level scenario definition language, to describe scenarios. Several generated scenarios for a car parked in front of the AV are shown in Fig. 10. Scenic assists in the distribution of the target validation scenario over the entire operational area.

The generated scenarios are then simulated inside a high-fidelity simulator, which in this case is LGSVL. Fig. 11 displays 4 different passing scenarios generated by scenic and simulated in the simulator 3D environment. Each scenario was simulated in

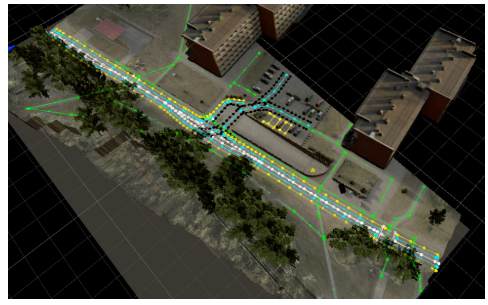

Figure 9: OpenDRIVE map over the 3D environment.

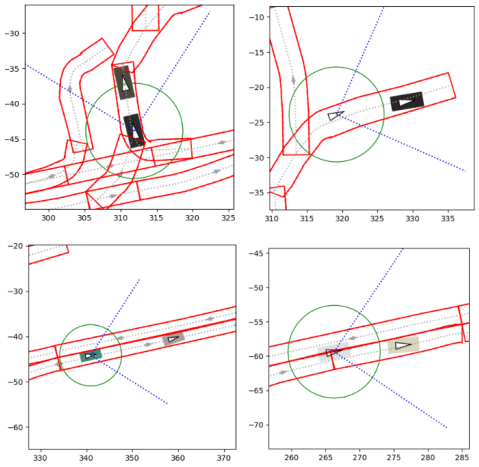

Figure 10: Scenic generates different scenarios over the xodr imported map

real-time, allowing the user to test the system's performance and safety. In this way, the user is able to identify the strengths and weaknesses of the system and make any necessary adjustments.

5 Research Questions and Enablement

V&V of AV systems is a very difficult problem and there is a need to build research frameworks that can accelerate the state-of-art. The proposed platform provides an open-source software framework providing all the key ingredients to experiment with novel V&V algorithms. Supporting this experimentation is a software flow including simulation, a default AV stack, a symbolic test generation environment, and a relatively automated digital twin creation flow. Furthermore, a case study including a live AV shuttle environment is provided. The environment is built with Autoware, so there is a potential to test improvements with the cooperation of the TalTech research team. Finally, five example designs of experiment flows are provided as instructive examples from which to build more sophisticated solutions. However, a current limitation is that the current examples take into account AI components only in the detection module. Many research questions arise from the use of AI, for instance, AI fundamentally builds a model from data with effectively an opaque lookup function for inference.

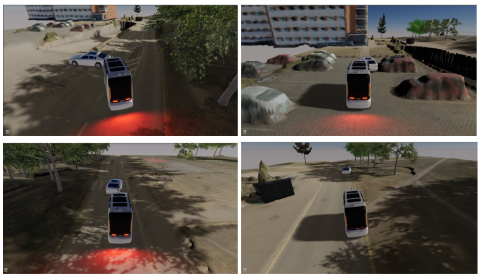

Figure 11: Scenarios generated by the scenic inside the LGSVL

This means data in the "algorithm" does not have a deterministic outcome in the operational domain as even a slight variation might generate unexpected outcomes. How can one validate the data projected through training for conformance to the appropriate Operational Design Domain (ODD) state space and its behavioral transformations? For AI, how does one capture "expectation" functions to determine correctness when there is a lack of a system design modeling methodology? Many AI applications use AI to "discover" the highest level system transformation. The answers to the above questions lead to the computational convergence questions.

An intuition would be to build a formalization of ODD state spaces and create a method for examining the data sets under that constraint. In the AI area, the only well-established method is cross-validation, involving the swap between several train-validation sub-datasets to confirm the model performances within a specific variance threshold. While cross-validation provides a measure of the knowledge abstraction capabilities of AI modules, it does not ensure that the final model is built in compliance with any well-established standard in the area.

> **Research Problem 1**: For AI training/inference, is there a more robust theory of convergence?

Current convergence criteria are based on loss-functions minimization and regularization methods. This means that the training stops when the minimization of the loss function does not improve anymore over time, and the best model is chosen over the best loss function value or using any early stopping criteria that measure the accuracy of the validation data. These criteria seem weak from a general knowledge abstraction point of view as validation and training datasets might be slightly different and the mathematical assurance of convergence exists only asymptotically (for the dataset size that goes to infinity).

> **Research Problem 2**: For AI V&V, is ther any theory of convergence?

The questions might seem similar at first glance, but they consider two different aspects, the training procedure of the model, and the validation procedure as the model is integrated into a product (e.g. a vehicle). Typically, V&V is exponential in terms of scenarios to consider, it is possible to use a number of techniques that employ abstraction to manage complexity but most of these techniques do not work with AI inference or work only on a limited subset of cases.

For AV in particular, further open research questions include:

- **Newtonian Physics**: Autonomy exists in the physical world. The physical world is governed by physics (Maxwell, Newton). This should be a great aid in helping set a governing framework for validation. How might one use the properties from physics to build a validation governor around AI-based autonomy systems?
- **Component Validation**: Each of the major steps in the AV pipeline (Detection, Perception, Location Services, Path Planning, etc) has its challenges. Can one build robust component-level validation for each of these?
- **Abstraction**: Complex problems are solved by the use of abstraction. Is it possible to leverage component validation such that deeper scenario validation can be done at a higher level of abstraction? If so, what are the abstractions of concern?

The field of AV and AV V&V is rich with open research problems. However, it is very difficult to make progress without a very large level of infrastructure. A cooperative open-source model is critical for progress, and the proposed platform is designed to help researchers quickly experiment with state-of-the-art ideas in this direction.

6 CONCLUSIONS

In conclusion, this paper underscores the pivotal role of digital twins in addressing validation and verification challenges associated with the principal components in AVs. Through a comprehensive review of current methodologies, this study elucidates the nuanced connection between the digital twining process and the imperative task of ensuring safety-critical systems reliability. The assessment of strengths, weaknesses, and opportunities for future research reveals the intricacies involved in constructing digital twins with high predictive value. The case study involving automated driving platforms serves as a tangible illustration of digital twin capabilities, showcasing their integration with their physical counterparts and operating environments. Recognizing the challenges inherent in digital twin construction, this conclusion advocates continued research efforts aimed at refining modeling capabilities, enhancing predictive value, and addressing identified limitations. Ultimately, the advancements in digital twin technology discussed in this paper bear significant implications for the broader development and deployment of autonomous vehicles. They offer promising avenues for bolstering their safety and reliability in real-world scenarios.

ACKNOWLEDGEMENTS

This work has been supported by the European Commission through the H2020 project Finest Twins (grant No. 856602).

REFERENCES

Alnaser, A. J., Akbas, M. I., Sargolzaei, A., and Razdan, R. (2019). Autonomous vehicles scenario testing framework and model of computation. *SAE International Journal of Connected and Automated Vehicles*, 2(4).

Ashtari Talkhestani, B., Jung, T., Lindemann, B., Sahlab, N., Jazdi, N., Schloegl, W., and Weyrich, M. (2019). An architecture of an intelligent digital twin in a cyber-physical production system. *at-Automatisierungstechnik*, 67(9):762–782.

Autoware Foundation (2022). TIER IV AWSIM. https://github.com/tier4/AWSIM.

Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

Fremont, D. J., Kim, E., Dreossi, T., Ghosh, S., Yue, X., Sangiovanni-Vincentelli, A. L., and Seshia, S. A. (2022). Scenic: A language for scenario specification and data generation. *Machine Learning*, pages 1–45.

Gu, J., Lind, A., Chhetri, T. R., Bellone, M., and Sell, R. (2023). End-to-end multimodal sensor dataset collection framework for autonomous vehicles.

ISO-IEC-IEEE (2017). Systems and software engineering -vocabulary. IEEE.

Kalra, N. and Paddock, S. M. (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193.

Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando,

T., Fujii, Y., and Azumi, T. (2018). Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE.

Li, H., Makkapati, V. P., Wan, L., Tomasch, E., Hoschopf, H., and Eichberger, A. (2023). Validation of automated driving function based on the apollo platform: A milestone for simulation with vehicle-in-the-loop testbed. *Vehicles*, 5(2):718–731.

Löcklin, A., Müller, M., Jung, T., Jazdi, N., White, D., and Weyrich, M. (2020). Digital twin for verification and validation of industrial automation systems–a survey. In *2020 25th IEEE international conference on emerging technologies and factory automation (ETFA)*, volume 1, pages 851–858. IEEE.

Malayjerdi, M., Baykara, B. C., Sell, R., and Malayjerdi, E. (2021). Autonomous vehicle safety evaluation through a high-fidelity simulation approach. *Proceedings Of The Estonian Academy Of Sciences*, 70(4):413–421.

Malayjerdi, M., Goss, Q. A., Akbaş, M. İ., Sell, R., and Bellone, M. (2023a). A two-layered approach for the validation of an operational autonomous shuttle. *IEEE Access*.

Malayjerdi, M., Kaljavesi, G., Diermeyer, F., and Sell, R. (2023b). Scenario-based validation for autonomous vehicles with different fidelity levels. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

Matute-Peaspan, J. A., Zubizarreta-Pico, A., and Diaz-Briceno, S. E. (2020). A vehicle simulation model and automated driving features validation for low-speed high automation applications. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7772–7781.

Negri, E., Fumagalli, L., and Macchi, M. (2017). A review of the roles of digital twin in cps-based production systems. *Procedia manufacturing*, 11:939–948.

Razdan, R., Akba, M. ., Sell, R., Bellone, M., Menase, M., and Malayjerdi, M. (2023). Polyverif: An open-source environment for autonomous vehicle validation and verification research acceleration. *IEEE Access*, 11:28343–28354.

Roberts, A., Malayjerdi, M., Bellone, M., Maennel, O., and Malayjerdi, E. (2023). Analysing adversarial threats to rule-based local-planning algorithms for autonomous driving. *Network and Distributed System Security (NDSS) Symposium*.

Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., et al. (2020). LGSVL simulator: A high fidelity simulator for autonomous driving. In *23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE.

Sell, R., Malayjerdi, E., Malayjerdi, M., and Baykara, B. C. (2022). Safety toolkit for automated vehicle shuttle-practical implementation of digital twin. In *2022 International Conference on Connected Vehicle and Expo (ICCVE)*, pages 1–6. IEEE.

Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., and Wang, L. (2012). Modeling, simulation, information technology & processing roadmap. *National Aeronautics and Space Administration*, 32(2012):1–38.

Talkhestani, B. A., Jazdi, N., Schlögl, W., and Weyrich, M. (2018). A concept in synchronization of virtual production system with real factory based on anchor-point method. *Procedia Cirp*, 67:13–17.

Tao, J., Li, Y., Wotawa, F., Felbinger, H., and Nica, M. (2019). On the industrial application of combinatorial testing for autonomous driving functions. In *2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 234–240. IEEE.

Thorn, E., Kimmel, S. C., Chaka, M., Hamilton, B. A., et al. (2018). A framework for automated driving system testable cases and scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety .

Wachenfeld, W. and Winner, H. (2016). The release of autonomous vehicles. *Autonomous Driving: Technical, Legal and Social Aspects*, pages 425–449.

Curriculum Vitae

1. Personal data

Name	Heiko Pikner
Date and place of birth	December 28, 1985, Tartu, Estonia
Nationality	Estonian

2. Contact information

E-mail heiko.pikner@taltech.ee

3. Education

2018–2024	Tallinn University of Technology, School of Engineering, Mechanical Engineering program, Production engineering and robotics, PhD studies
2010–2013	Tallinn University of Technology, Faculty of Mechanical Engineering, Mechatronics, MSc
2006–2010	Tallinn University of Technology, Faculty of Mechanical Engineering, Mechatronics, BSc

4. Language competence

Estonian	native
English	fluent

5. Professional employment

2021– ...	Tallinn University of Technology, School of Engineering, Department of Mechanical and Industrial Engineering, Junior Researcher
2018–2021	Tallinn University of Technology, School of Engineering, Department of Mechanical and Industrial Engineering, Engineer
2012–2018	OÜ ITT Group, Electronics, and robotics engineer

6. Field of research

- ETIS RESEARCH FIELD: 4. Natural Sciences and Engineering; 4.13. Mechanical Engineering, Automation Technology, and Manufacturing Technology
- CERCS RESEARCH FIELD: T125 Automation, robotics, control engineering
- SPECIFICATION: Autonomous systems and self-driving vehicles; Techiques and methods for the early design of mechatronic systems; Mobile robots

7. Honours and awards

- Securing the honorary II place in the Tallinn University of Technology Development Work of the Year 2020 is a testament to the significance of our research in the field of manufacturing and robotics, “Smart Manufacturing and Digital Twins: Self- Moving Robot Vehicle Boxbot in Production Logistics”, K. Karjust, R. Sell, T. Otto, M. Eerme, M. Pärn, V. Kuts, H. Pikner, T. Velsker, M. Kirs, J. Nõu, E. Malayjerdi, T. Raamets, A. Hermaste, K. Mahmood
- Recognition The Best Article In The Theme “Manufacturing” At The Conference MMM 2023 „Robot Bus Low-Level Control System Transformation To An Open-Source Solution", H. Pikner, J. Gu, R. Sell

8. Defended theses

- 2013, Universal data infrastructure for unmanned mobile robot, MSc, supervisors R. Sell and M. Leini, Tallinn University of Technology, Faculty of Mechanical Engineering, Mechatronics
- 2010, Steering module for the autonomous robot, supervisors R. Sell and M. Leini, Tallinn University of Technology, Faculty of Mechanical Engineering, Mechatronics

9. Scientific work
Papers

1. H. Pikner, R. Sell, K. Karjust, E. Malayjerdi, and T. Velsker, “Cyber-physical control system for autonomous logistic robot,” in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pp. 699–704, IEEE, Apr. 2021
2. H. Pikner and M. Malayjerdi, “Cyber-physical universal safety and crash detection system for autonomous robot,” *Robotic Systems and Applications*, vol. 1, no. 2, pp. 46–52, 2021
3. H. Pikner, R. Sell, J. Majak, and K. Karjust, “Safety system assessment case study of automated vehicle shuttle,” *Electronics*, vol. 11, p. 1162, Apr. 2022
4. H. Pikner, R. Sell, and E. Malayjerdi, “Level 4 commercial autonomous vehicle control system transition to an open-source solution,” *Proc. Eston. Acad. Sci.*, vol. 73, no. 2, pp. 124–133, 2024
5. H. Pikner, M. Malayjerdi, M. Bellone, B. Baykara, and R. Sell, “Autonomous driving validation and verification using digital twins,” in *Proceedings of the 10th Interna- tional Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, pp. 204–211, INSTICC, SciTePress, 2024
6. H. Pikner, R. Sell, and J. Gu, “Robot bus low-level control system transformation to an open-source solution,” *AIP Conference Proceedings*, vol. 2989, 01 2024
7. T. Pivoňka, R. Sell, H. Pikner, and L. Přeučil, “Fiducial Marker-Based monocular localization for autonomous docking,” *IFAC-PapersOnLine*, vol. 56, pp. 2957–2962, Jan. 2023

8. I. Astrov, A. Udal, H. Pikner, and E. 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)*, pp. 473–478, 2022
9. H. Pikner and K. Karjust, “Multi-layer cyber-physical low-level control solution for mobile robots,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1140, p. 012048, May 2021
10. H. Pikner and M. Malayjerdi, “Cyber-physical universal safety and crash detection system for autonomous robot,” in *MSM2021 ABSTRACT BOOK*, pp. 16–17, 2021
11. K. Karjust, J. Majak, H. Pikner, and R. Sell, “Multi-layer cyber-physical control method for mobile robot safety systems,” *Proceedings of the Estonian Academy of Sciences*, vol. 70, no. 4, pp. 383–391, 2021
12. H. Pikner, “Overview of cyber-physical control systems for self-driving vehicles,” in *Proceedings of the 19th International Symposium “Topical Problems in the Field of Electrical and Power Engineering” and “Doctoral School of Energy and Geotechnology III: School of Engineering, Tallinn University of Technology, Tartu 2020*, pp. 105–106, Tallinn University of Technology, 2020
13. M. Sarkans, R. Sell, K. Sonk, and H. Pikner, “Energy efficiency monitoring system for technology mapping driven by fof concept,” in *Proceedings of 9th International Conference of DAAAM Baltic Industrial Engineering, 24-26th April 2014*, Tallinn, Estonia, vol. 1, pp. 187–192, Tallinn University of Technology, 2014
14. M. Kremer, S. Seiler, M. Kuvaja, D. Ptasik, T. Lehtla, R. Ellermaa, L. Pähnapuu, T. Randmaa, H. Pikner, U. Heinaste, D. Bernhard, and R. Sell, “Õpisituatsioonid mehhatroonikas ja robootikas,” *Robolabor.ee kirjastus (ITT Group)*, 2013

Elulookirjeldus

1. Isikuandmed

Nimi	Heiko Pikner
Sünniaeg ja -koht	28.12.1985, Tartu, Eesti
Kodakondsus	Eesti

2. Kontaktandmed

E-post	heiko.pikner@taltech.ee

3. Haridus

2018–2024	Tallinna Tehnikaülikool, Inseneriteaduskond, Mehhanotehnika, doktoriõpe
2010–2013	Tallinna Tehnikaülikool, Mehhatroonika instituut, Mehhatroonika, MSc
2006–2010	Tallinna Tehnikaülikool, Mehhatroonika instituut, Mehhatroonika, BSc

4. Keelteoskus

eesti keel	emakeel
inglise keel	kõrgtase

5. Teenistuskäik

2021– ...	Tallinna Tehnikaülikool, Inseneriteaduskond, Mehaanika ja tööstustehnika instituut, doktorant-nooremteadur
2018–2021	Tallinna Tehnikaülikool, Inseneriteaduskond, Mehaanika ja tööstustehnika instituut, Insener
2012–2018	OÜ ITT Grupp, Elektroonika ja robootika insener

6. Teadustöö põhisuunad

- ETIS VALDKOND: 4. Loodusteadused ja tehnika; 4.13. Mehhanotehnika, automaatika, tööstustehnoloogia
- CERCS VALDKOND: T125 Automatiseerimine, robootika, juhtimistehnika
- TÄPSUSTUS: Autonoomsed süsteemid ja isejuhtivad sõidukid; Mehhatroonikasüsteemide varajase projekteerimise faasi metoodikad ja tehnikad; Mobiilsed robotid

7. Autasud

- Aukiri Tallinna Tehnikaülikooli aasta arendustöö 2020 II koht, „Nutikas tootmine ja digitaalsed kaksikud: Iseliikuv robotsõiduk Boxbot tootmise logistikas", K. Karjust, R. Sell, T. Otto, M. Eerme, M. Pärn, V. Kuts, H. Pikner, T. Velsker, M. Kirs, J. Nõu, E. Malayjerdi, T. Raamets, A. Hermaste, K. Mahmood
- Parim "Tootmine" teemalaline artikkel konverentsil MMM2023 „Robotbussi madala taseme juhtimissüsteemi ümberkujundamine avatud lähtekoodiga lahenduseks“, H. Pikner, J. Gu, R. Sell

8. Kaitstud lõputööd

- 2013, Universaalne andmevahetuse infrastruktuur mobiilsele mehitamata robotile, MSc, juhendajad Prof. R. Sell ja M. Leini, Tallinna Tehnikaülikool, Mehhatroonika instituut, Mehhatroonikasüsteemide õppetool
- 2010, Mobiilse roboti roolisüsteemi juhtmoodul, BSc, juhendajad Prof. R. Sell ja M. Leini, Tallinna Tehnikaülikool, Mehhatroonika instituut, Mehhatroonikasüsteemide õppetool

9. Teadustegevus
Teadusartiklite loetelu on toodud ingliskeelse elulookirjelduse juures.

ISSN 2585-6901 (PDF)
ISBN 978-9916-80-187-1 (PDF)