Development of a Digital Twin System for Optimizing Logistics Operations in a Distribution Center

Quartile 4 - 2022-2023

4AT100 Course
Final report

Group 2 - Team IntellION

Full Name	Student ID	Study
A. Kumar	1889486	Automotive Technology
A. Gopichand	1857304	Automotive Technology
S.K. Peram	1879634	Automotive Technology
M. den Besten	1494503	Automotive Technology
A. Gali	1660004	Automotive Technology
L.A.J.M. Thelen	1486543	Automotive Technology

Eindhoven, Saturday 28th December, 2024

Contents

1 Abstract

This report presents a digital twin framework for an autonomous driving truck system in a distribution center. With the rapid advancement of autonomous technologies, the transportation industry is exploring new possibilities for enhancing efficiency and safety in the distribution process. The digital twin approach offers a virtual representation of the real-world system, enabling accurate simulation and analysis.

The proposed digital twin integrates various components of the autonomous driving truck system, including perception, planning, control, and communication. It incorporates real-time data from sensors and cameras, allowing the virtual model to mimic the behavior and response of the physical trucks in the distribution center environment. By leveraging advanced algorithms and machine learning techniques, the digital twin can optimize route planning, cargo loading, and unloading operations.

The results of our experiments demonstrate that the digital twin framework significantly improves the operational efficiency of the autonomous driving truck system. It reduces delivery times and enhances overall safety. The digital twin also enables predictive maintenance, allowing proactive identification of potential failures and optimizing maintenance schedules.

Overall, the digital twin approach for an autonomous driving truck system in a distribution center presents immense potential for revolutionizing the logistics industry. It provides a cost-effective and risk-free environment for system optimization, ensuring smooth operations and increased productivity. The findings of this study contribute to the growing field of digital twin technology and pave the way for the future deployment of autonomous driving systems in distribution centers.

2 Introduction

The significance of fast and efficient transportation of goods is progressively growing alongside the expansion of the e-commerce industry. In order to meet this demand and maintain their competitive edge, distribution facilities must operate at their highest level of efficiency. Currently, there exists a control system that records the number of trucks entering or exiting the Distribution Centers (DC) as well as the idle time of these trucks. This existing control system falls short in determining the effective usage of trucks and docking stations. Additionally, a major concern is pertaining to the fatalities occurring within the DCs. The DC employees have to be on a constant lookout for their safety while working, as the truck drivers may not have a complete awareness of their surroundings. To address these issues, autonomous trucks and digital twin systems can be implemented within the DC premises. By effective utilization of these trucks, mishaps and accidents within the DC can be significantly reduced, along with an increase in overall efficiency.

This report presents the progress of the development of such a digital twin system, which mimics the behaviour of a real DC with the behaviour of autonomous trucks as closely as possible. The main component of this digital twin is the Intellion, responsible for controlling the functionalities inside the DC. This report outlines the project description, problem analysis using the 9-Box method and TRIZ methodology, and literature review. Furthermore, domain introductions give a general overview of the area in which the project operates, along with the potential difficulties and opportunities. Further, the system context, project context, and project boundaries are defined to recognise the scope of the project. The SEDEF/SYSMOD approach, functional safety and risk analysis contribute to the system's reliability.

The system operating environment specifies the possible hardware and software requirements to carry out this project. To ensure that the potential risks are mitigated and handled efficiently, potential issues or drawbacks associated with the project are also identified. Finally, in order to fulfill the needs of stakeholders and provide value to end users, the services offered in this project are described.

3 Project Description

3.1 Literature Review

The Literature Survey provides a concise yet comprehensive insight of Distribution Centers, Autonomous Driving, and Liability and Safety. This section explains the role of distribution centers in supply chain management, terminologies of autonomous driving technology, and ethical and legal considerations linked to liability and safety, drawing from a variety of academic sources and industry publications.

3.1.1 Distribution Centers:

As part of this project, a literature review was conducted to examine the changes in the operation of Distribution Centers with respect to increasing levels of automation. The primary focus was on identifying the highest achievable level of automation to model a smart Distribution Center for the future.

Traditionally, Distribution Centers relied on manual processes, including docking, un-docking, truck driving, and sorting, all performed by the drivers. However, as the number of products passing through these centers increased, the need for a faster, more efficient, and technologically advanced management system became evident [21]. Such advancements can reduce labor costs, minimize errors caused by human intervention, and increase overall efficiency.

In the present day, Distribution Centers use using various technologies for tasks such as data entry, barcode scanning, racking and other rudimentary solutions, leading to partially automated warehouses. These Distribution Centers are less labour-intensive due to the use of sensors, forklifts, cameras, RFID, and other software [1].

The future focus of this project is on fully automated Distribution Centers. These DCs will be cost-effective, almost completely automated, highly efficient, and capable of functioning with minimal errors and accidents. This will be made possible with futuristic technological advancements like automated Warehouse Management Systems, Automated Guided Vehicles, Automated Picking tools, Self-driving trucks, a centralized server for inter-vehicle communication, and the Internet of Things (IoT) (one such DC is shown in Figure 1). Among these, the use of complex IoT technology, which involves a network of physical objects with technologies like sensors and software that can communicate with each other and other systems over the internet or other communication networks, has rapidly increased [2].

Figure 1: A snapshot of a highly automated Distribution Center

3.1.2 Autonomous Driving

Another important topic of interest is autonomous or self-driving trucks and other motor vehicles. The Society of Automotive Engineers (SAE) [3] has established six levels of driving automation, ranging from Level 0 (no automation) to Level 5 (full automation) as shown in Figure 2. Levels 0 to 2 include manually controlled motor vehicles or those equipped with partial driving automation technologies like Advanced Driver Assistant Systems (ADAS) and Cruise Control. Levels 3 to 5 automation levels range from conditional driving automation to full driving automation of motor vehicles, where no human attention is required for the vehicle to function. This means that the user can be completely out-of-loop [4].

Currently, Level 2 automation is the most common in road vehicles. However, Mercedes-Benz has received permission to launch the first-ever Level 3 autonomous driving system, DRIVE PILOT, in the US states of Nevada and California, along with their Level 4 autonomous parking technology INTELLIGENT PARK PILOT [5]. This level of automation gives the driver the ability to take over when necessary, which instantly sets it apart from its predecessor.

This categorization also applies to the automation of trucks, ranging from fully manual to fully automated trucks. The history of self-driving trucks has progressed significantly, from the development and testing of a fleet of autonomous trucks used in a Chilean mine during the early 2000s to the involvement of companies like Daimler, and Ford in the late 2010s. Ford unveiled an SAE Level 4 semi-concept heavy-duty truck in September 2018. Kodiak Robotics also won the contract in late 2022 to prototype autonomous software that can navigate complex, off-road terrain with various operating conditions [6]. These developments can further help in improving the self-driving truck industry innumerably in the near future.

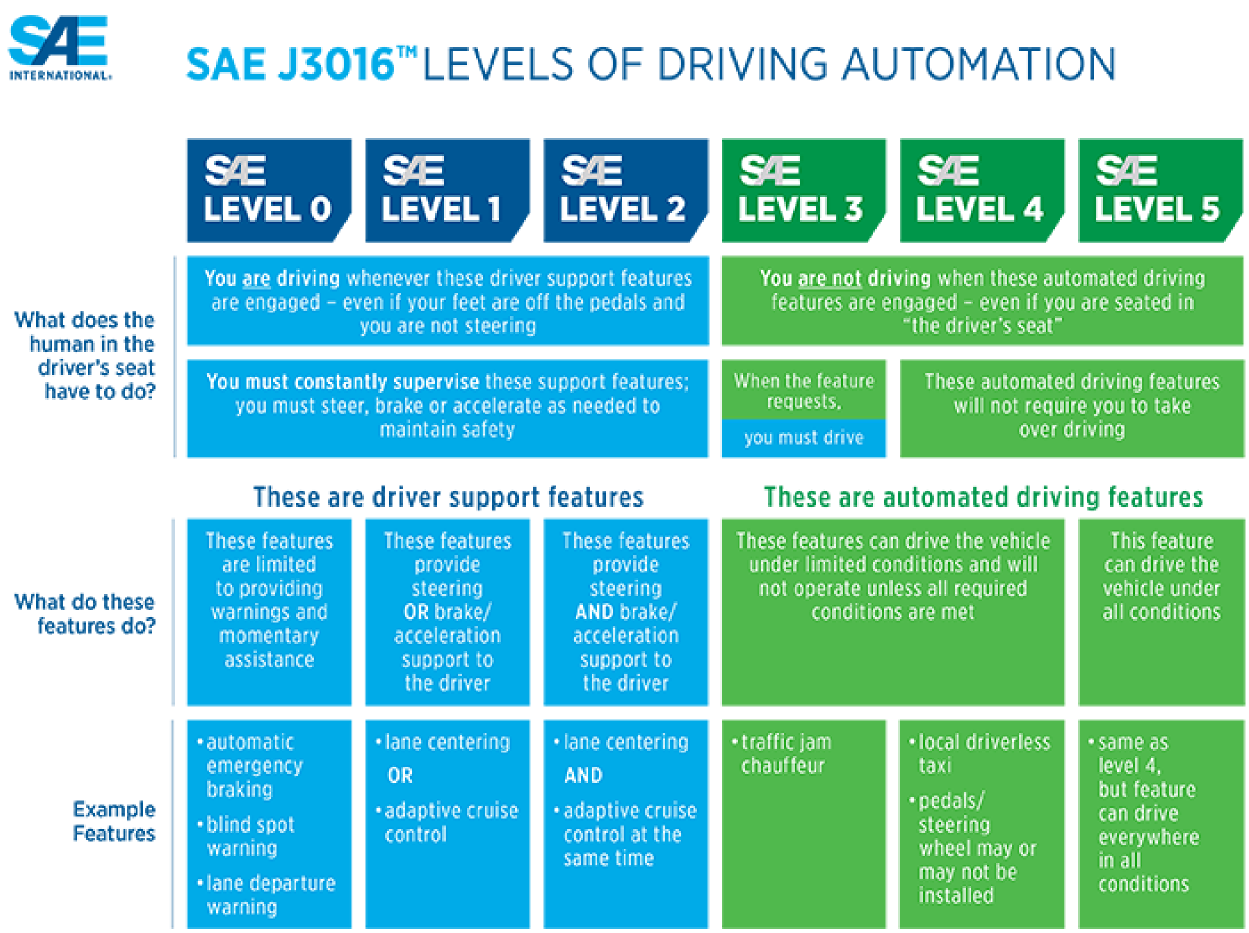


Figure 2: Chart showing SAE Levels of Driving Automation™

3.1.3 Liability and Safety

The use of self-driving trucks, particularly in distribution centers, can reduce logistics costs by up to 40% while improving the overall operating efficiency of the distribution center [7]. These trucks offer drivers the opportunity to rest while operating within the distribution center, leading to improved driver well-being. Furthermore, they utilize advanced technologies such as path planning, vehicle-to-vehicle communication, and obstacle detection, which contribute to an increased sense of security and a reduction in errors and accidents.

However, the introduction of this futuristic technology also presents complex challenges and legal issues. The current liability legislation is designed based on the existing levels of automation, holding drivers accountable for accidents. As automation levels increase and human involvement decreases, determining liability becomes uncertain. Another major concern is the lack of standardized regulations for testing autonomous vehicles, resulting in non-uniformity in different parts of the world [8].

While it is true that self-driving trucks may not be capable of handling all situations, overall, they are considered a safer option than manually operated trucks due to the aforementioned reasons. Concerns regarding the safety of autonomous trucks in worst-case scenarios will always exist, but it can be concluded that they outperform their manually driven counterparts.

3.2 Project Definition

This project aims to develop a digital twin system of an autonomous truck (Digital Twin and Autonomous Driving Trucks System (DTADTS))for manoeuvring in the Distribution Center comprising of Functional Safety, a Kinematic model of the truck, Positing and Sensing System with appropriate sensors. The project will follow the System Engineering approach based on Model Driven Methodology to develop, implement and test the correlation between Virtual Twin and Physical Twin.

3.3 Objective of the project

The objective of this project is to investigate the feasibility of implementing autonomous driving in a closed environment (Distribution Center) by designing a system which integrates the Ware House (Docking/Undocking), the Transport (Truck) and the Premises (the space between the docking/undocking and the outer wall). The goal of this project is to test the safety and efficiency of the digital twin of autonomous driving vehicles in a closed environment and confined path.

The features of the project are to address the technical and security issues that may arise while integrating autonomous driving technology into a Distribution Center (DC). The project investigates methods for precisely locating the vehicles using sensors such as cameras and LIDAR. In order to guarantee the safe operation of the systems even in the event of a problem, the project will also assure functional safety for both the software and hardware deployed in the vehicle and the system.

3.4 Scrum Framework

Scrum is a project management framework that is used for the development and sustenance of complex products. It follows an AGILE approach to tackling problems which emphasizes adaptability, collaboration, and customer satisfaction. The AGILE methodology utilizes a scientific approach to address the challenges posed by complex problems, replacing the conventional algorithmic programming approach with a more heuristic one. The traditional alternative to AGILE, known as the "Linear Waterfall model", relies heavily on detailed planning and extensive documentation. However, the flexible and dynamic AGILE framework has replaced it, helping teams to adapt to changing requirements and shifting priorities in response to changing market demands.

AGILE promotes self-learning and self-organization among team members as opposed to a lot of planning and documentation, empowering them to work independently and collaboratively to solve complex problems. Requirements and solutions evolve as a result of the collaborative effort of self-organizing and cross-functional teams. By reducing the amount of planning required, teams can focus on delivering value to customers in the form of working software. This approach helps in reducing the risk of delivering software that does not meet customer expectations by allowing teams to incorporate customer feedback into each sprint or iteration.

Figure 3 gives an understanding of the scrum life cycle. It commences with input from the stakeholders, followed by the presentation of ideas by the product owner culminating in the creation of the final product which is then reviewed in the sprint retrospective.

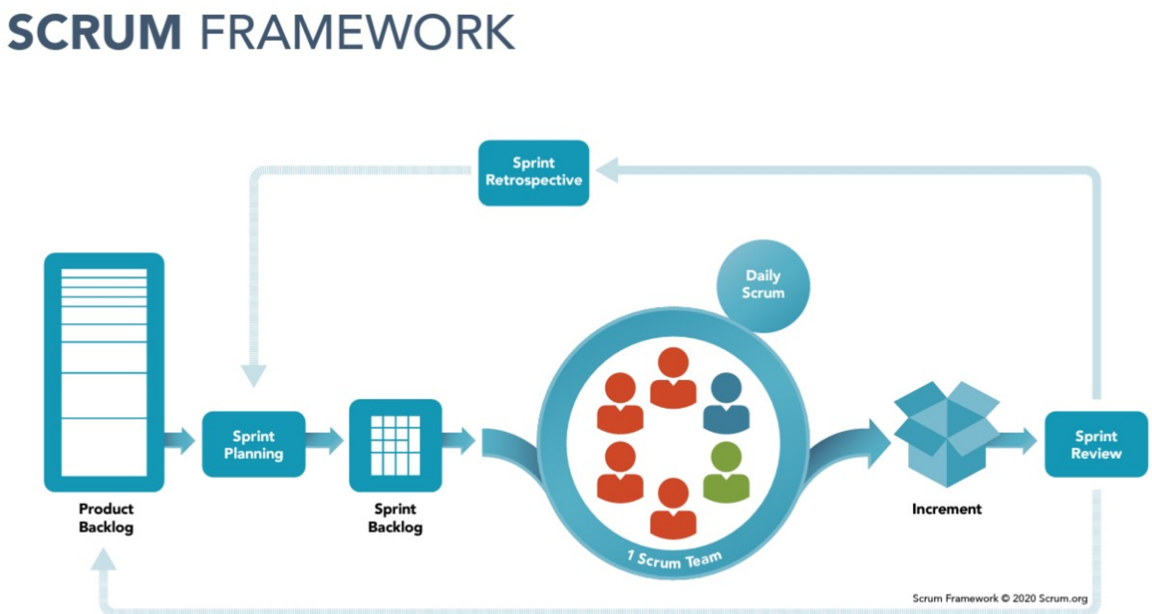


Figure 3: Scrum Framework

At the core of Scrum is the Product Backlog, which represents a comprehensive list of requirements for the product from start to finish. The Product Backlog is managed by the Product Owner, who is responsible for ensuring that it is prioritized, refined, and updated regularly. The Product Backlog is constantly evolving as it keeps changing throughout the product life cycle, based on feedback from stakeholders, market conditions, and emerging requirements. During Sprint Planning, the Development Team gets together to select a subset of items from the Product Backlog that they will work on during the Sprint. This subset of items is referred to as the Sprint Backlog, and it represents the list of tasks that the Development Team needs to work on during the Sprint. Throughout the Sprint, the Development Team works on the tasks identified in the Sprint Backlog. To facilitate collaboration and communication, the Development Team meets regularly for the Scrum Meeting, which is held once a week and is overseen by the Scrum master. This is followed by the sprint retrospective, which acts as a feedback loop during which the Development Team reflects on the Sprint and identifies areas for improvement. Rather than attempting to plan every aspect of a project upfront, AGILE encourages teams to focus on planning only what is necessary to get started, and then react to the feedback received during each sprint. This approach allows teams to continuously improve their work and respond quickly to changing customer needs.

3.5 TRIZ Methodology

The Theory of Inventive Problem Solving (TRIZ) is created to assist organizations in finding creative solutions to complicated issues. The foundation of TRIZ is the notion that there are patterns of innovation and invention that can be found and applied to various issues, spanning various industries and domains. [16] gives a roadmap to TRIZ.

The main aspects of the TRIZ methodology are as follows:

1. Systematic method: TRIZ offers a methodical approach to problem-solving that can assist organizations in coming up with more creative solutions. It provides a methodical approach for examining issues, spotting inconsistencies, and coming up with solutions.
2. Unique viewpoint: TRIZ offers a distinct viewpoint on problem-solving. It encourages people to investigate solutions that have been effective in other fields or sectors in addition to their own. This can aid in removing obstacles and promoting interdisciplinary thinking.
3. Efficiency: TRIZ offers an organized method to problem-solving, which can help organizations save time and resources. It can aid in rapidly and accurately determining the roots of issues and produce creative solutions that could not have been thought of otherwise.
4. Scalability: TRIZ can be used to solve a variety of issues, from minor technical issues to significant organizational difficulties. It is a flexible problem-solving methodology that may be applied in a variety of sectors and fields.
5. Enhancement of patentability: TRIZ can assist people and organizations in coming up with novel solutions that may qualify as patents. The process can be used to find novel and distinctive solutions that weren't previously thought of, which can result in the creation of priceless intellectual property.

The automobile industry places a high focus on safety, and TRIZ can assist in enhancing safety features in vehicles. Functional modelling is one of the TRIZ methods that can assist engineers to find the underlying causes of safety problems and come up with creative solutions to fix them. For instance, TRIZ can assist in the development of novel safety systems that employ sensors and artificial intelligence to find and address possible safety issues. Below are the usage of TRIZ with respect to the domains:

1. Autonomous Truck: TRIZ can be used to further the technology behind autonomous trucks. Navigating in challenging conditions, such as urban regions or construction zones, presents a difficulty for autonomous trucks. TRIZ can help to find creative solutions to this problem. The design of the truck's sensors, software, and control systems can also be optimized using TRIZ methods like creative principles and substance-field analysis.
2. Path Planning: A key component of autonomous driving is path planning. Using algorithms to optimize the path depending on several parameters, such as traffic volume, the state of the weather, and potential hazards towards the destination, TRIZ can help to generate creative ways for enhancing path planning algorithms. The best and most efficient path planning solutions can be found using TRIZ tools like function analysis and separation principles.
3. Automobile Human Factors: TRIZ can also be used to enhance the design's consideration of human considerations. This entails enhancing the comfort, safety, and convenience of the driver and passengers through the design of the vehicle's interior and appearance. The contradiction matrix and other TRIZ techniques, such as ideality, can be used to find creative answers to problems with competing needs, such as enhancing comfort while also lowering weight or cost.
4. powertrains: TRIZ can contribute to the creation of original ideas for enhancing the effectiveness and performance of vehicle powertrains. To find novel approaches to powertrain design, such as the use of materials or the optimization of the layout of the powertrain components, TRIZ methods like the creative principles and the substance-field analysis can be employed. TRIZ can be used to resolve needs that are in conflicts, such as enhancing performance while also lowering emissions or fuel usage.

3.5.1 Relation between TRIZ and the Automotive Field

To solve a variety of design and production issues, the automotive industry has used the TRIZ (Theory of Inventive Problem Solving) methodology. With new technologies and guidelines for energy economy, safety, and sustainability, the vehicle industry is always changing. TRIZ offers a framework for dealing with these problems in a methodical and effective way. To boost fuel efficiency or lower pollutants, for instance, TRIZ concepts can be used to optimize the design of engine parts like pistons and valves. By locating and removing bottlenecks, lowering waste, and enhancing quality control, TRIZ can also be used to enhance industrial processes.

Moreover, TRIZ can assist automakers in developing original responses to new trends like autonomous driving and electric automobiles. By utilizing TRIZ, automotive engineers can create innovative technologies and products that satisfy the demands of these developing markets by assessing existing solutions and spotting discrepancies. The placement and design of sensors and control systems in autonomous vehicles, for instance, can be optimized using TRIZ, as can the weight and range of electric vehicles. Additionally, this methodology provides a disciplined approach to innovation in the automotive sector, helping businesses to solve technical problems and gain an edge in a market that is changing quickly.

3.5.2 Relation between TRIZ and Distribution Centers

The current focus of the Distribution/Supply Chain industry is "Green logistics" [14]. Researchers have found multiple issues such as delays in the delivery of goods, and packaging waste within the logistics sector due to the increasing significance of sustainability and emission control. One of the problems is the "Just-in-Time" strategy which requires a high level of precision in logistics movement. The trucks have to be present at the docking/undocking station at the correct time. Another problem is the number of harmful gases that the drivers or the DC Employees inhale while being inside the Distribution Centers. The primary source of these "harmful gases" inside the Distribution Centers is the idle trucks. The above-mentioned issues have to be tackled to reach the desired goal of having an "efficient" Distribution Centers.

Figure 4 shows the parameters that will be considered for solving the problem using TRIZ [9]. The parameters are universal and the engineers are tasked with identifying the specific parameters influenced in specific domains. This process is a fundamental step towards finding solutions to the proposed problems [10]. In the subsequent sections, the domains are discussed with respect to the parameters and how these influence the design and development of the system.

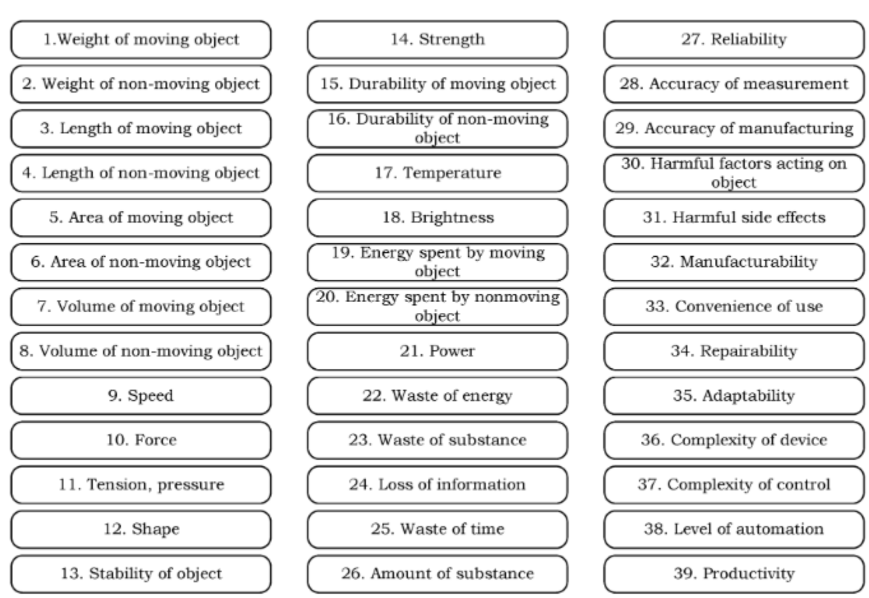


Figure 4: 39 TRIZ Engineering Parameters

3.5.3 9 Boxes Analysis

The 9 boxes method is an effective tool for idea generation and solving complex problems by exploring issues and their potential impacts by looking at the past, present and future. Additionally, this method provides various insights due to multiple levels of analysis, such as the super system, system and subsystem. Therefore, it proves to be valuable in this project by enhancing creativity, promoting collaboration through providing a structured approach, and ensuring team involvement.

Further, the aim of this project is to support the transition from present to future in the form of automating the docking and undocking processes of trucks. The 9 boxes diagram shown in Figure 5 provides a vision of future DCs and their functionalities on a supersystem, system, and subsystem level. Moreover, the diagram is also useful for identifying the problem's scope and the relations between higher and lower-level visions of a distribution center.

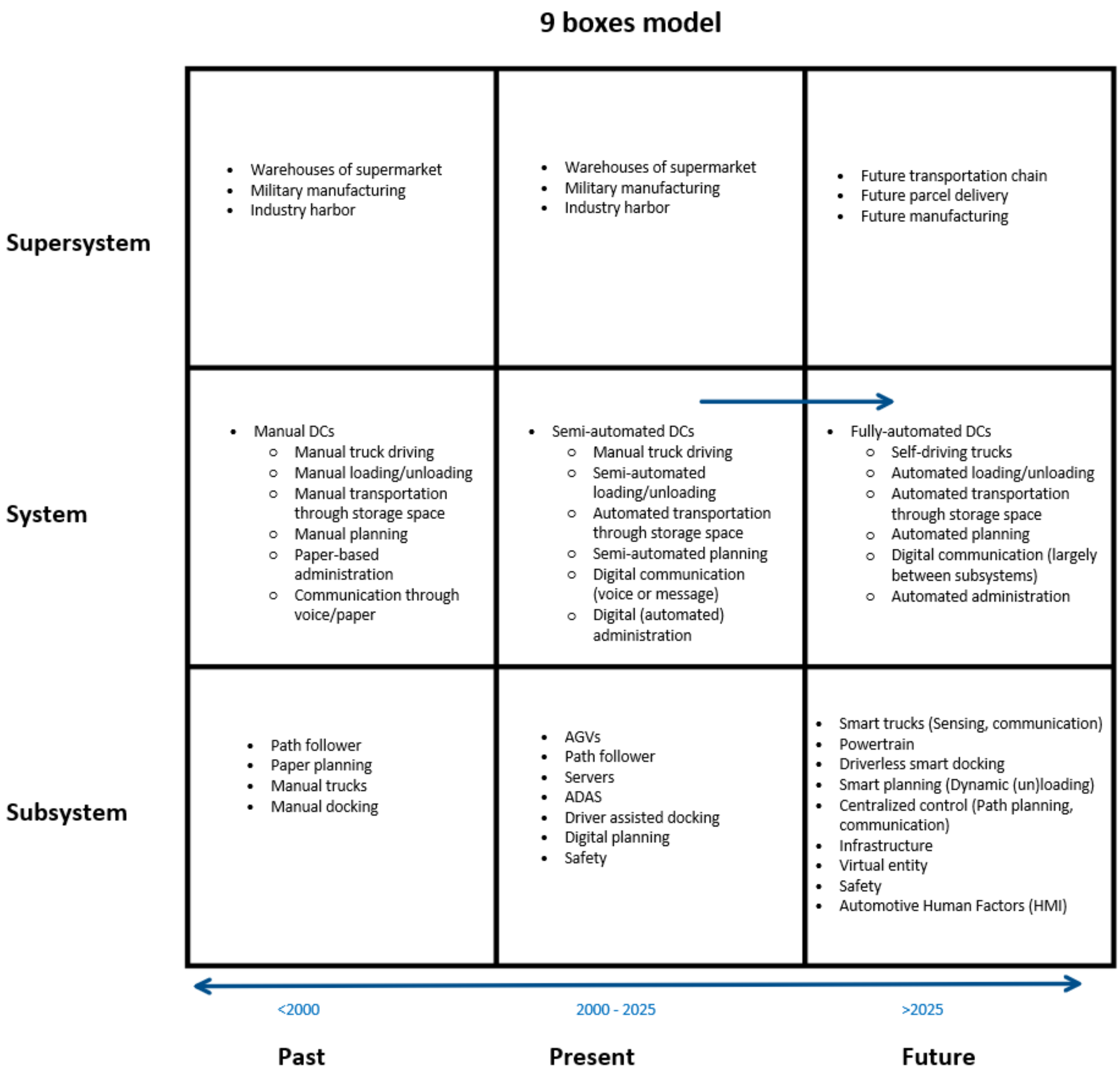


Figure 5: 9 boxes model

3.5.4 39 TRIZ engineering parameters

TRIZ defines 39 engineering parameters (Figure 4) to analyze and describe complex systems. The parameters cover various aspects of any engineering system. Therefore, engineers can analyze problems and generate inventive solutions to all included aspects. These aspects capture elements as dimensions, weight, shape, energy consumption and much more.
Accordingly, the TRIZ parameters can be applied to multiple domains of the Digital Twin system. As being a complex system, the Digital Twin is therefore distributed in 9 domains. As a result of that, the TRIZ parameters are applied per domain. Altogether, the domains are listed below, whereas the extensive motivations for choosing certain parameters for the domains are shown in Appendix A.

1. Virtual entity [12, 18, 27, 28, 29, 33, 34, 36, 39]
2. Automotive Human Factors (HMI) [17, 18, 19, 24, 27, 29, 34, 39]
3. Centralized Control (Path Planning, and Communication) [2, 3, 4, 6, 9, 15, 20, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
4. Smart Planning [16, 25, 27, 28, 29, 31, 33, 35, 36, 39]
5. Smart trucks (Sensing and Communication) [2, 3, 4, 5, 6, 9, 15, 18, 24, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39]
6. Driver-less smart docking [2, 3, 4, 5, 6, 9, 16, 27, 28, 29, 32, 33, 36, 38]
7. Powertrain [1, 9, 15, 19, 21, 22, 27, 29, 32]
8. Safety [1, 2, 3, 4, 5, 6, 9, 10, 15, 24, 29, 31, 35]
9. Infrastructure [2, 3, 5, 9, 18, 20, 29, 32]

In this section the methods which will be used to realize the aim of this project will be discussed. These methods include the way the system will be approached, the way the literature will be analyzed and how parts of the project will be simulated.

3.6 SYSMOD

The main methodology for approaching the project and reaching the aim of the project is SYSMOD. SYSMOD is the Systems Modeling Toolbox for SysML methodology, which is commonly used in the fields of systems engineering, software engineering, and project management to gain insights into complex systems and promote effective system design and management. Through this, a model-based system architecture can be constructed, starting with the stakeholders and their needs and desires, decomposed from the high- to the low-level design. After the system is designed on the component level, the design can be realised, integrated and tested, until the final system is realized. All of these steps in building the architecture and models of the system are iterative steps and can be continuously updated and changed throughout the scope of the project.

3.6.1 Role of SYSMOD in design process

3.6.2 System context

After the requirements are composed, the context of the system will be modelled. This context includes all the stakeholders, actors and other influences that are inputs or outputs of the system. The system itself will be modelled as a black box, and the goal of modelling the system context is to show the interactions with the external environment of the system.

3.6.3 Requirements

The first step of the SYSMOD methodology is the gathering of the requirements. This is done by first defining the stakeholders in the project and from their needs, wants and desires the requirements for the system can be composed. In this step, it is really important to thoroughly understand the project and the roles of the stakeholders, as the requirements will define how the system will function in the end.

3.6.4 Use Cases

Next, the use cases or the scenarios which the system will be put through will be modelled. In this way, all the functionality of the system will be designed and the interactions between the different use cases and actors of the system.

3.6.5 Realization

After the use cases are modelled, they can be realized. In this way, the behaviour of the system for the different functionalities and in different scenarios is modelled and can be understood. From these models, the system can be designed on the component level.

3.7 Project Boundaries, Priorities, and Considerations

1. Project Scope: This project will focus on developing an autonomous driving system specifically for trucks operating within a Distribution Center, with the goal of improving efficiency and safety in warehouse operations.
2. Limitations: The project will not address the technical and legal challenges associated with autonomous driving on public roads.
3. Confined Environment: The distribution center will be considered a confined area, where some of the restrictions related to autonomous driving do not apply, making it an ideal environment to develop and test the system.
4. Benefits: The autonomous driving system will be designed to enable drivers to rest while their vehicles operate autonomously, reduce damages caused by collisions and increase the efficiency of warehouse operations.
5. Power Train Technologies: The autonomous driving system will focus on trucks equipped with hybrid or battery-electric power trains, enabling silent and emission-free full-electric operation within Distribution Centers. However, the development of new power train technologies for trucks is beyond the scope of this project.
6. Infrastructure: The project will not address the development of new infrastructure or equipment in the Distribution Center.
7. Fully electric: The trucks used in the project will be actuated fully electrically.
8. Low speed and Precise Layout: The development of an autonomous driving system for trucks within the Distribution Center will take into consideration the low speed of the trucks, short stop distances, and precisely -known layout of the center to ensure safety and efficiency.
9. Dedicated sensors: The project will utilize sensor systems to accurately locate vehicles and facilitate the autonomous driving of trucks within the Distribution Center
10. Functional Safety: Ensuring functional safety in the Distribution Centers is essential for both the software and hardware deployed in the vehicle and will be a critical focus of the project.
11. Sustainability: The project will be developed with a focus on sustainability and reducing emissions within the Distribution Center, in line with the environmental goals.
12. Standards: The development of an autonomous driving system for trucks within the Distribution Center will adhere to ISO 26262 Functional Safety of Road Vehicles standards, ensuring that the system operates safely and reliably even in the event of malfunctions or failures.
13. Knowledge gaps: There might be limitations in the team's expertise regarding the integration of various components in the DTADTS. The project also relies on advanced sensing and perception technologies for localizing and controlling autonomous trucks. The team may need to acquire additional knowledge to bridge these gaps and ensure the successful implementation of the DTADTS.
14. Time Limitations: The project is expected to be completed within a limited time frame, which may impose constraints on the scope, depth, and quality of the work. These time limitations may require the

team to prioritize specific aspects of the project and make compromises in other areas. Additionally, the team may need to allocate resources efficiently and adhere to the Agile SCRUM project management framework to ensure the timely completion of the project.

15. Influence of Stakeholders: Balancing the needs of the stakeholders and ensuring that their concerns are addressed is crucial for the successful completion and adoption of the project.

3.8 Project Context

The development of a project, like any system, occurs within a specific environment or context. This context includes the scope, setting, environment, stakeholders, and other relevant factors that shape the project. In addition to a well-defined set of objectives, work scope, and budgets, understanding the project's context is essential. A lack of understanding of context is a major source of risk and may lead to project failure. The project context is created based on a diagram template that is centered around a 'black box' approach. This approach offers a simple and clear representation of the context using minimal technical information about the system.

Figure 6 shows the project context diagram. The central block represents the black box, surrounded by relevant subjects that collectively define the project's context. Following the black box approach, the model includes inputs, processes and outputs, although the specific processes may not be known. The model also includes acceptable and unacceptable inputs and outputs. Acceptable inputs are defined as inputs are those that contribute positively towards the project, such as a strong team, clear mission and objectives and a proper definition of resources and stakeholders. On the other hand, unacceptable inputs are those that have a negative impact, such as high threats, an incorrect understanding of the project context, a demotivated team, a high possibility of system failure, and low team working hours.

Often, an unacceptable input results in an unacceptable output. For example, a poorly understood project context may lead to project failure or an increased project time duration. Unacceptable outputs include missed deadlines, failure to meet requirements or agreements, safety issues, or even bankruptcy. On the contrary, acceptable outputs include a growing selection of opportunities, strong team morale, positive experiences, and successful implementation.

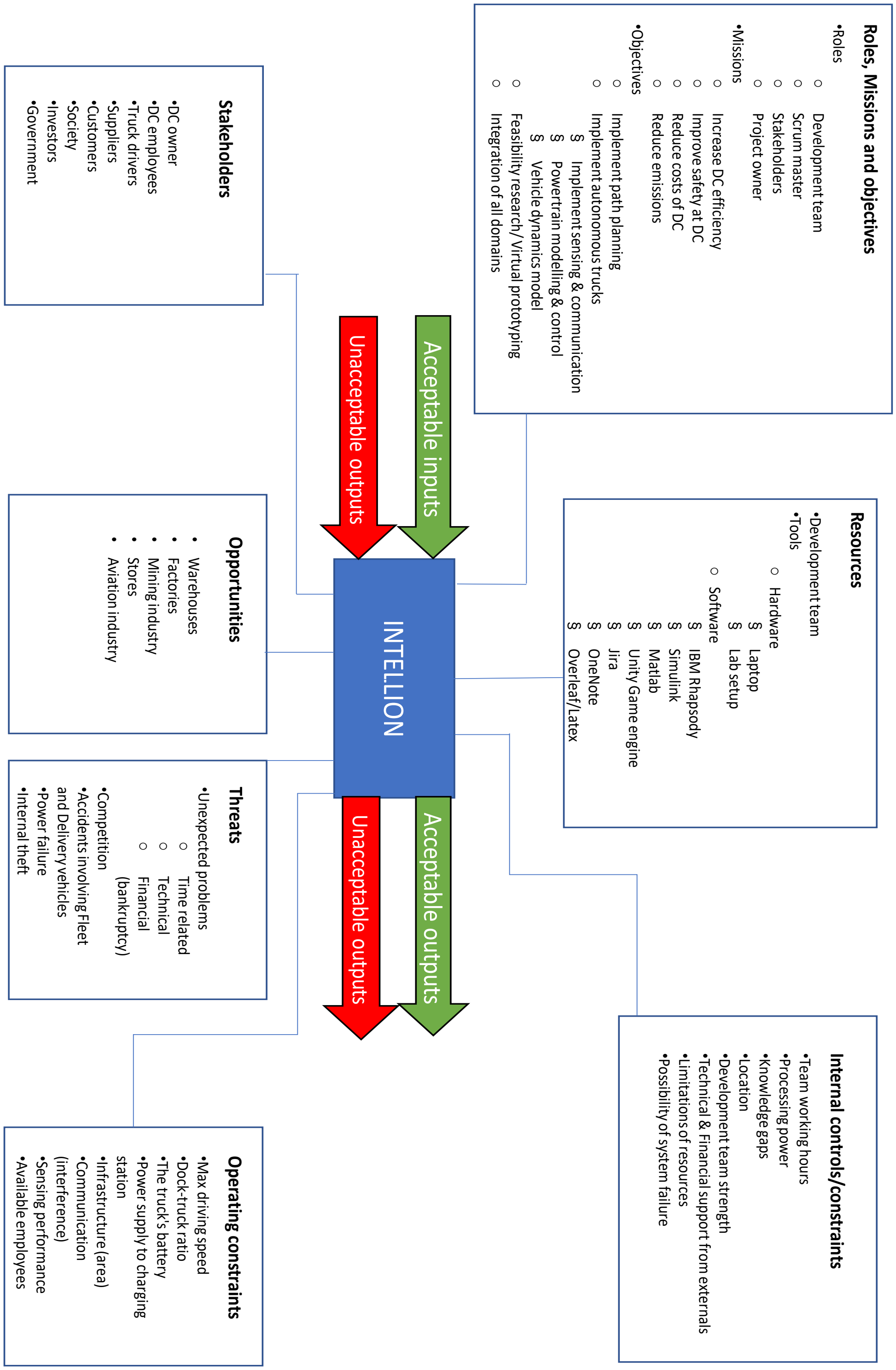


Figure 6: Project context diagram

Project Management is executed using SCRUM Process, which is an agile approach that aids teams in creating, delivering, and maintaining complex products. It is a gradual, iterative strategy that prioritizes adaptability, teamwork, and ongoing development. The SCRUM Process consists of the following phases:

1. Product Backlog: A prioritized list of the features, improvements, and bugs that need to be fixed in the product is known as a product backlog. The Product Owner is in charge of the Product Backlog, which is always changing.
2. Sprint: A time-boxed window (often 1-3 weeks) during which the team works to deliver a potentially marketable product increment.
3. Daily Scrum: A quick (often 15-minute) daily meeting when the team discusses the day's goals, obstacles, and progress.
4. Sprint Review: At the conclusion of each Sprint, the team gathers for a Sprint Review meeting where they present the work they have done and solicit input from stakeholders.
5. Sprint Retrospective: A team meeting in which the team reviews the prior sprint and identifies opportunities for improvement.

3.9 Cost Analysis

Below are ten key elements to take into account in a cost analysis for a digital twin autonomous driving truck system in a distribution center.

1. **Hardware and software costs:** The first costs to consider are the hardware and software costs. These are the physical resources needed to develop and maintain the system. Aspects that need to be considered are things such as sensors, cameras, processors, and communication equipment. For the development of the system, these are the Truck Lab, laptops, MATLAB, Simulink, IBM Rhapsody and Unity Game Engine. For maintaining the system the main thing that is needed is processing power in the form of any sort of computer and sensors, distributed over the complete distribution center to guide and sense the position of the trucks. Lastly, a user interface for both the employees and the truck drivers has to be made.
2. **Implementation costs:** For the implementation of the system the installation, configuration testing and integration have to be considered. As all of this has to be executed by experts, which both have knowledge from the developed system and the distribution center, where the system is being deployed, these expenses will be rather high. And as the system includes autonomous trucks there is quite some risk involved if a part of the system fails and therefore, all tests have to be executed thoroughly.
3. **Training costs:** The training costs for the system will be the costs for educating the employees of the distribution center and the truck drivers to work with and maintain the newly deployed system. These training costs will depend on the number of employees and therefore on the size of the distribution center. Next to the normal employees, system experts have to be trained, have some knowledge of the working of the system and can handle simple errors in the system.
4. **Maintenance and support costs:** The costs for maintaining and supporting the system include software updates, repairs, and replacement parts. As the system mainly consists of software and the hardware used is mainly for computing, the main costs for the maintenance and support will be in software updates, bug fixes and releases of new software versions and the minor part will be the replacement of sensors or parts of the user interface.
5. **Operational savings:** The operational savings of the system will be huge because if all the employees are trained to work with the system, the distribution center will work more efficiently, the throughput of the distribution center will increase and the distribution center can operate with fewer employees. Next to that, the truck drivers will be able to get rest while the truck is inside the distribution center, which increases their productivity. So the system will result in increased efficiency and reduced labour costs.
6. **Safety and liability costs:** Safety costs are associated with ensuring that autonomous driving trucks are safe to operate and that they do not pose a risk to workers or the environment. This includes

the development and implementation of safety protocols, regular maintenance and inspections, and training for workers on how to operate and interact with autonomous-driving trucks. Liability costs are associated with the potential risks and damages that may occur if autonomous driving trucks malfunction or are involved in accidents. This includes insurance costs, legal fees, and potential damages to property or persons.

7. **Regulatory compliance costs:** These costs are associated with ensuring that autonomous driving trucks comply with relevant laws, regulations, and standards. The Dutch government has established a legal framework for autonomous vehicles that are designed to ensure their safe operation. The Netherlands Vehicle Authority (RDW) is responsible for approving autonomous vehicles for use on public roads and sets technical requirements for their design and performance. Compliance costs can include the development and implementation of compliance programs, training for workers on compliance requirements, and the cost of hiring regulatory experts to ensure that autonomous driving trucks meet all relevant standards. It is important to carefully consider and address regulatory compliance costs to ensure that the implementation of Digital Twin Autonomous Driving Trucks in a distribution center is compliant with all relevant laws and regulations. Failure to comply with these requirements can result in legal and financial penalties, as well as reputational damage.

8. **Security costs:** These costs are associated with ensuring that the autonomous driving trucks and the systems they are connected to are secure from cyber-attacks and other security threats. The implementation of autonomous driving trucks requires a network infrastructure that connects the trucks to the central control system. This creates new opportunities for cyber-attacks and other security breaches. Malicious actors could attempt to gain unauthorized access to the network, disrupt the operation of the trucks, or steal sensitive data. To mitigate these risks, it is important to implement robust security measures, such as firewalls, encryption, and intrusion detection systems. Additionally, it is important to regularly update software and hardware to address new security threats as they emerge. Security costs can include the development and implementation of security protocols, hiring cyber security experts to monitor and secure the network, and the cost of purchasing and installing security hardware and software.

9. **System scalability:** : Scalability costs refer to the expenses involved in increasing the capacity of the system to handle more trucks, more routes, and more complex operations. The scalability of the system can be influenced by factors such as the size and complexity of the distribution center, the number of trucks needed to be integrated into the system, and the amount of data generated by the system. To ensure that the Digital Twin Autonomous Driving Trucks system can scale as needed, it is important to design the system to be modular and flexible, with the ability to add and remove trucks and routes as needed. This may require additional hardware, software, or network infrastructure, which can be costly. Additionally, as the system scales up, there may be a need to hire additional personnel, such as system administrators, truck operators, and maintenance personnel. This can also add to the overall scalability costs of the system.

10. **Return on Investment (ROI):** ROI costs include the upfront costs associated with purchasing and installing the system, as well as the ongoing costs associated with maintenance, upgrades, and personnel. However, the benefits of implementing a Digital Twin Autonomous Driving Trucks system can potentially outweigh the costs and generate a positive ROI. The benefits of a Digital Twin Autonomous Driving Trucks system include increased efficiency and productivity, reduced labour costs, improved safety, and more accurate tracking of inventory and shipments. By automating the process of moving goods within the distribution center, the system can help to reduce errors and delays, which can lead to increased customer satisfaction and reduced costs associated with product loss or damage. To accurately calculate the ROI for a Digital Twin Autonomous Driving Trucks system in a distribution center, it is important to consider both the costs and benefits of the system over its lifetime. This includes factors such as the initial cost of the system, ongoing maintenance and upgrade costs, labour savings, and improvements in efficiency and productivity.

Considering these 10 elements in a cost analysis will help ensure a comprehensive evaluation of the financial implications of implementing a digital twin autonomous driving truck system in a distribution center.

In Table 1 a rough estimation of the total cost, based on the 10 elements as described above, is given. From this table, it can be seen that the cost of the development of the digital twin autonomous driving system will be around €130,000.

Table 1: Cost overview of the development of the system

	Price	Amount	Total
Working hours			
Software development engineer	50	1680	84000
			0
hardware+software			0
Matlab+simulink	3500	1	3500
IBM Rhapsody	7000	1	7000
Unity game engine	400	2	800
Truck lab	0	1	0
Implementation			
Quality ensurance engineer	30	100	3000
Training			
IBM Rhapsody training	3200	2	6400
Maintenance and support			
Replacement parts (5 years)	2500	1	2500
Support costs (5 years)	5000	1	5000
Regulatory, Security and Compliance			
Product safety engineer	60	100	6000
Government license costs	12000	1	12000
Total cost			130200

3.10 Project Deliverables

The final project deliverables consist of several domains and solutions that need to be addressed. Firstly, in the domain of System Engineering and Real-Time systems, the objective is to design a Digital Twin system that can simulate the behaviour of the movement of a truck within a distribution center (DC). This requires taking into account the Safety Goals specified by ISO 26262 for the truck's functionality.

In terms of Vehicle Dynamics, the focus is on providing a path for the truck to follow depending on the availability of both the truck and an unoccupied docking/undocking station. The truck's dynamics play a crucial role in ensuring smooth manoeuvrability of the truck around the DC and avoiding Jack-Knifing.

Furthermore, a Powertrain solution needs to be provided, taking into consideration the demands of stakeholders and DC owners. Additionally, a Human Machine Interface (HMI) is to be developed to allow the driver to adjust the HVAC and lighting inside the truck. The HMI should also provide important information such as the location of the truck, autonomous level, and battery percentage, among other relevant details.

This project also includes the domain of Communication and Sensing, focussing on developing a Collision Avoidance System for the truck. This requires analysing the effect of various factors, such as cyber security, loss of communication network, and improved network functionality on the INTELLION System.

3.10.1 Elaboration of Deliverables

The following sections further provide a detailed explanation of the primary domains considered within the scope of this project:

1. **Powertrain** - Distribution facilities are increasingly using electric trucks and hybrid electric trucks because of their quiet operation and lack of emissions. They are especially well-suited for use in cities, where air pollution and noise are important issues. Due to their lower operating and maintenance

expenses, these trucks are also significantly less expensive than conventional diesel-powered vehicles. Also, fleet managers can optimize routes and cut down on energy use by using sophisticated monitoring systems that enable remote monitoring and control of the vehicle's performance. Although they cost more upfront than conventional diesel trucks, many businesses are starting to see the long-term cost savings and environmental advantages they provide, making them a wise investment.
Thus for our solution, we consider a Truck with either Electric Powertrain or a Hybrid Electric Powertrain to lower the overall DC emissions and provide a safe working environment for the DC Employees and Drivers. A final decision on the type of powertrain to be used will be made after further research.

2. **Path Planning -** A key component of a Distribution Center's effective operation can be path planning. It entails choosing the most efficient paths and orders for the Center's numerous processes, including product processing, storage, and transportation. Path planning is becoming more important as autonomous vehicles become more widely used since it forms the basis for the algorithms these vehicles use to navigate and operate in the Center. Path planning algorithms at a Distribution Center must take into account a number of elements, including the layout of the facility, the locations of loading docks and storage spaces, and the presence of other vehicles and pedestrians. Path planning must take the order of operations into account in order to ensure the best use of time and resources. A Distribution Center's overall efficiency can be greatly enhanced by an efficient path planning system, which can decrease the time and resources required for product handling and transportation while enhancing customer satisfaction.

3. **Vehicle Control** – With autonomous features, trucks that can drive themselves have the ability to completely change Distribution Centers by streamlining and improving the flow of commodities. Autonomous trucks may now operate in the constrained spaces of distribution facilities, where there is little to no pedestrian or general traffic, thanks to technological developments. These vehicles can easily move through the storage and shipping regions, precisely navigating through constrained lanes and loading docks thanks to specialized sensors like cameras and LIDAR (Light Detection and Ranging). The truck has the ability to make judgments based on its environment model, taking into account the positions of other vehicles and potential routes in order to optimize its movements. In addition to lowering fatigue and improving driver safety, autonomous vehicles enable truck drivers to take breaks while their vehicle is being loaded or unloaded.

4. **System Integration -** Effective system design and validation for Distribution Centers can be made possible by the integration of various technologies like SysML, Simulink, and Unity Game Engine. SySML can be applied to system-level modelling, requirement gathering, and system architecture definition. Simulink may be used to model and simulate the truck and autonomous driving system in great detail. For testing the autonomous driving algorithms and creating a realistic and immersive depiction of the Distribution Center environment, use the Unity Game Engine. It is feasible to achieve a complete and accurate picture of the system and its surroundings by integrating these technologies, which will improve system design and validation. Also, early in the design phase, this integration can assist in identifying and resolving any possible problems or conflicts, making the process more effective.

5. **Sensing and Communication -** Trucks have to incorporate sensing and communication technologies as part of autonomous driving. Camera and LiDAR sensors can facilitate collision avoidance. Hazards like crashes, spills, and other situations that potentially endanger people or property can be identified and dealt with. Vehicle to Infrastructure (V2I) network helps in the direct communication of Trucks with the DC Control System. Truck tracking is a crucial use of sensing and communication in Distribution Centers. Trucks can be equipped with sensors to track their location, speed, and direction of travel. Through this communication network, the trucks can produce alarms both locally and send signals to the DC Control System.

6. **Centralised Control and Smart Planning -** The assignment of trucks to the correct docking/undocking stations has to be done strategically. The main purpose of smart planning would be to not keep any truck idle, apart from when it is charging. This can also be solved by providing charging points at the docking stations as generally, a truck stays stationary at this point for more than one hour, which can be utilised for charging. The centralised control in this context means the DC Control

system which will compute the assignment of the trucks and path planning. Another control system could be inside the truck which includes Autonomous Driving and collision detection algorithms.

A concise understanding of the Project and its different modules can be seen in Figure 10.

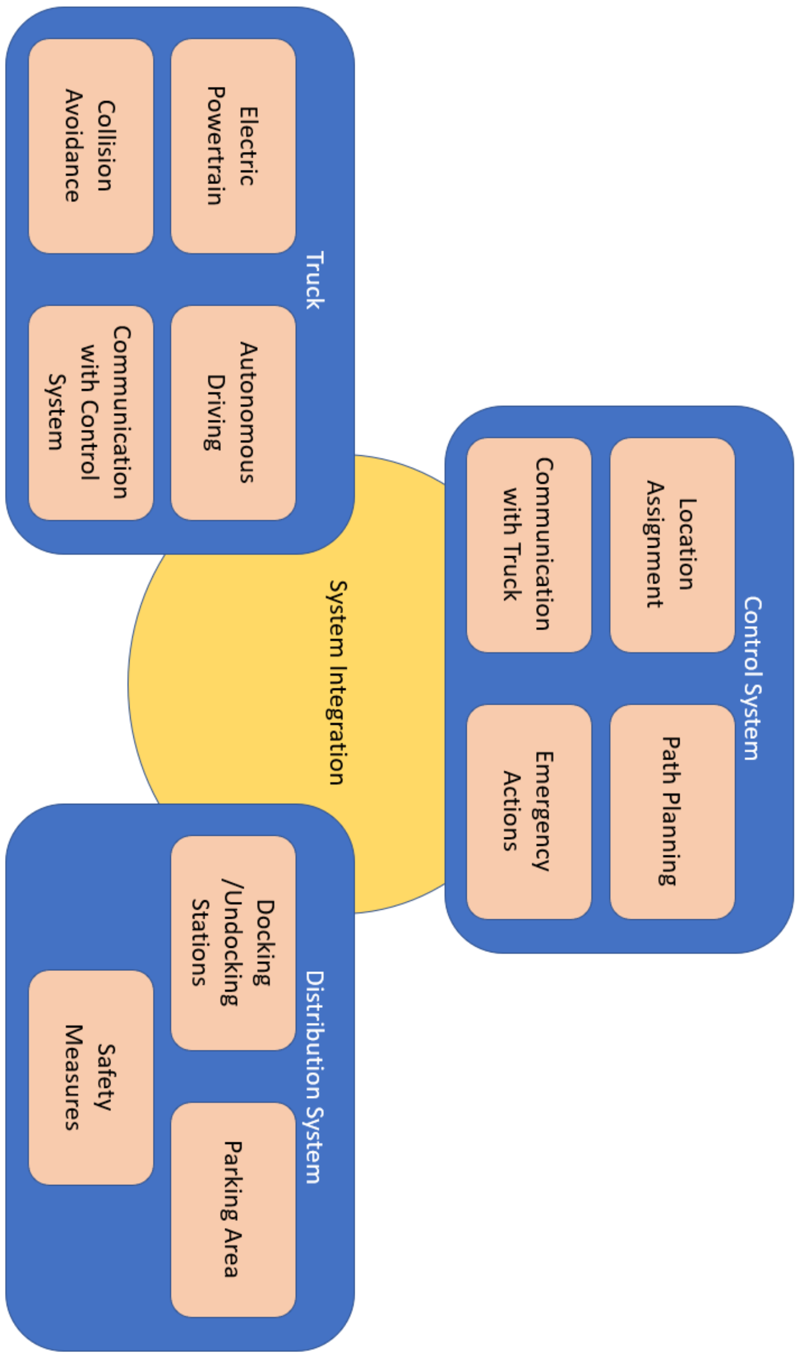


Figure 7: System Integration and its modules

3.11 Project plan

The project spans 5-6 months, starting from the basic concept project plan to implementing the digital twin model in an autonomous driving truck. The yellow vertical line in Figure 8 shows the current position in the project and the purple bars show the timeline of various sprints and deliverables throughout the course. Further, the green bar below the epics shows the completion of the tasks linked to them, while the grey bars show the incomplete and yet-to-start epics.

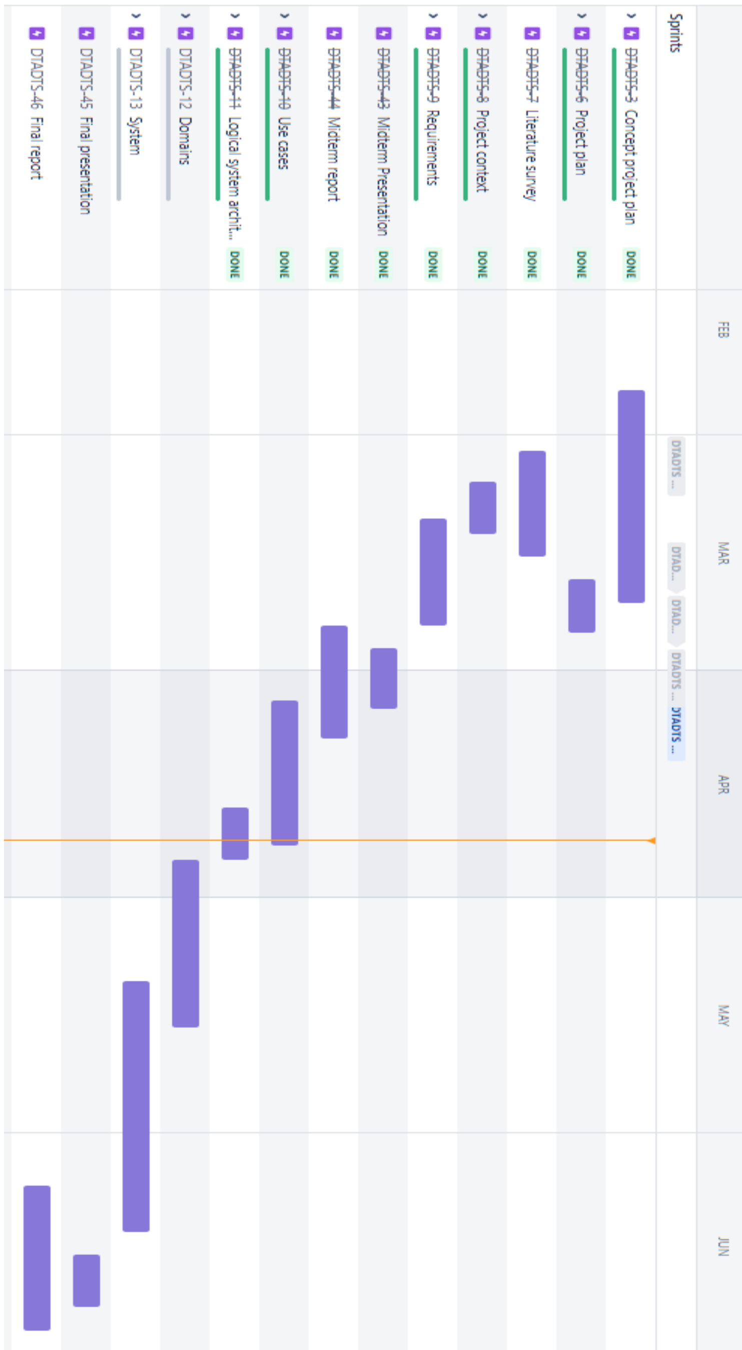


Figure 8: Project Timeline

4 System description

4.1 System Context

For any system to be developed, it is important to consider the system context. The system context is the specific system elements, boundaries, interconnections, interactions, and operational environment that define a system [19]. Understanding the system context is important as it determines the factors influencing the development of a system. In this project, the system context is modelled using a diagram. This diagram provides a concise overview of the system's scope, boundaries, and interfaces, and it is designed to be easily understood without requiring extensive technical knowledge.

As shown in Figure 9, the following stakeholders are considered to be relevant to this project:

1. DC owner: A person, group or company that owns the Distribution Center.
2. DC employees: People who are employed in the Distribution Center. For example, managers, supervisors, inventory clerks and material handlers.
3. Truck drivers: People that are responsible for operating the trucks towards and from the Distribution Center.
4. Suppliers: Companies that supply the Distribution Center.
5. Customers: Companies that request goods from the Distribution Center.
6. Society: The group of individuals that live together in a particular social system and that are associated with the Distribution Center.
7. Investors: Individuals or entities that utilize their capital for the Distribution Center in order to receive a return.
8. Government: The group of people with the authority to govern a country or state.

The external systems that are relevant are:

1. Environment: The surroundings or conditions in which the Distribution Center operates.
2. Autonomous trucks: The trucks that deliver and/or pick up goods.
3. Risk management: The management of risks related to the Distribution Center.
4. Standards: Levels of quality or achievement that are generally accepted or desired.

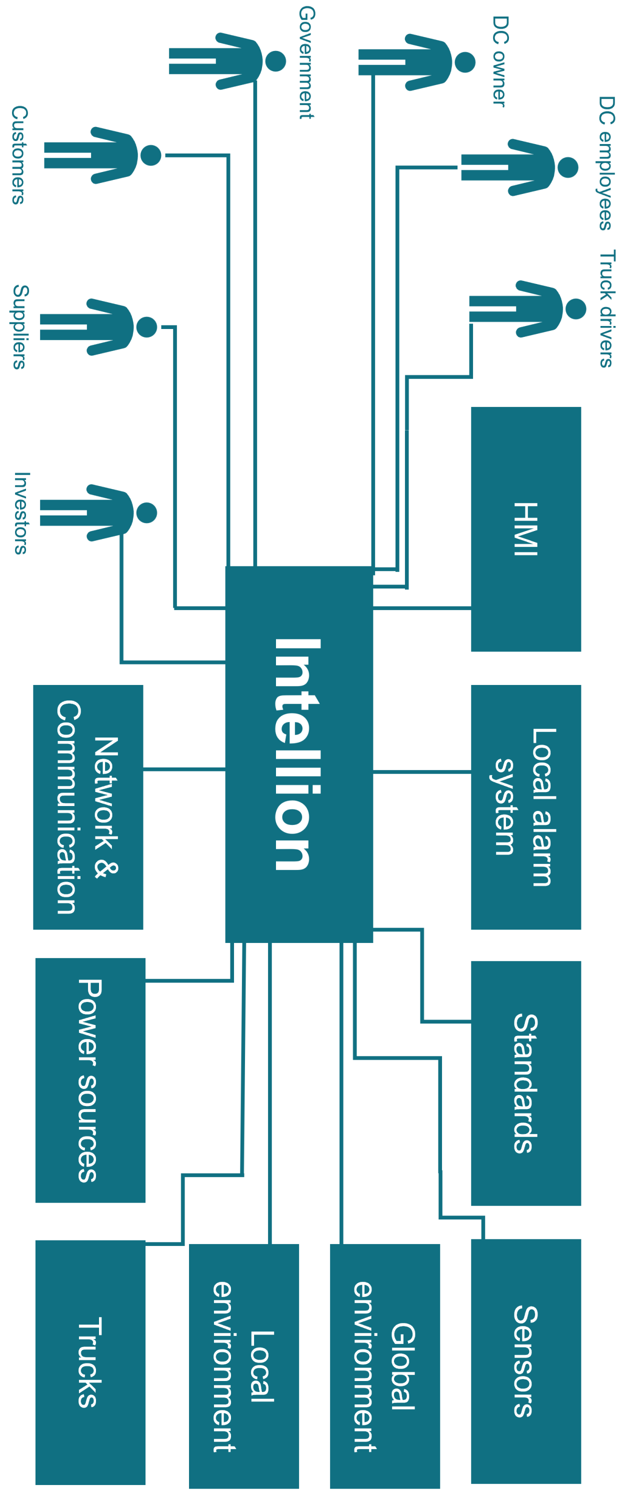


Figure 9: System context diagram

4.2 System Operating Environment

The system operating environment for Digital Twin Autonomous Driving Trucks in a Distribution Center is a complex ecosystem of hardware and software components that work together to enable safe, reliable, and efficient autonomous trucking operations within the Distribution Center. Among the main components of the system operating environment are Sensors, Control and Decision-making Algorithms, Communication Systems, Warehouse Management Systems, Cloud Computing, Machine Learning and Artificial Intelligence, Cybersecurity, Power and Energy Systems, and Human-machine Interfaces Figure 10. Each of these components is further described below:

Sensors: Autonomous trucks used in Distribution Centers are equipped with various sensors, including LIDAR, radar, and cameras, to capture data about their surroundings. These sensors are used to detect obstacles, monitor traffic, and prevent collisions, among other things.

Control and Decision-making Algorithms: The data from the sensors is processed by sophisticated control and decision-making algorithms, which are responsible for making real-time decisions about how the truck should move and respond to different road conditions within the Distribution Center.

Communication Systems: Autonomous trucks in Distribution Centers need to communicate with other vehicles and systems. Communication systems include wireless communication protocols and other networking technologies that enable the truck to transmit and receive data.

Warehouse Management Systems: Digital Twin Autonomous Driving Trucks in Distribution Centers rely on warehouse management systems to coordinate their movements with other vehicles and systems in the Distribution Center. These systems enable the truck to optimize its route and minimize its travel time.

Cloud Computing: Autonomous trucks in Distribution Centers generate a massive amount of data that needs to be stored and processed. Cloud computing technologies enable the storage and processing of this data, allowing the truck to make more informed decisions about how to operate.

Machine Learning and Artificial Intelligence: Autonomous trucks in Distribution Centers use machine learning and artificial intelligence to learn and improve their driving skills over time. These technologies enable the truck to adapt to changing road conditions and improve its efficiency and safety.

Cybersecurity: Autonomous trucks in Distribution Centers need to be protected from cyber-attacks and other security threats. Cybersecurity technologies and protocols are used to ensure the safety and security of the truck's operation and the data it generates and collects.

Power and Energy Systems: Autonomous trucks in Distribution Centers require a reliable and sustainable source of power to operate. This includes batteries, charging infrastructure, and other power and energy systems.

Human-machine Interfaces: Autonomous trucks in Distribution Centers need to provide feedback and information to the warehouse operators. Human-machine interfaces include dashboards, displays, and other user interfaces that enable the operators to interact with the truck and monitor its operations.

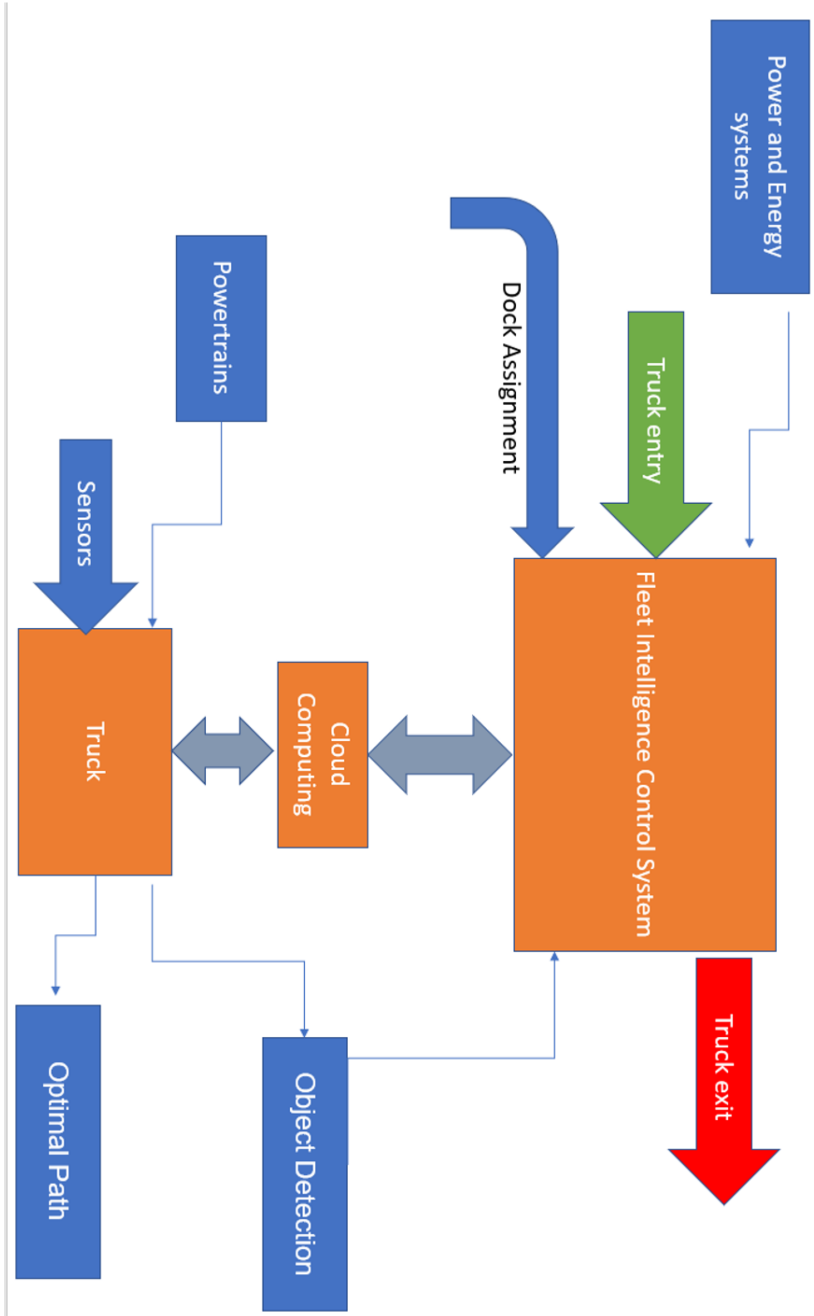


Figure 10: System operating environment

4.3 Functional Safety and Risk Management

With the continuous movement of trucks and personnel inside the Distribution Center (DC), the system has to consider the safety aspects of the area. The functional safety ISO 26262 provides an excellent platform to design the risk mitigation or safety implementation inside the confined space of DCs. [17] and [22]provide good insight into Functional Safety and below are some of the functional safety aspects of the system:

1. **Collision Avoidance:** The system should be capable to prevent collisions between trucks, trucks and DC walls and between trucks and personnel. This can be achieved by using Cameras, LIDAR or other technology to detect the obstacle.
2. **Speed Limitation:** The speed of the trucks should be governed or controlled by the system when trucks enter the DCs. The top speed inside DCs should be set to an appropriate level. This can be implemented using Adaptive Cruise Control, speed governors and other technologies to ensure safe driving.
3. **Pedestrian Safety:** The system and layout should be designed in such a way that the personnel come in contact with the truck's movement as less as possible. Where the pedestrian crossing has to be made on the truck's path, the system should ensure the trucks travel as slowly as possible and stop when needed.
4. **Emergency Stoppage:** The system should incorporate an immediate stop mechanism in the trucks in the case of emergency. This can be implemented using the same technology as in Collision Avoidance. Emergency Stop buttons can also be provided for the control of drivers and other personnel of DCs.
5. **Regular Maintenance:** The pathways and hardware on the trucks should have maintenance checks at regular intervals. This is done to ensure the efficient working and productivity of the components involved.
6. **Alarms and Warnings:** The system should provide adequate warning signs and visual aids to the drivers and personnel of DCs to ensure safety. The alarms and warnings should be installed at various places inside the DCs and also inside the trucks moving around.
7. **Integrity of the system:** The system should be designed keeping in mind the robustness, reliability and efficiency.
8. **Cyber Security:** The software must be protected from illegal access, manipulation, or destruction. This may entail taking actions like using access controls, intrusion detection systems, encryption, and routine security audits and updates.
9. **Redundancy:** Intentional redundancy should be incorporated in the system for it to operate safely even in the event of failure of hardware or software.

Figure 11 depicts the ranking given as Automotive Safety Integration Level specified under ISO26262. This table helps in judging the risk factor involved in the movement of trucks inside the Distribution Centers.

Severity	Exposure	Controllability		
		C1 - Simple	C2 - Normal	C3 - Difficult
S1 - Light	E1 (very low)	QM	QM	QM
	E2 (low)	QM	QM	QM
	E3 (medium)	QM	QM	A
	E4 (high)	QM	A	B
S2 - Severe	E1 (very low)	QM	QM	QM
	E2 (low)	QM	QM	A
	E3 (medium)	QM	A	B
	E4 (high)	A	B	C
S3 - Fatal	E1 (very low)	QM	QM	A
	E2 (low)	QM	A	B
	E3 (medium)	A	B	C
	E4 (high)	B	C	D

Figure 11: Automotive Safety Integration Level in accordance to ISO 26262([11] [12] [13])

The second phase of the safety aspects is risk management, which consists of the methods to be considered to mitigate risk or bring down the level of risks to an acceptable level. Below are the steps to be considered for risk management inside the DCs:

1. **Risk identification:** The Distribution Center should perform a thorough risk analysis of truck movement, taking into account any possible dangers and risks related to the use of trucks inside the facility.
2. **Risk Analysis:** The risks that have been identified should be examined in terms of their potential seriousness, likelihood, and effects on the operations and security of the DCs.
3. **Risk Mitigation:** To decrease the possibility and severity of recognized hazards, appropriate risk mitigation measures should be put into place. The application of safety protocols, the use of safety gear, and training courses for drivers and other staff members are some examples of these measures.
4. **Risk Monitoring:** To make sure that the risk mitigation measures are still effective, the system should continuously assess their efficacy.
5. **Emergency Preparedness:** In the event of a truck accident or other hazardous occurrence, the system should have a thorough emergency response plan in place. This strategy should include procedures for handling injuries, preventing dangers, and guaranteeing the security of all facility staff [18].
6. **Regulatory Compliance:** The system must make sure that all truck transportation operations abide by all applicable regulations, including those pertaining to vehicle speed and manoeuvrability, and safety gear for the personnel.
7. **Reporting of Incidents:** The system needs to set up processes for reporting and looking into occurrences involving truck movements, like collisions or near-misses. Future incidents should be avoided and safety procedures should be improved using this information [15].

4.4 TruckLab environment

The development of a digital twin system includes the real-world counterpart or representation of the virtual model to be designed, developed and tested first, before being integrated in the digital twin. As being an essential element, the physical entity is used to ground the virtual model in reality, provide real-time data, enable validation and verification, facilitating predictive analysis and supports maintenance and optimization efforts. The 4AT100 project focuses on implementing a digital twin system for autonomous driving in distribution centers based on provided templates, frameworks and (communication) infrastructures. Based on the fundamental elements of the TU/e TruckLab environment, both hardware and software (control) components, the physical entity of the digital twin system for the 4AT100 project is built. This section aims to introduce the physical entity, provide an overview of components and give insight on the corresponding software architecture.

TruckLab is a research laboratory that is based on a scaled-down version of an actual distribution center. TruckLab enables developers to design, implement and test autonomous driving features for trucks. In this laboratory, the aim is to accelerate development of autonomous driving of trucks and advance automation in the transportation chain. Figure 12 shows the lay out of the TruckLab laboratory where two trucks are parked in loading/unloading spots.

Figure 12: TruckLab laboratory

TruckLab is a complex system with a significant amount of components. The most important ones are:

- Infrastructure
- ROS network
- Tractors
- Semi-trailes
- Master PC
- Controller PC
- OptiTrack system

where the Master PC acts as Robot-Operating-Software (ROS) master and the Controller PC is running the control files for the autonomous trucks. An illustration of the TruckLab components and communications is shown in Figure 13.

To demonstrate the components and their linkages in an more understandable way, Figure 13 is created. The linkages in the figure represent wireless signals that are transmitted over the TruckLab's WLAN. The ROS network facilitates communication between the ROS nodes that are present on multiple components of the TruckLab system. The information flowing in a ROS network is called a topic. A topic defines the type of messages that will be send concerning that topic. A publisher-subscriber model is used to actually convert information from one ROS node to another.

Another component is the OptiTrack system. This system is important to keep track of the truck's global position within the TruckLab area. By making use of 8 cameras pointing at the distribution center area, a precise estimation of the truck's position can be made.

The eventual goal for the 4AT100 project within the implementation phase was to drive 3 trucks simultaneously in the TruckLab environment that are using different controller designs. Since the control designs originate from different developer groups, this problem is extremely affected by communication and teamwork. It was therefore important to design a common communication framework to allow smooth communication between the trucks. The framework consists of a distribution of running Simulink files. The DC dashboard, to give commands to the autonomous trucks runs on a seperate Simulink file, whereas the three truck controllers are integrated in a single Simulink file as shown in Figure 14.
The groups together agreed on a common information share between the trucks. This information included the following components:

1. x position of a truck
2. y position of a truck
3. heading angle of a truck
4. priority of a truck
5. path vector
6. truck state
7. x trailer position
8. y trailer position
9. trailer heading angle

To receive the information, a truck can subscribe to the corresponding ROS topics. Using the Matlab-ROS integration, the subscriber block for truck 1 was created (Figure 15). A slight modification of this subscriber block was used for truck 2 and 3 as well.

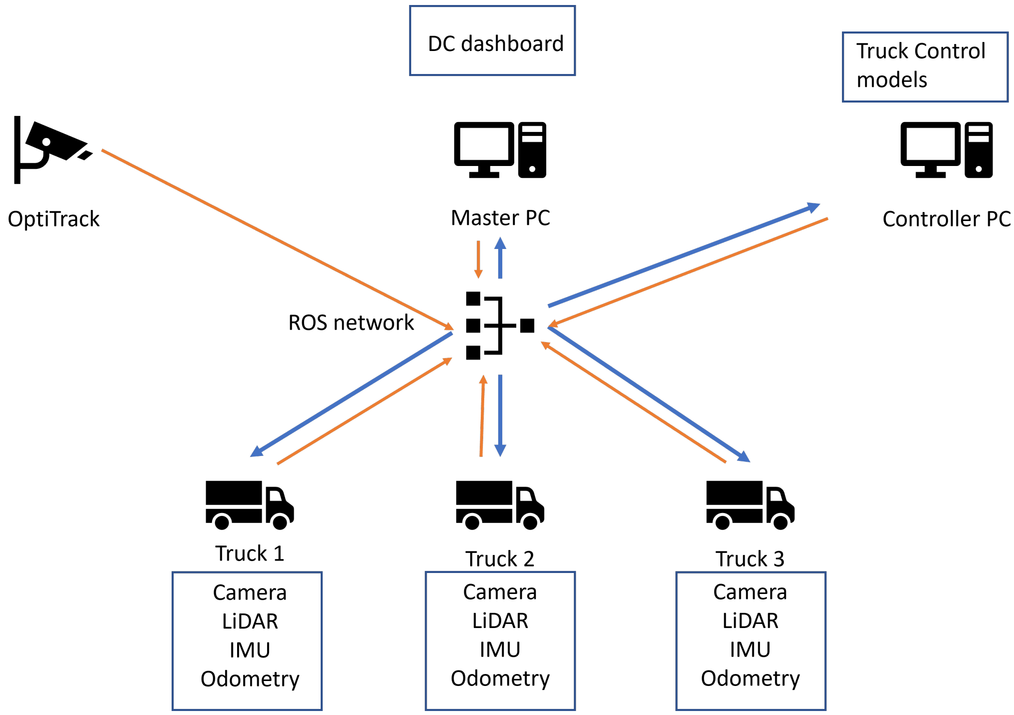


Figure 13: Components & communication links

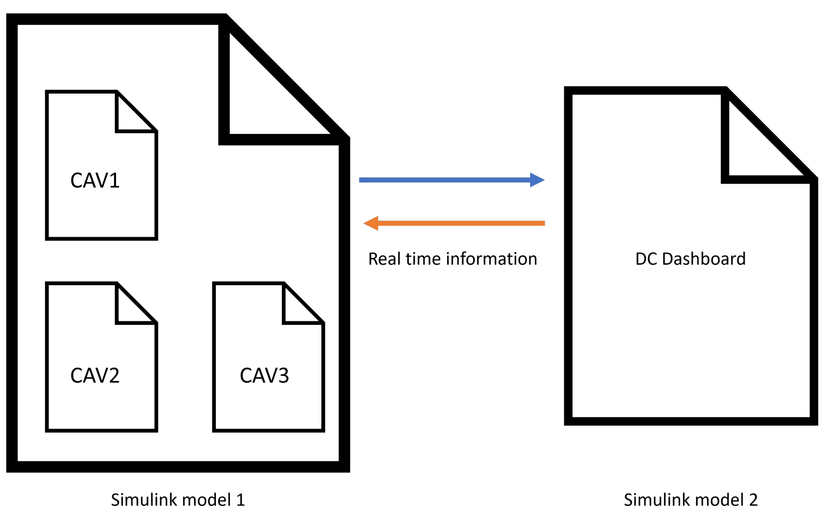


Figure 14: Common software architecture

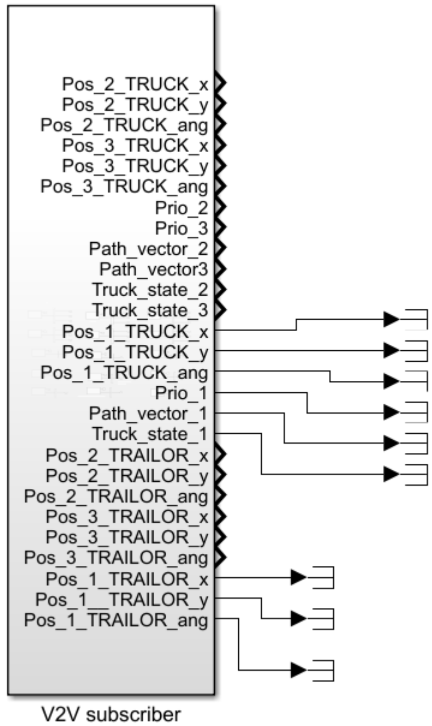


Figure 15: Subscriber block for autonomous truck

4.5 System architecture

The conceptual layout and organization of a computer system or software program are referred to as system architecture. It specifies how a system's various parts are organized, how they communicate with one another, and how they work together to produce the intended functionality and performance. The layout and design of hardware, software, communication protocols, data storage, and other significant components make up system architecture. For developing, deploying, and sustaining a system or application, it offers a blueprint or road map. This section provides information on the conceptual architecture which will be used in the project. The section is divided into three sections providing insight on different entities involved in the project.

4.5.1 Digital Twin

The system architecture consists of the Control and Data flow between all the operating entities. The architecture also shows the internal components of the system. INTELLION manages the assignment of the docks to the trucks. It stores the status of the Docks and available trucks. The trucks have control system which receive signals from the sensors and correspondingly control the movement of the truck. The Docking and Parking stations have sensors to detect the presence of the trucks and notify it to the system. The detailed architecture can be seen in Figure 16

4.5.2 Virtual entity

Virtual Twin can be designed using various Softwares such as Unity Game Engine, Unreal Engine, Blender etc., where the environment such as the Distribution Center can be designed to a required scale. The behaviour of the truck which consists of the Wheelbase, Trackwidth, Powertrain and Kinematic Behaviour. This allows the designers to visualise the movement of the trucks within the Distribution Center. The Truck Lab present in TUE has the software "Unity Game Engine". This software has a physics engine and scripting features using "C-Sharp" language, which facilitate the creation of different scenarios and evaluate potential bottlenecks. This will inturn help to optimize the Distribution Center's efficiency and safety. Lastly, Unity

Game Engine serves as a valuable tool for designing and verification process and facilitates iterative testing before implementing the technology into the physical twin.

4.5.3 Physical entity

The outcomes of the virtual testing of autonomous trucks can be used as a useful benchmark when deploying the technology in actual trucks and evaluating the results. Engineers can fine-tune and perfect the algorithms, control systems, and navigation tactics of the autonomous trucks by carefully examining the data and insights gained during virtual testing in Unity Game Engine. The extensive experimentation and scenario testing made possible by the virtual environment aid in identifying any potential flaws or areas that could want improvement. Developers can confidently incorporate the improved algorithms and control systems into the physical vehicles using the knowledge they obtained from virtual testing, ensuring that they have improved capabilities for autonomous navigation within the distribution center. Engineers can verify the accuracy and dependability of the virtual simulation by rigorously comparing the performance of the physical truck with the outcomes of virtual testing. This procedure serves to foster trust in the technology's ability to be used in the real world, ensuring that the autonomous trucks can operate safely and effectively in the setting of the distribution center.

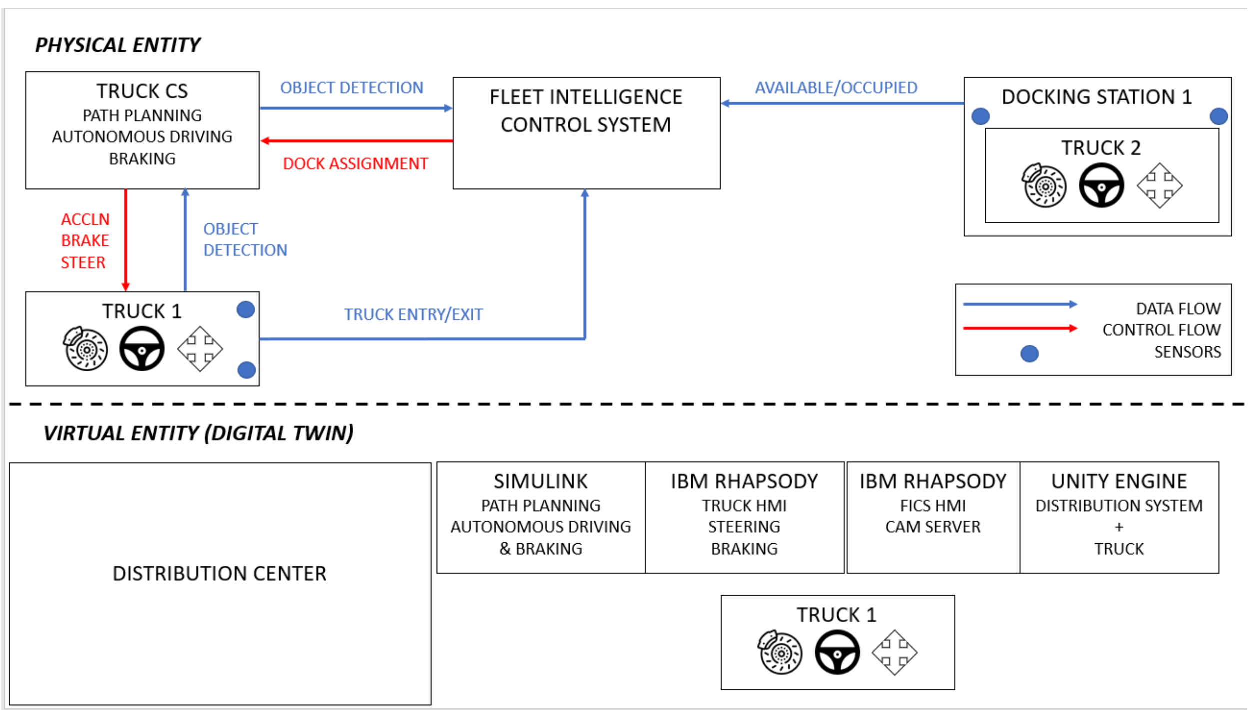


Figure 16: Conceptual Architecture

5 User Stories

User stories are a critical component of the development methodology. A user story is a short, simple description of a system feature of functionality forms the perspective of the stakeholders. It captures the who, what and why of a particular feature and therefore allows the development team to better understand the user's needs. This section enumerates a set of user stories from the perspective of all stakeholders mentioned in subsection 4.1.

1. As a Distribution Center (DC) owner, I want only one truck at each docking/undocking station. The rest of the trucks have to be parked in the parking area.
2. As a DC owner, I want the truck's velocity to be a maximum of 12 Kmph inside the DC.
3. As a DC owner, I want the Autonomous trucks to detect each other and any other obstacles and apply emergency brakes if required.
4. As a DC owner, I want the truck control system to take into account the battery level and charging stations' availability to ensure that the autonomous truck can complete its delivery without running out of power.
5. As a DC owner, I want the autonomous trucks to follow an optimized path while navigating through the Distribution Center to minimize travel time and reduce congestion.
6. As a DC owner, I want the trucks to drive autonomously so that fewer accidents occur and costs are reduced.
7. As a DC owner, I want the DC to be cost-effective and more efficient compared to the current DC.
8. As a DC owner, I want the DC to minimize emissions so that I can promote my brand to be sustainable.
9. As a DC owner, I want INTELLION to detect and report any technical issues.
10. As a DC owner, I want INTELLION to meet all the safety and legal requirements.
11. As a DC employee, I want to have a safe working environment so that I experience my work as safe.
12. As a DC employee, I want INTELLION to have an easy-to-use human-machine interface.
13. As a DC employee, I want electric trucks so that my work environment is cleaner.
14. As a truck driver, I want to use my waiting time to rest so that I can use my time more efficiently.
15. As a truck driver, I want to receive updates from the system, so that I know where my vehicle is.
16. As a truck driver, I want the user interface to be intuitive so that it is clear what is happening and what actions I should undertake.
17. As a supplier, I want the Distribution Center to support my supply amount.
18. As a customer, I want my goods as fast as possible, so that my revenue is not affected.
19. As a customer, I want my goods to be distributed/grouped efficiently so that my processing time is minimized.
20. As a society, I want to experience little noise from the DC, so that I can live conveniently.
21. As a society, I want the DC to minimize emissions, so that the environment and health are not affected.
22. As an investor, I want to receive credibility so that I remain sure about the investment.
23. As an investor, I want the DC to pay back the investment within 3 years
24. As a government, I want the DC to be a safe working area so that little or no injuries occur.
25. As a government, I want the DC to contribute to sustainability so that the environmental regulations are met.
26. As a government, I want the DC to retain most of its employees so that employment is preserved

6 Requirements

To convert the user view expressed in user stories to a form that is usable for engineers, the user stories were translated into requirements. By using requirements engineers can implement the required features and services in a convenient and structured way. Since INTELLION is a complex system, capturing a lot of different domains, the requirements are originating from multiple sources. Therefore, it is of high importance that engineers can keep an overview of which requirement is relevant to which system component.
This challenge is solved by grouping the requirements and showing them as diagrams. The requirements are separated into two main groups: Business requirements which express the team's vision and features, and General requirements which express the requirements originating from the INTELLION system. The general requirements are further divided into the domains that are captured in the project.

Figure 17 shows the overall structure of the requirements. The sub packages of the general requirements represent the domains for which a set of requirements is defined. To provide an idea of what the requirements look like for Intellion, the following examples are presented:

1. When a docking station X has status "occupied", the DTADTS shall freeze X to be assigned until it's status is changed to "available".
2. Under all driving conditions, the Truck Control System shall not allow the truck and semi-trailer to jack-knife.
3. The Truck Control System shall monitor the charge of the truck's battery and display it on the truck's HMI dashboard.
4. The autonomous truck shall be equipped with sensors such as LiDAR and camera to detect other autonomous trucks and obstacles in its path.

Besides these four examples, there are numerous other requirements. Appendix B shows the per-domain requirement diagrams and all the project relevant requirements.

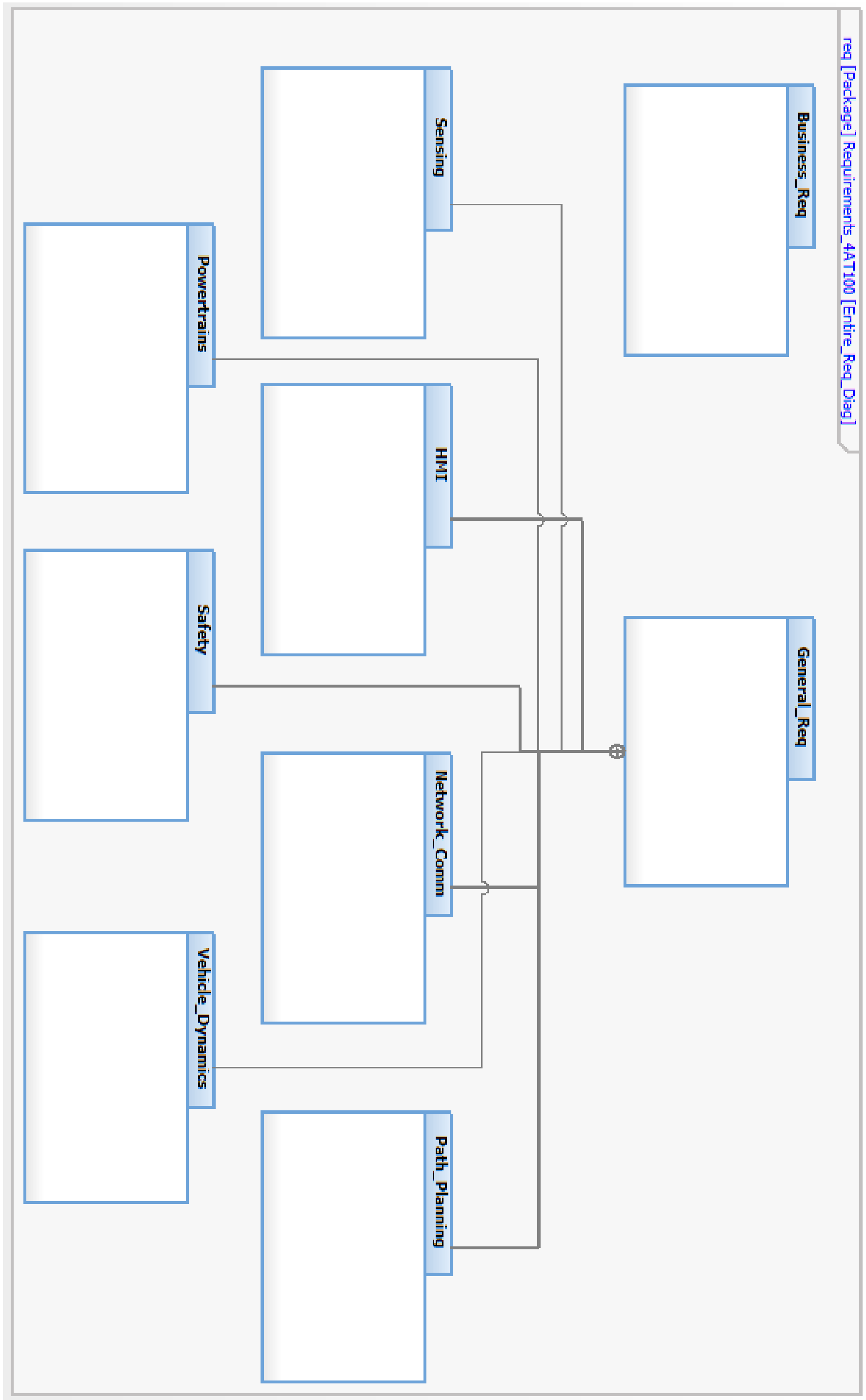


Figure 17: Requirements Overview

7 Use Cases

This section covers the use cases related to the project. The use cases capture, model and specify the system requirements and provide the engineers with a better understanding of the system. The use cases are based on the formulated requirements in section 6.
This section will cover the use case diagrams first followed by descriptions. In the use case descriptions, every use case is explained in more detail, including relevant actors, conditions and flow components. The last part covers the functional safety aspect of every use case.

In this section, only one of the four scenario parts is presented in terms of use case diagram, textual description and ASIL safety rating. The other three scenario parts are shown in Appendix C.

7.1 Use Case Diagrams

The use case diagrams are based on phase components of the system. Each truck entering the DC area will experience three phases before leaving the DC area. Each phase has its own use cases and services. For every system phase, a separate use case diagram is made.
INTELLION consists of the following system phases:

1. Truck entry phase (Figure 18)
2. Docking phase (Figure 50)
3. Truck exit phase (Figure 51)

The truck entry phase is the phase in which the truck enters the DC area and the driver parks the truck at the parking location. The docking phase is the phase in which the truck is unloaded and loaded again. To complete the cycle, the third phase covers the truck parking at the parking location and is ready to be picked up by the driver again.
Besides the use cases that are belonging to a single system phase, there are other use cases that are relevant during the whole system operation. Therefore, there is made another use case diagram to cover this aspect as well. This use case group mainly aims to cover error situations and is therefore called the 'error situations' diagram and is shown in Figure 52.

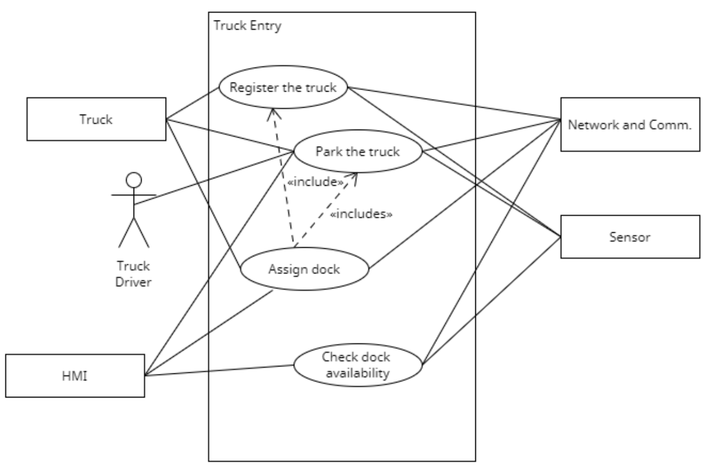


Figure 18: Use case diagram: Truck entry phase

7.2 Use Case Description

Since a single use case name might not be entirely clear, a description format is used to show the relevant actors attached to the use case, flow components, error situations and possible alternative flows. The use case descriptions are shown in Table 4. In this table, flow components written in bold, refer to another use case.

Table 2: Use case descriptions

no. 1	**Use Case Name**	**Register the truck**
	Short description	The INTELLION establishes a connection with the incoming truck
	Primary Actors	Truck, Truck Driver
	Secondary Actors	Network and Communication
	Preconditions	Truck enters the DC area
	Normal flow	1. The INTELLION sends the request signal to the truck 2. Truck receives and accepts the request 3. INTELLION adds truck to database 4. INTELLION sends an acknowledgement back to the truck 5. The truck receives the acknowledgement
	Postconditions	There is a connection between INTELLION and truck
	Error situations	1. INTELLION does not receive the request signal 2. The truck does not receive the acknowledgement
	Alternative flow(s)	1.1 The truck driver initiates the request in the truck HMI 1.2 The INTELLION receives the request 1.3 Go to 3
no. 2	**Use Case Name**	**Park the truck**
	Short description	The truck driver drives the truck from DC entrance to an available parking space
	Primary Actors	Truck, Truck driver
	Secondary Actors	Sensors, Network and Communication, HMI
	Preconditions	Truck is registered to the INTELLION
	Normal flow	1. Driver drives to parking space 2. HMI displays the truck's battery level 3. Truck driver plugs the truck into a charger 4. Driver leaves the truck 5. Parking spot becomes unavailable
	Postconditions	Truck is parked
	Error situations	
	Alternative flow(s)	1.1 Truck switches to electric mode (if hybrid)
no. 3	**Use Case Name**	**Check dock availability**
	Short description	INTELLION senses the presence of the trucks and determines dock availability
	Primary Actors	Truck, Sensors
	Secondary Actors	Network and Communication, HMI
	Preconditions	none
	Normal flow	1. Present truck leaves dock X 2. Sensor detects the truck that's leaving the dock 3. INTELLION processes the sensor signal 4. INTELLION changes the status of dock X to 'available' 5. HMI displays that dock X is available
	Postconditions	The dock status of the docks is up to date
	Error situations	1. INTELLION does not receive the signal of the sensor 2. HMI does not update accordingly
	Alternative flow(s)	1.1 DC employee changes the status of Dock X by manually pressing a button from Dock X 2.1 **Network failure**
no. 4	**Use Case Name**	**Assign dock**
	Short description	Assign a truck to a dock
	Actors	Truck, Network and Communication
	Secondary Actors	HMI
	Preconditions	Truck A is available at the parking area
	Normal flow	1. INTELLION checks the dock ID of the available dock X 2. INTELLION sends the available dock ID to truck A 3. INTELLION changes the status of dock X to 'occupied' 4. HMI displays that dock X is 'occupied'

Table 3: Funtional safety ratings

Use Case Name	S	E	C	Motivation	ASIL rating
Register the truck	S0	E4	C1	1. S0 - No damage done if registration fails. 2. E4 - Every incoming truck is registered. 3. C1 - Simple exchange of signals for registration.	QM
Park the truck	S2	E4	C2	1. S2 - Possibility of collision with other trucks. 2. E4 - Frequent parking of trucks before and after docking. 3. C2 - Moderate complexity in executing parking procedure.	B
Check dock availability	S0	E4	C2	1. S0 - No damage done if dock availability check fails. 2. E4 - Frequent checking of availability of docks. 3. C2 - Simple process to check dock availability.	QM
Assign dock	S0	E4	C2	1. S0 - No damage done if dock assignment fails. 2. E4 - Frequent assignment of the docks to the trucks. 3. C2 - Moderate complexity in dock assignment.	QM

8 Implementation

This section covers the steps taken to achieve the working of the intended system. Beginning with the design of the motion controller, the next sub-sections cover the Human Machine Interface (HMI), System Architecture and Complete Integration.

8.1 Motion Controller

After the requirements and use cases were thoroughly defined, the primary goal was to get the truck moving. The objective was to enable the truck to follow a specific trajectory within the distribution center, utilizing a specially designed controller. This controller would input the desired path for the vehicle and output the corresponding steering angle of the truck. This controller was first designed on a kinematic model of the truck in Matlab, secondly tested on a virtual model of the truck in Unity Game Engine and lastly tested on a physical scale model of the truck and distribution center.

8.1.1 Pure pursuit controller

To get the truck to drive along a given path, a pure pursuit controller was developed. The primary functionality of this controller revolves around its ability to follow a reference path, relying solely on the vehicle's kinematic geometry and the characteristics of the desired path. In doing so, it simplifies the complexity of real-world vehicle dynamics by disregarding dynamic forces exerted on the vehicle. This assumption is primarily based on the premise that wheels maintain a no-slip condition, effectively adhering to the contact surface without skidding. The controller operates by continuously adjusting the vehicle's steering angle to ensure it is always directed towards a "goal point" on the reference path. This point is also known as the "Look Ahead" point. The algorithm continually shifts the look-ahead point along the path based on the vehicle's current position and continues until the vehicle reaches the final point on the path. The positioning of this Look Ahead point is determined by the "Look Ahead Distance" parameter, which dictates the distance between the vehicle and the look-ahead point and this is the main tuning parameter of the controller [20]. As shown in Fig. 19, the path is oscillatory and accurate for a smaller look-ahead distance and, for a larger look-ahead distance, the path is less oscillatory, but the tracking is poor.

8.1.2 Kinematic Bicycle model

The proposed analysis uses a simplified bicycle model, shown in Fig. 20(a), to represent the vehicle, where the vehicle's front and rear wheels are each represented by a single wheel. This model provides a useful tool for exploring the vehicle's kinematics. Within this model, we establish a reference point, denoted as X, Y, which can be variably positioned on the vehicle. This point can be positioned at various locations, such as at the centre of the rear axle, the centre of the front axle, or the centre of gravity, often referred to as 'cg'. For the pure pursuit controller, the centre of the rear axle is selected as the reference point, denoted by 'x_r'

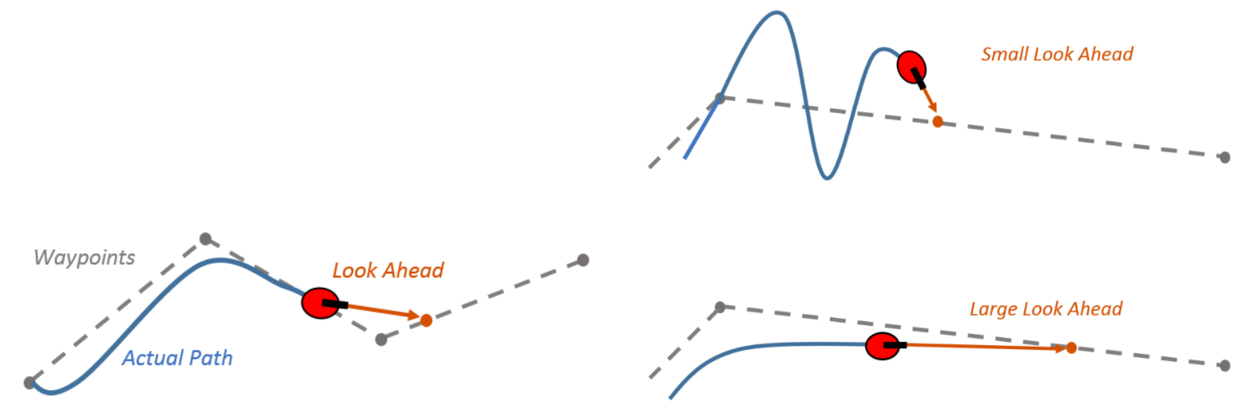
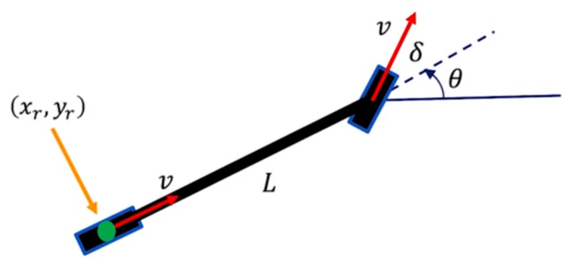
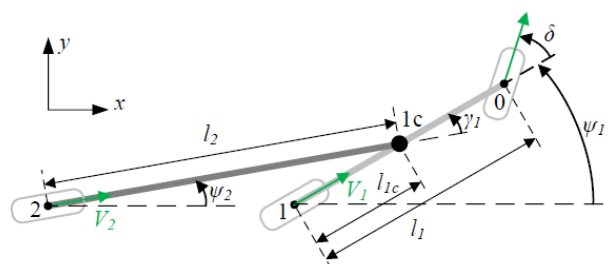


Figure 19: Effect of varying look ahead distance on pure pursuit path tracking.



((a)) Rear axle kinematic bicycle model.



((b)) Tractor semi-trailer kinematic model.

Figure 20: Bicycle Models.

and 'y_r'. 'θ' refers to the heading angle. Another key parameter is the wheelbase, denoted as 'L', which is the distance between the front and rear axles. The steering angle, denoted as 'δ', is measured in relation to the forward direction of the bicycle model. The velocity 'v' corresponds with the direction of each wheel, adhering to the 'no slip' condition. From the no-slip condition,

$$\dot{\theta} = \omega = \frac{v}{R}, \tag{1}$$

where ω is the bicycle's rate of rotation. From the similarity of triangles formed by R and L, and δ and v,

$$tan(\delta) = \frac{L}{R}, \tag{2}$$

From (1) and (2),

$$\dot{\theta} = \omega = \frac{v}{R} = \frac{v \cdot tan(\delta)}{L}, \tag{3}$$

From this configuration, the components of velocity for the reference point in the x and y directions are,

$$\dot{x_r} = v \cdot cos(\theta), \tag{4}$$

$$\dot{y_r} = v \cdot sin(\theta), \tag{5}$$

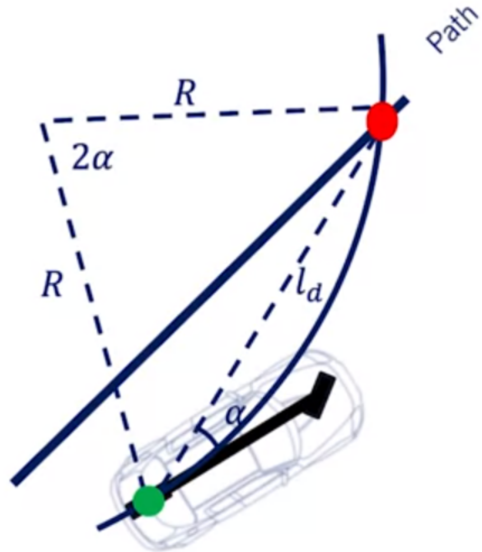


Figure 21: Pure Pursuit geometry indicating the look ahead point and look ahead distance.

These equations, along with the equation for the bicycle's rate of rotation derived earlier, collectively constitute the rear axle bicycle model and is shown below,

$$\dot{\theta} = \frac{v \cdot tan(\delta)}{L}, \tag{6}$$

This model provides a concise mathematical representation of the vehicle's motion, which is crucial for designing the path-following controller.

For implementing the pure pursuit controller for ADTs (Autonomous Driving Trucks), this research considers the dynamics of a Tractor Semi-Trailer model shown in Fig. 20(b). However, the existence of three non-steered trailer axles adds complexity. During a turning manoeuvre, side slip angles inevitably manifest on multiple trailer axles, irrespective of the speed. The consequential forces may induce side slip angles on the tractor as well. In the context of low-speed manoeuvring, it is reasonably assumed that no side slip angle occurs on the tractor and the second axle of the semi-trailer. Under these circumstances, the first and third trailer axles are disregarded for simplicity. One significant challenge with the tractor-semi-trailer configuration is the occurrence of jackknifing. In this situation, the articulation angle between the tractor and the semi-trailer becomes excessively large, potentially leading to a collision between the tractor cabin and the semi-trailer. Constraints are set on the maximum steering angle to avoid this scenario, especially during forward motion.

8.1.3 Geometric interpretation of pure pursuit algorithm

For the Pure Pursuit control strategy, the key reference point on the vehicle is the centre of the rear axle. The line connecting this reference point to the target point on the planned path is defined as l_d, commonly referred to as the look-ahead distance, represented by the blue dashed line in Fig. 22. The angle 'α' represents the angle between the heading of the vehicle and the look-ahead line. The interpretation of Pure Pursuit control from a geometric perspective relies on applying the instantaneous centre of rotation (ICR) principle. The ICR, the centre of the rear axle, and the target point on the path together form a triangle with two sides of length R (radius of the ICR circle) and one side of length 'l_d' (the look-ahead distance). The arc the vehicle needs to follow to reach the target point from its current reference point can be defined within this triangle. This arc lies on the circumference of the ICR circle and spans an angle of 2α. By utilizing fundamental trigonometric identities and applying the law of sines, the angle 2α can be obtained as shown below,

$$\frac{l_d}{sin(2\alpha)} = \frac{R}{sin(\frac{\pi}{2} - \alpha)}, \tag{7}$$

$$\frac{l_d}{2 \cdot sin(\alpha) \cdot cos(\alpha)} = \frac{R}{cos(\alpha)}, \tag{8}$$

$$\frac{l_d}{sin(\alpha)} = 2 \cdot R, \tag{9}$$

$$\kappa = \frac{1}{R} = \frac{2 \cdot sin(\alpha)}{l_d}, \tag{10}$$

where κ is the curvature of the circular arc, which is the inverse of the radius of the ICR circle (R). Combining (2) and (10), the steering angle needed to track the arc is given by,

$$\delta = \tan^{-1}\left(\frac{2 \cdot L \cdot sin(\alpha)}{l_d}\right). \tag{11}$$

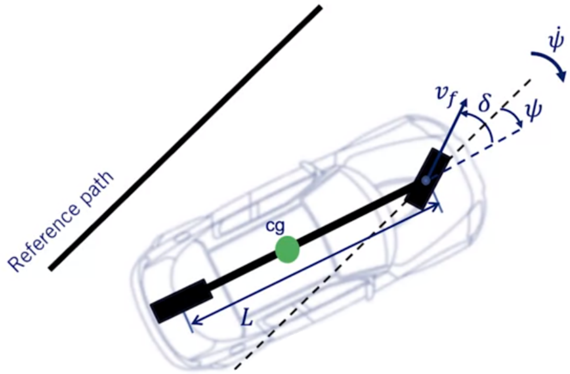
8.1.4 Pure pursuit algorithm- error dynamics

The controller ensures that the autonomous truck aligns precisely with the intended route, minimizing deviations. The variable 'ψ', shown in Fig. 23(a). represents the relative heading angle of the truck in relation to the desired path. Two critical types of errors come into play in this scenario: heading and cross-track errors. Heading error, the discrepancy between the intended path direction and the truck's actual heading, is a primary indicator of how well the vehicle aligns and moves in the direction of the desired path. The rate of change of the heading error, denoted as '$\dot{\psi}$', provides insight into the evolution of the heading error over time. The equation is shown below,

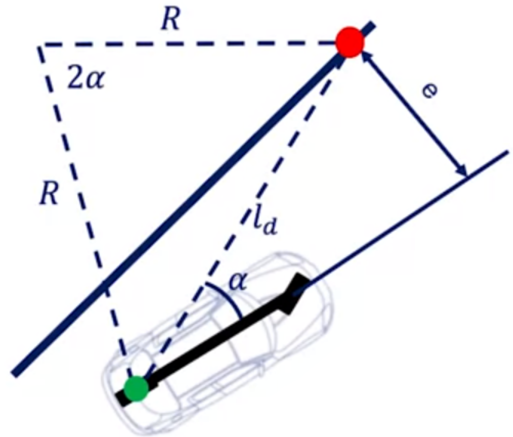
$$\text{Rate of heading error} = \dot{\psi}_{des}(t) - \dot{\psi}(t), \tag{12}$$

where $\dot{\psi}_{des}(t)$ is the desired rate of change of heading angle. The second form of error, the cross-track error, represents the displacement between the vehicle's reference point and the nearest point on the planned path. This is shown in Fig. 23(b). A flawless path-tracking mechanism would require the heading and cross-track errors to be reduced to zero. In the context of the pure pursuit controller for an autonomous truck, the cross-track error is the gap between the truck's directional vector and the target point on its path, represented with the symbol 'e'. The following equation is obtained,

$$sin(\alpha) = \frac{e}{l_d}. \tag{13}$$



((a)) Bicycle model showing the vehicle heading angle 'ψ'.



((b)) Bicycle model showing the vehicle cross track error 'e'.

Figure 22: Error Dynamics.

Integrating this with the previously derived expression for curvature, from (10) and (13),

$$\kappa = \frac{2}{l_d^2} \cdot e. \tag{14}$$

Equation (14) indicates that the path's curvature dictated by the pure pursuit controller is directly proportional to the cross-track error at the look-ahead point on the path.

8.2 Pure pursuit controller design and simulation setup

A kinematic vehicle model of a tractor-semitrailer truck was provided in MATLAB/Simulink, enabling the simulation of low-speed manoeuvring. The model allows for the prescription of velocity and steering as a function of distance, facilitating the computation of the resulting vehicle motion. In the pure pursuit controller, critical parameters derived from the kinematic model, including the rear axle coordinates (x1, y1), the truck's current position (x1c, y1c), and the heading angle of the truck 'ψ', serve as the input variables. The controller's function block shown in Fig. 23, processes these inputs and outputs the required steering angle accurately guide the truck along the planned path. The system diagram can be visualized in Fig. 24 A series of 1-D lookup tables implemented in Simulink handle the dynamic update of waypoints as a function of the distance travelled. The simulation is designed to halt once the vehicle reaches or is within a set proximity threshold of the last point in the waypoint.

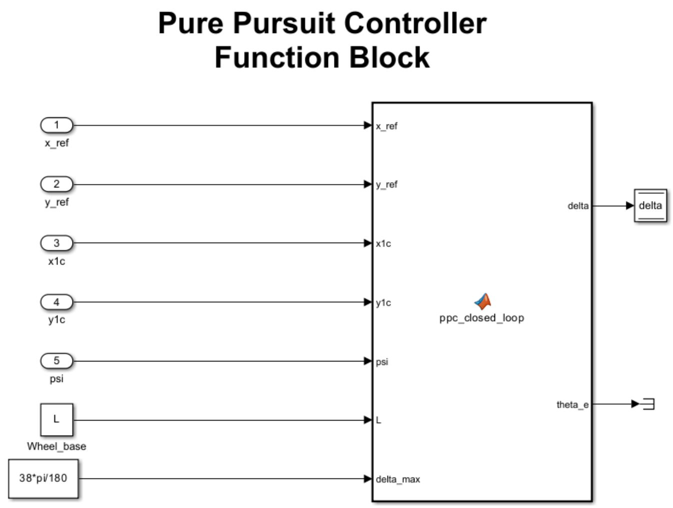


Figure 23: Pure pursuit controller function block implemented in Simulink.

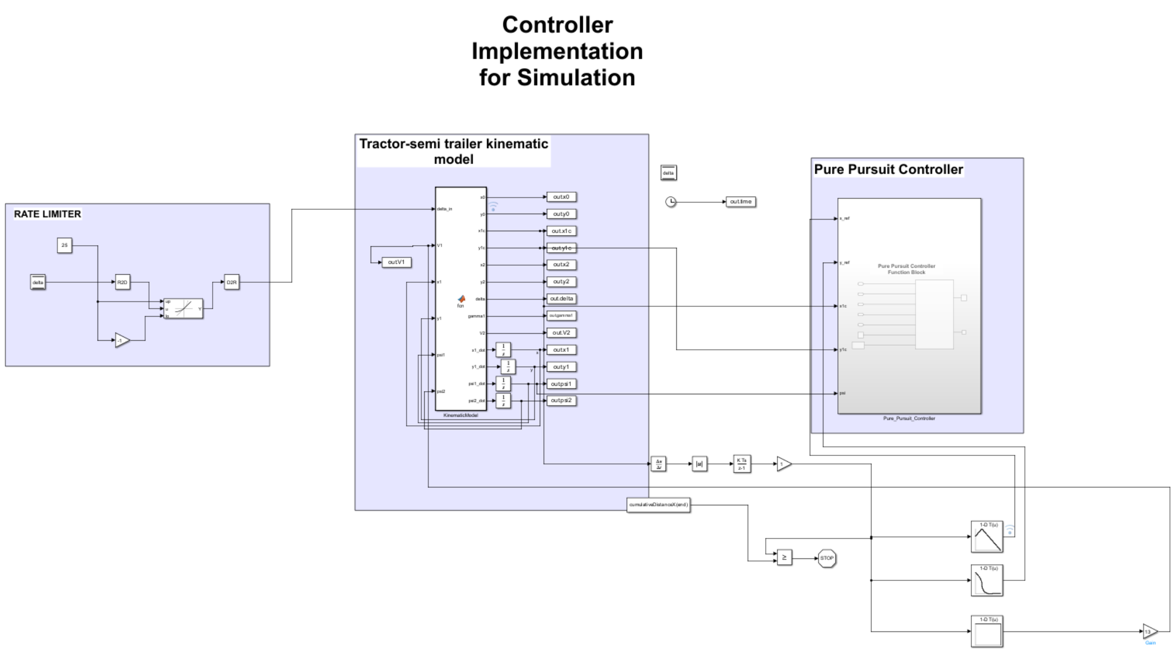


Figure 24: An overview of the kinematic tractor-semitrailer Simulink block paired with its controlling entity. The system also features the integration of a rate limiter block to prevent jackknifing.

Upon successfully validating the pure pursuit controller using the provided kinematic model, the next stage involves its integration and validation within a 3D virtual environment called the Virtual Entity. This environment has been designed utilizing the Unity Game Engine. The practical application of the controller, translated into MATLAB/Simulink blocks for driving the tractor within the Virtual Entity, is detailed in Figure 25. For a more comprehensive understanding of the controller's operational code used in the virtual simulation, readers are referred to Appendix B. This approach ensures a thorough validation process, extending from abstract model simulations to an immersive 3D environment, thus underlining the potential for practical deployment of the pure pursuit controller in real-world applications.

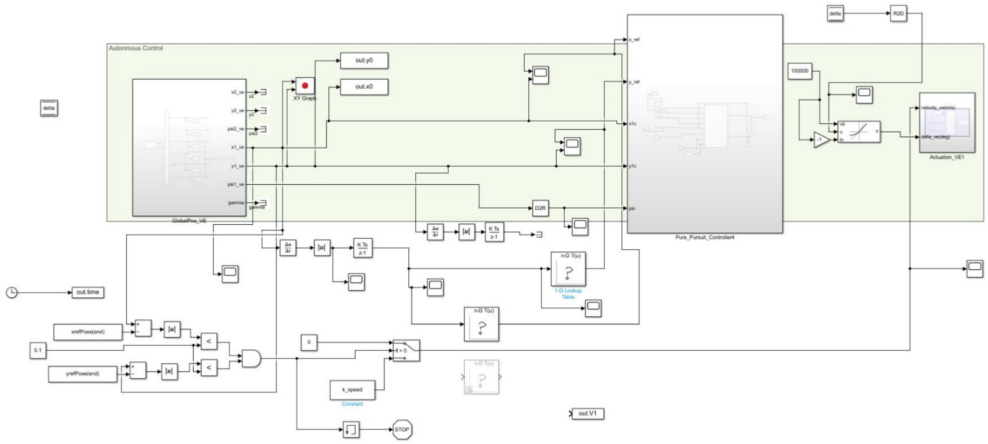


Figure 25: Depicts the integration of the pure pursuit controller with the Virtual Entity blocks.

In the final stage of methodology, the pure pursuit controller is subjected to testing and validation using the AES Lab's physical truck. To facilitate this physical testing, the Virtual Entity blocks within the simulation setup are replaced with Physical Entity blocks. These new blocks interface with the global position sensor and actuator of the physical truck, ensuring a direct link between the controller and the hardware of the truck. This configuration is outlined in Figure 26

Through this multi-tiered approach, progressing from kinematic modelling, through a virtual environment, to actual physical testing, our methodology ensures a comprehensive validation of the pure pursuit controller, reinforcing its suitability for real-world ADT applications.

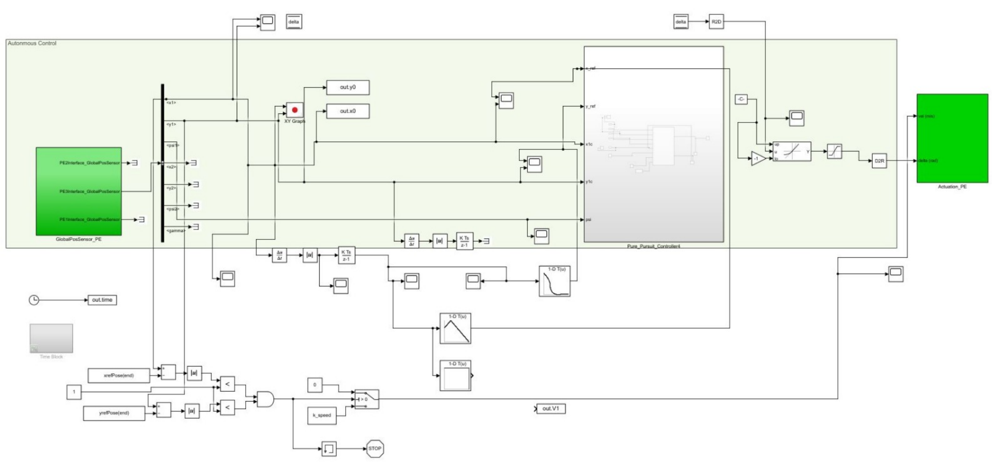
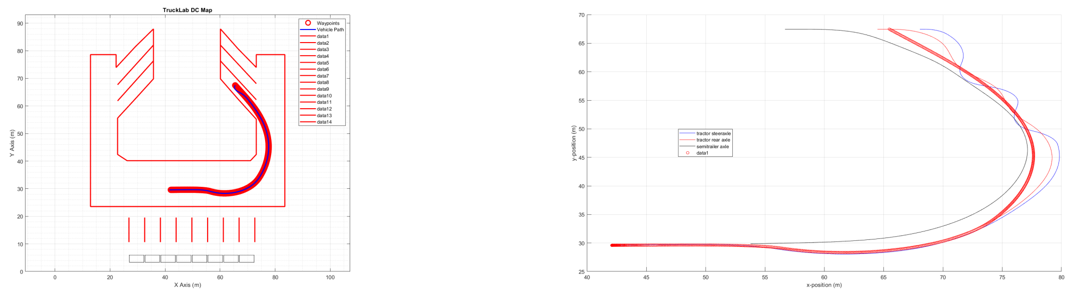


Figure 26: Depicts the integration of the pure pursuit controller with the Physical Entity blocks.

8.2.1 Implementation results

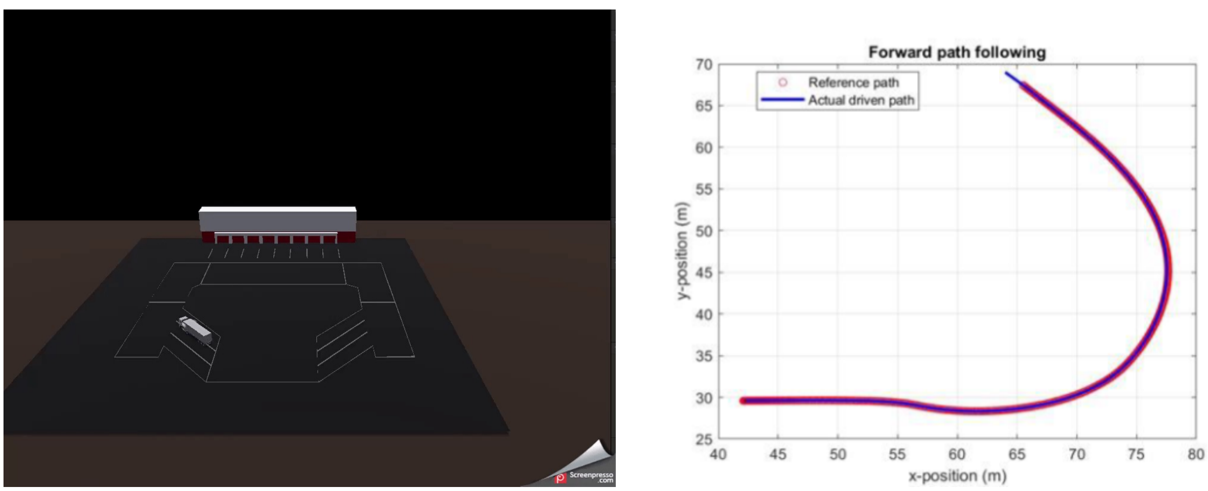
In this section, the key findings from the three-tiered simulation process are showcased, highlighting the performance of the pure pursuit controller under various simulation conditions. In the first stage, the kinematic model simulation, the plot shown in Fig. 28(b), demonstrates the truck's ability to almost precisely track the given path, which highlights the accuracy and reliability of the pure pursuit controller. An interesting observation from the plot pertains to a series of oscillations in the path. These oscillations can be attributed to the truck's starting position being proximate to the beginning of the waypoint. This proximity leads to the computation of a relatively small look-ahead distance, which subsequently introduces oscillations into the truck's trajectory. The TruckLab map is also provided in Fig. 28(a), illustrating the truck's path along the designated waypoints. This graphic representation affirms the practicality and feasibility of our controller when deployed in real-world distribution center scenarios.



((a)) Illustration of the TruckLab map. ((b)) Truck navigating through a set of waypoints.

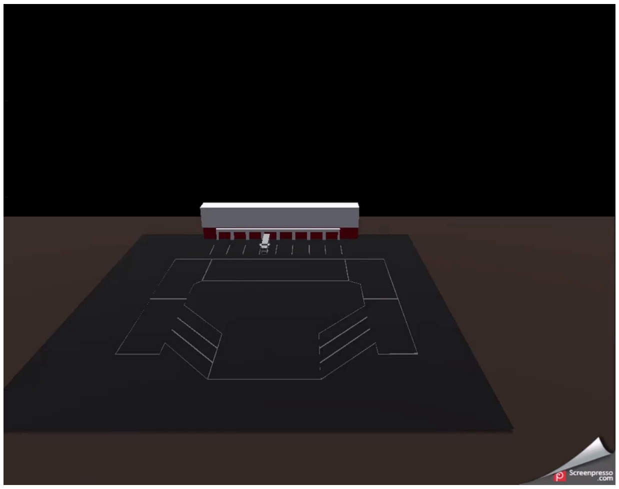
Figure 27: Kinematic model simulation results.

In the second stage, the findings from the virtual simulations are shown in Fig. 29(b) and Fig. 30(b). For these simulations, two calibrated path-following tasks were conducted involving the truck operating within a virtual representation of the distribution center. The first task involved navigating from the entrance of the distribution center to the docking area. The second task involved navigating the truck as it made its way from the docking area back toward the exit of the distribution center. In both scenarios, shown in Fig. 29(a) and Fig.30(a), the simulation plots reveal that the truck maintained a high degree of accuracy in following the pre-defined path.

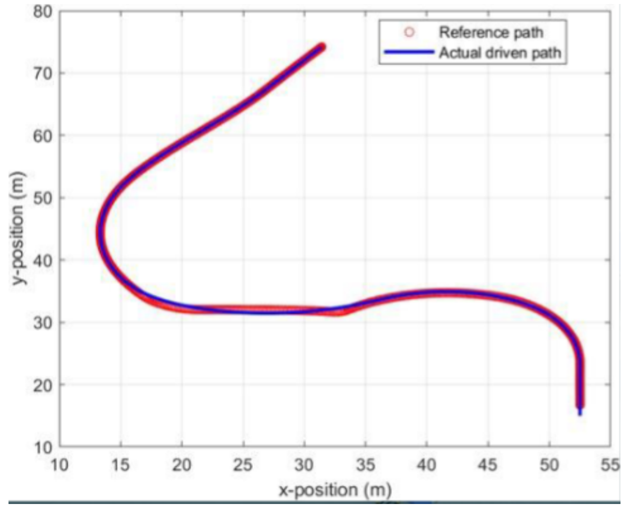


((a)) Scenario 1: Truck located at the entrance. ((b)) Truck precisely following calibrated path 1.

Figure 28: Virtual simulation results (Path 1).



((a)) Scenario 2: Truck located at the docking area.



((b)) Truck precisely following calibrated path 2.

Figure 29: Virtual simulation results (Path 2).

In the last stage, the controller was applied to the physical entity. Also here, the controller proved that it was working, as the truck followed the path precisely. Unfortunately, there is no figure with the results available, but a video of the truck driving along the path of the distribution center will be added to the report (some snapshots of this video are shown in Section 9). It was noted that the performance of the physical entity was a bit less good than that of the virtual entity. This poorer behavior can be explained by the fact that there was quite some noise in the position sensors of the physical system, which caused the driven distance of the truck to accumulate faster than the truck drove in reality. And as the waypoints supplied to the controller depend on the driven distance, this caused a bit of error, but still, the truck followed the path very well.

The only flaw of the controller is that it did not work in reverse. This is due to the fact that after the forward controller was working, the time of our team was taken up by integrating this into the overall project, such that a reverse controller could not be developed anymore.

8.3 Human Machine Interface (HMI)

A Human Machine Interface (HMI) has been designed using the Panel Diagram package in IBM Rhapsody to track the movement of the trucks from the Distribution Center entrance and parking, through the undocking and docking areas, until the trucks exit the DC. An HMI helps visualize the position and movement of the trucks within the boundary of the Distribution Center at both the Central Server level, as well as the drivers. At the central level, the HMI can help ensure smooth operation within the DC. At the driver level, it helps them monitor the status and battery level of their truck and also helps adjust various ergonomic factors within the truck cabin, such as lighting, HVAC, seat position, and various audio-visual properties inside the truck.

Figure 30 is one such Graphical User Interface (GUI) with the two main Control Panels of the INTELLION System. The first control panel labelled 'DC Control Panel' shows the HMI of the entire DC at a system level, which can be used to assign trucks to docks. This is illustrated with the two push buttons for assigning Dock 1 and Dock 2 to the trucks, whose status can be seen in the DC Panel Status Box. Similarly, the subsystem level HMI labelled 'Truck Control panel' shows the Truck level control panel. Here, the acceptance of a dock by a truck is illustrated, whose status can be seen in the Truck Panel Status Box in IBM Rhapsody. Further, the status, i.e., the position of each truck can be seen in the display box labelled 'STATUS OF THE TRUCK.' Finally, this entire HMI system can be started on demand, as the name suggests, by pressing the start truck push button.

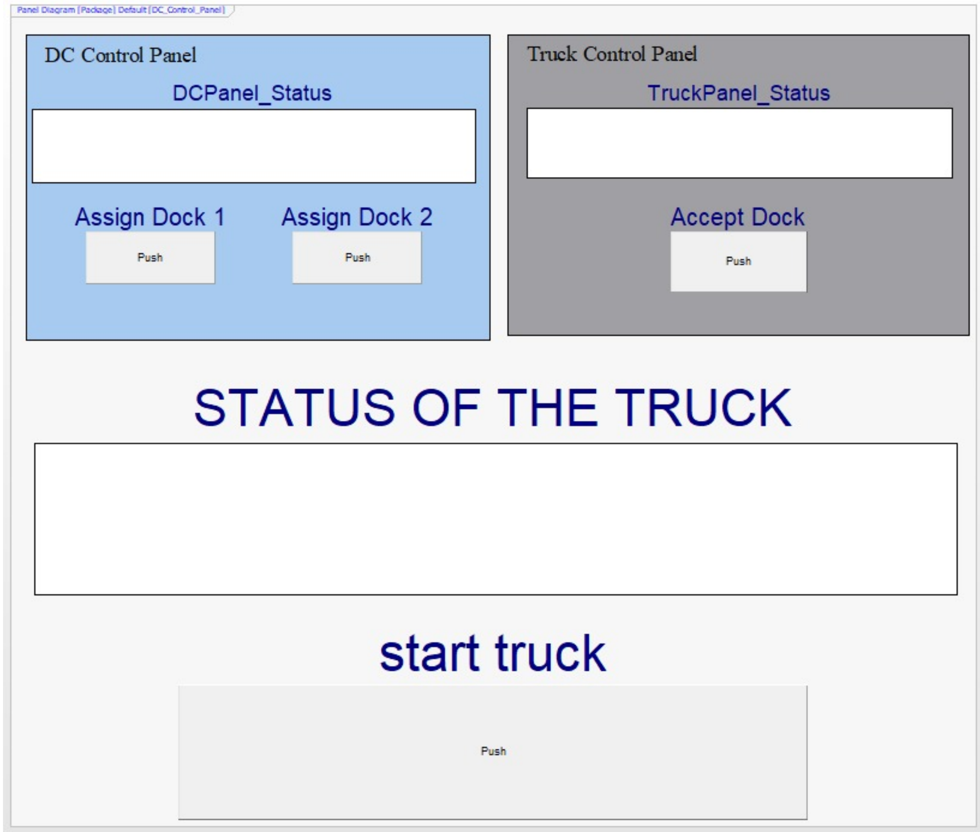


Figure 30: DC and Truck Control Panel HMI in IBM Rhapsody

Further, another HMI used in the project implementation is called Radar Mapping. This HMI has been designed in MATLAB and it is used to map the real-time position of the trucks with respect to one another.Figure 31 shows the illustration, with the option to select the truck on the right panel. As a result, the real-time position of the selected truck is shown on the radar map on the left, with respect to the truck of interest. Moreover, the MATLAB GUI also shows the Steering Angle of the truck.

In order to get the relative distance and position of the trucks, the concept of the resultant vector is used. With this, we first identify the distance of each truck from an origin, and then find the resultant distance between these two points. Finally, polar coordinates can be used to define the distance and angle between the trucks in the form of (r, θ), where 'r' stands for distance or magnitude, and θ stands for the angle or direction of the truck.

Figure 31: Radar Mapping HMI in MATLAB

8.4 System Architecture and Integration

For the Distribution Center to work seamlessly, many systems such as Sensing, Autonomous Trucks, Dock Assignment Control, Collision Detection and Avoidance, Network and Communication have to exchange data among each other. This involves the work of System Integration. The basis of System Integration is set of interfaces between different "Sub-Systems" to interact with each other. Current industries work with different Softwares to perform the System Integration. One has to keep in mind that System Integration is a method or a process and not a technology in itself. A "System Architect" has to think on a larger scale if he/she has to realise the project. The System Architect has to look into the Requirements and different Use Cases and build a map which can be visualized before actually working on the technology behind the project.

The simplest mapping can be in the form of a sketch or in technical terms an architecture. Figure 32 depicts the different systems involved in working of the Distribution Center along with the data and control flow between them. This architecture already provides the necessary information of what is supposed to be designed. The Architecture shows INTELLION being the Central Distribution Center Controller, communicating with all the available systems. The Application Logic of all the trucks provide the real time location and status through different on-board sensors to INTELLION. The Entry and Exit of the trucks are being stored and reported back to INTELLION. Similarly, the DOCKING STATIONS and PARKING STATIONS have the application logics respectively to sense the presence of a truck and notify accordingly. Considering the trucks which are autonomously driven, require complicated application logic which needs to deploy proper control over the steering and braking of the truck with respect to the received signals from various sensors such as Camera and LIDAR.

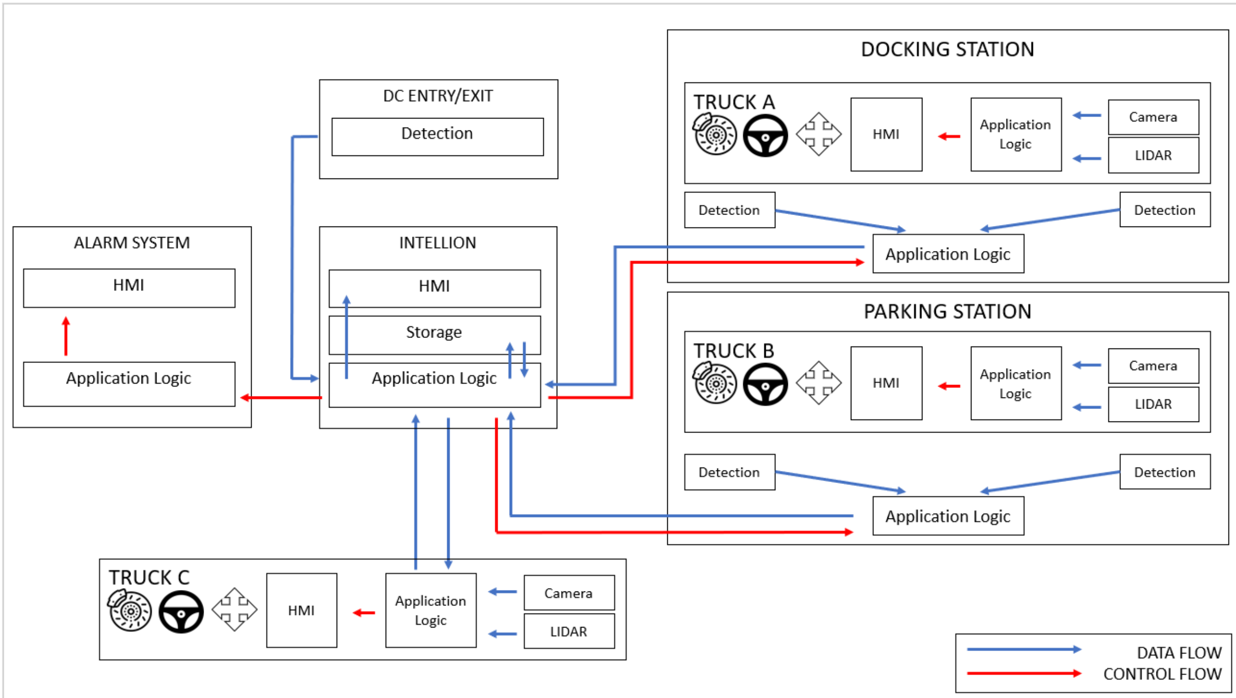


Figure 32: Detailed Architecture consisting data and control flow of different systems

The pictorial architecture shown in the above figure can be understood better by designing a working simulation. IBM Rhapsody is the tool used which has the features where any system can be designed using logics and display the working of the system using Lights, Switches, Scroll Bars, Textural Displays etc. Behavior of different functionality in the system can be realised using the Behaviour Diagrams from SysML Language such as State Machine Diagram, Sequence Diagram and Activity Diagram. These Behaviour diagrams provide the designers choice to make different assumptions, calculations, apply conditions and produce events. Figure 33 shows a Panel Diagram of a scaled-down version of Distribution Center with 4 Docks and 2 Trucks. The back-end logic is developed using State Machine Diagram which is shown in figure 34. The system designed can show the status of the trucks and the docks, assign specific docks to specific trucks and restrict the assignment of docks to the truck if its already in motion until it reaches back to the Parking Spot. This can be considered as a first stage of verification of the working of the system and the project.

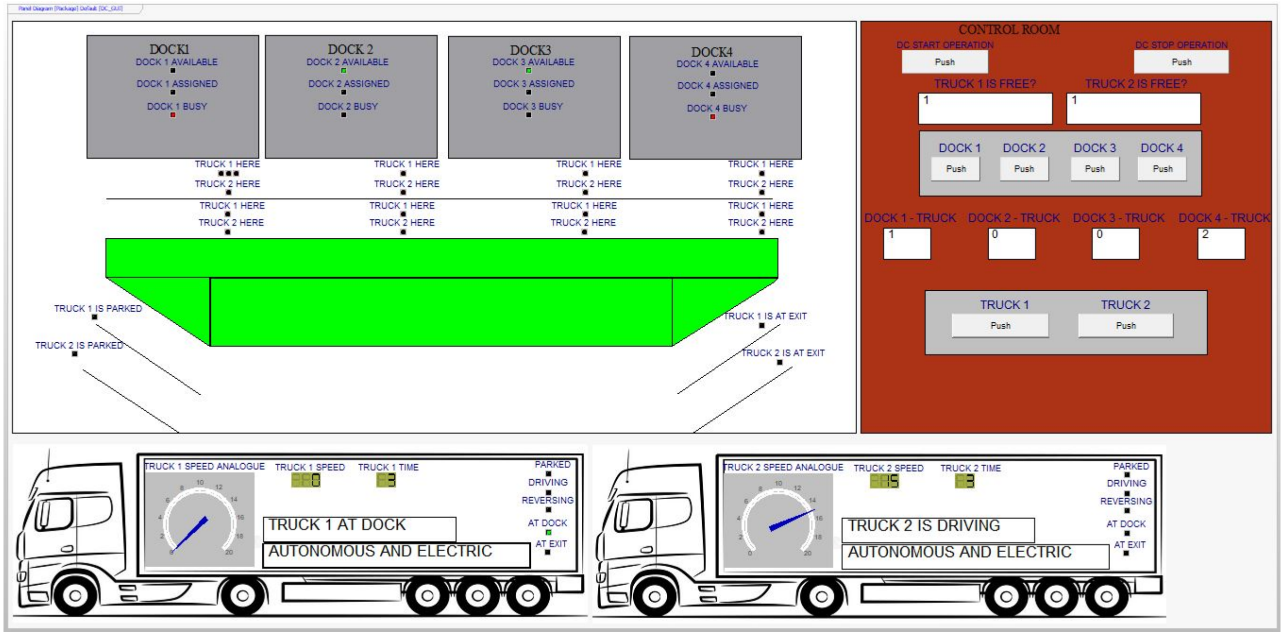


Figure 33: Logical Architecture of the Distribution Center with important functionality

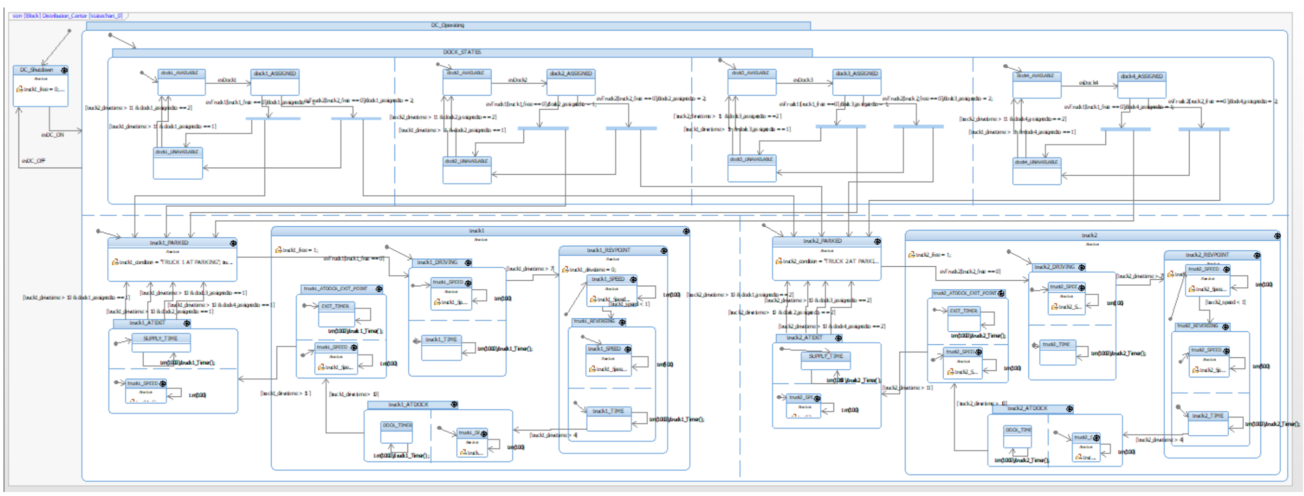


Figure 34: State Machine Diagram of working of Distribution Center

Once the team is satisfied with the Architecture, the System Architect has to plan the process of integration by considering the available work force and software and hardware components. The overview of softwares used for the project are shown in the figure 35. In the upcoming sections, the Interfaces are discussed to understand the work carried out in the project.

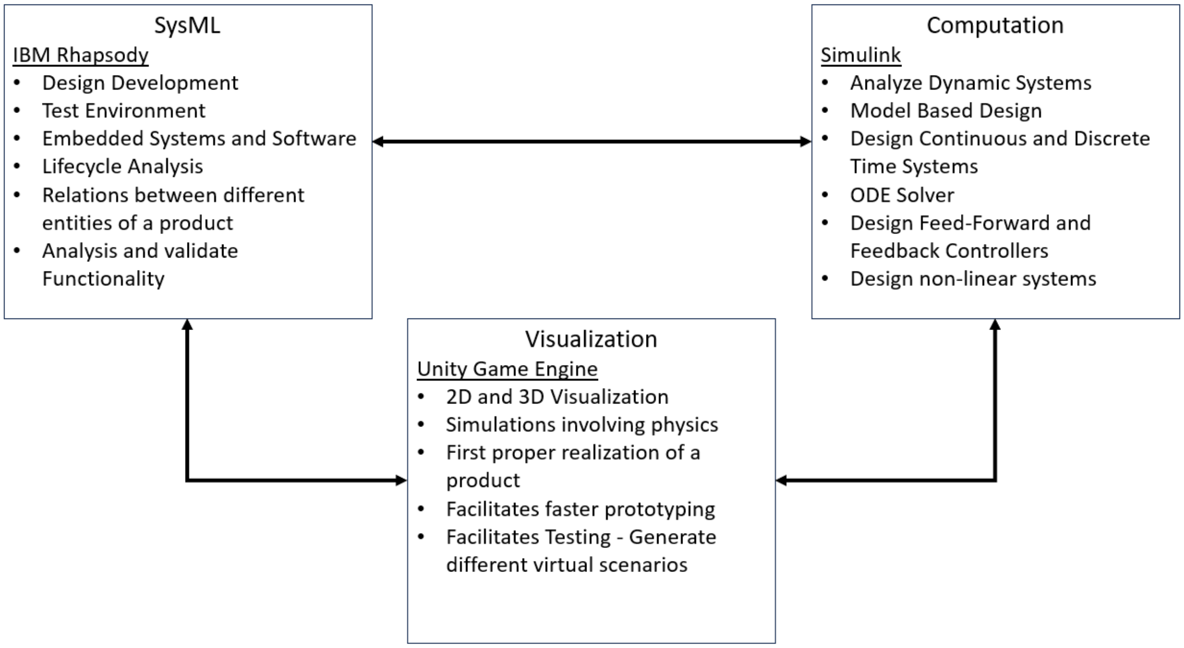


Figure 35: Overview of the Integration between different technologies used in the project

8.4.1 Interface 1: Simulink - Unity Game Engine

The motion controller is designed in Simulink along with the kinematic model and the virtual simulation consisting of the physics engine and Distribution Center Layout is designed in Unity Game Engine. Motion Controller is explained in detail in section 8.1. The two softwares are connected via ROS network. On the interface is established, the simulation in Simulink and in Unity Game Engine are started simultaneously and the movement of the truck can be visualized in Unity where the truck follows the path using the motion controller. A working model is provided as a video with this report. This is the second stage where the Architecture is realised. Figures 36 along with 25 show the snapshots of Unity Game Engine and Simulink respectively. This is the second stage in verification of the Motion Controller working as intended.

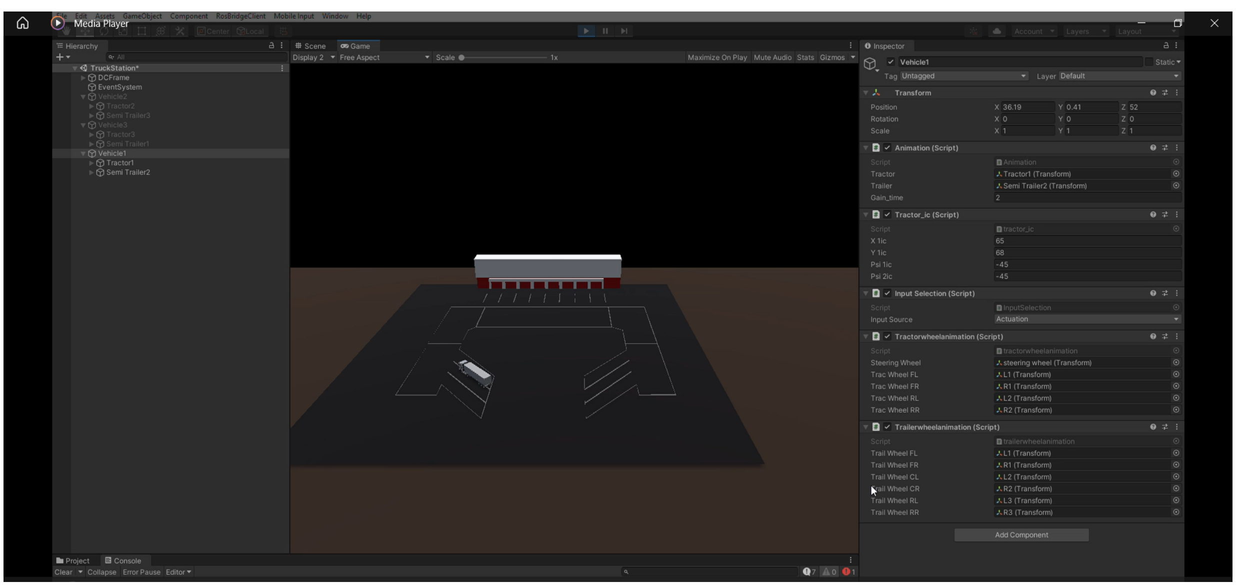


Figure 36: Snapshot of Unity Game Engine

8.4.2 Interface 2: Simulink - Physical Trucks

Once the Motion Controller is checked in Virtual World with different settings and scenario and it is confirmed that the Controller works correctly. Next stage (third) in verification of the system is connecting the controller to the physical trucks. TU Eindhoven has an Automotive Technology Lab (AT Lab) setup where certain area is dedicated to the scaled-down working of Distribution Center with Autonomous Trucks. This is descried clearly in Section 4.4. The motion controller designed in simulink is connected to the physical trucks with the help of ROS Network. The dynamics of Physical Truck matches the dynamics of the Kinematic model used to develop the motion controller, thus making sure the safe working of the physical trucks. Necessary changes in the simulink model such as Scaling of the truck and Velocity are done before deploying the algorithm into the Printed Circuit Board (PCB) present in the trucks. Figure 26 provides the snapshots from the working models. The videos are are provided with the report for the visualization of real time working of the physical trucks. The final stage will be connecting the Simulink, Virtual Trucks, Physical Trucks and IBM Rhapsody for functioning the whole system. This is described in the next section.

8.4.3 Interface 3: Simulink - IBM Rhapsody

The integration of the two fields is realized with SysML as the modeling language for the system's behavior. One of the main reason for using SysML is that it captures the many states and actions that occur during the design process. The simulation tool, on the other hand, aids in describing the system using mathematical expressions. The SysML field has multiple components, including the meta-model, requirements, use cases, system architecture, simulation execution profiles, and model libraries. These elements contribute to a complete representation of the system. Lastly, the simulation execution solver is critical in conducting the simulation on the given model. It is responsible for generating the result set depending on the input parameters provided, providing precise and trustworthy simulation outcomes.

The SysML paradigm is made up of blocks, input and output ports, and links that connect the blocks. A block is a modular unit that specifies the system's structure. The Internal Block Diagram (IBD) is a SysML diagram that depicts the internal structure of a block and its relationships with other components. Components within a block communicate via input and output ports, which include variables indicating various kinds of energy or information flow between components. Links, like SysML connectors, represent the connections that allow energy or data to be transported. They make it easier to describe the linkages between a block's components.

The simulation model, on the other hand, includes mathematical expressions or equations, functions, blocks, and connections, among other things. When the SysML model is translated onto the simulation tool, the tool will present a structure that roughly mimics the system's intended design in the SysML model.

The integration of multiple software sectors is used to illustrate the usability and flexibility of these softwares and the solution that they can provide. This enables the development of an system which is both interactive and also efficiently utilizes the full capability of both the software tools and the underlying technology. In the framework of the 4AT100 project, IBM Rhapsody is used as the Distribution Center's Control Panel. It is responsible for variety of functions, such as assigning docks to trucks, monitoring the number of trucks in the Distribution Center, and initiating system shutdown in the event of an emergency or present mistake. Simulink, on the other hand, works as a counterpart and offers the control algorithm for the trucks' autonomous movement.

The IBM Rhapsody Control Panel can be developed using a set of built-in diagrams such as the Block Definition Diagram (BDD), Internal Block Diagram (IBD), and State Machine Diagram (SMD). The "Structured Simulink Block" is selected as the main platform on which the system is designed within the BDD. The "Simulink Block" depicts one of the Simulink file's blocks. Figure 37 depicts a visual representation of the BDD created for the project. Once the BDD has been finished and the ports have been defined, the internal connections are built in the IBD while the stateflow is designed in the SMD. Figure 38 depicts a thorough representation of the connections and the Simulink file.

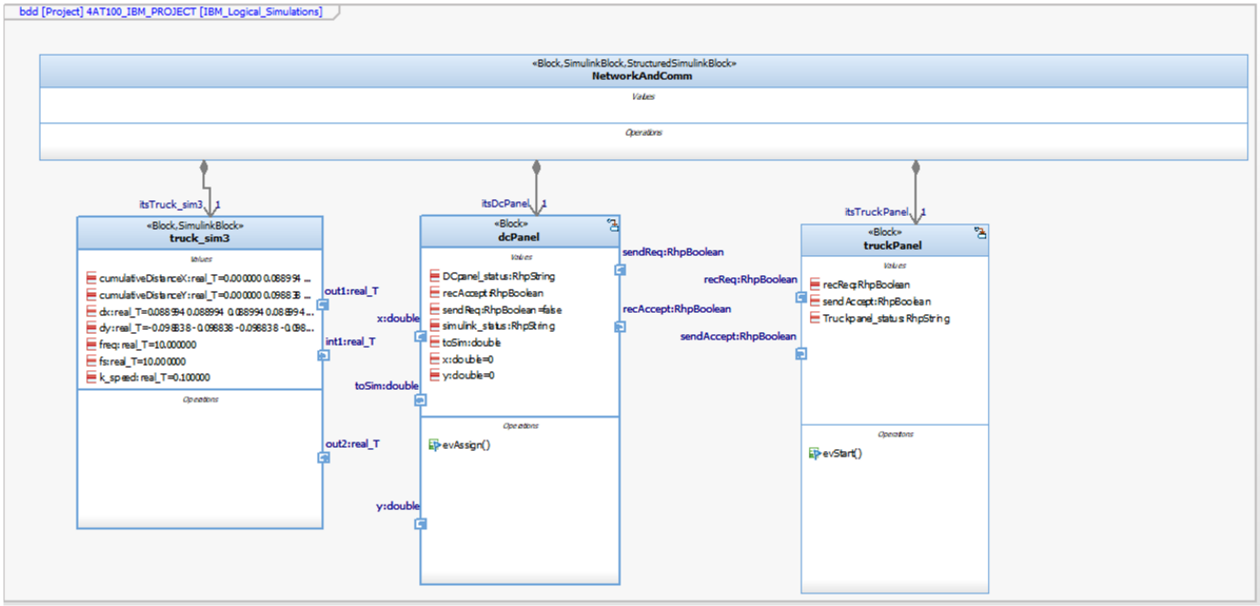


Figure 37: Block Definition Diagram for 4At100 Project

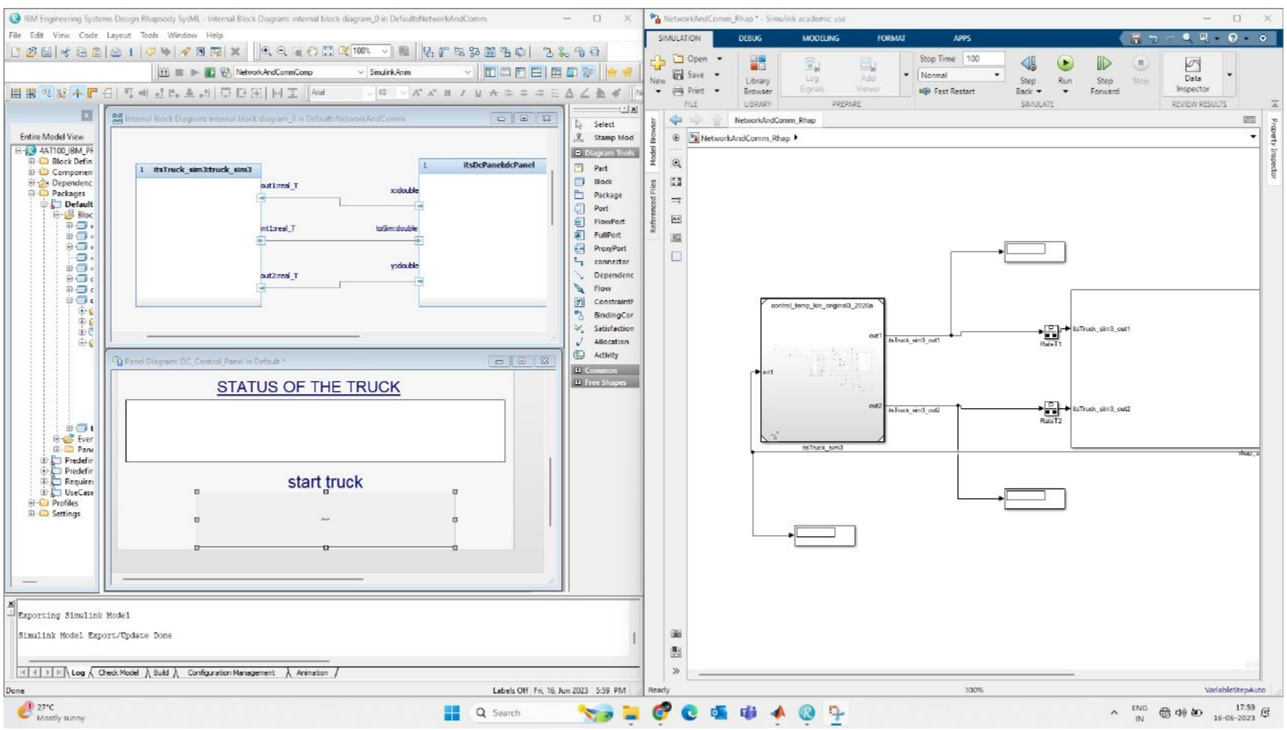


Figure 38: Integration of IBM Rhapsody and Simulink for 4AT100 Project

9 Project results

The objective of the 4AT100 Project was to investigate the possibility of implementing autonomous driving in a closed environment (Distribution Center) by designing a system which integrates and communicates between the Ware House, the Trucks and the Premises. Previous sections described how the algorithm for the motion controller, the HMI and the Integration were designed and implemented in 4AT100 Project. Firstly the working Motion Controller is tested in virtual space to check the 3-dimensional working of the motion controller and to make sure no unintended movements occur. Figure 39 depicts the screenshots of the truck starting on three different positions on the Distribution Center. The working of the motion controller is shown completely in the video provided with the report.

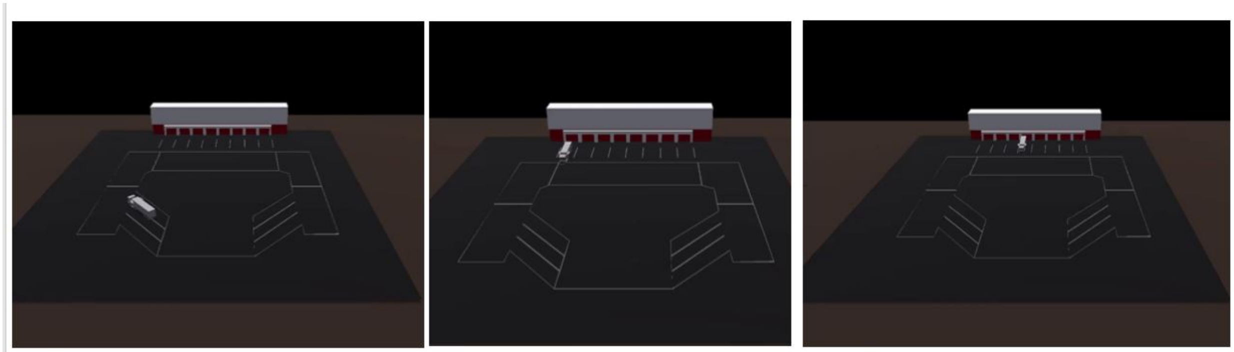


Figure 39: Pictures of Virtual Testing of motion controller in Unity Game Engine

Once the virtual testing is satisfied, next step was to implement the motion controller in the physical mode which is a scaled down model of an actual truck along with the trailer. Figure 40 shows the snapshots of the physical truck driving around the Distribution Center modelled inside the AT-Lab. THe complete working of the Physical Truck is provided as a video with the report.

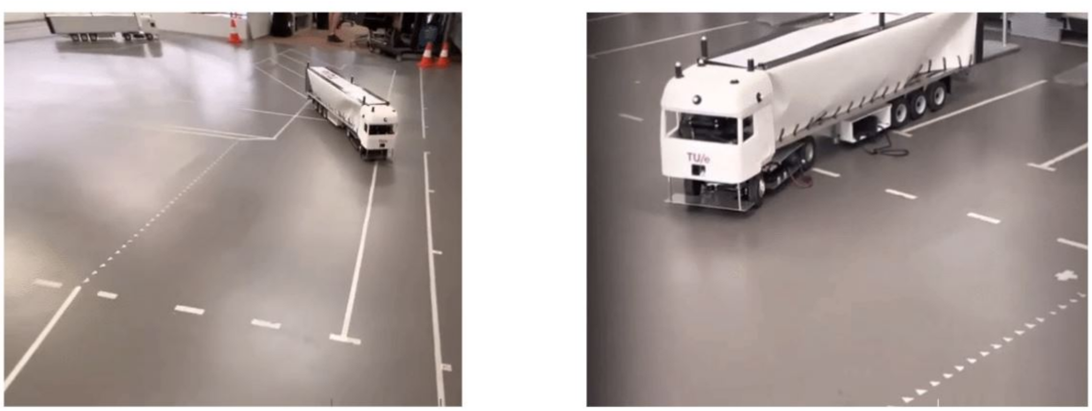


Figure 40: Snapshot of the physical truck driving in the AT Lab

Both the virtual and physical testings were in accordance with the requirements that were laid out in the beginning of the project. The forward control from our team was used to realise the big picture of multiple teams working together to get one truck to complete the whole route from parking area to the exit area. Also discussed the possibility of multiple trucks working in coordination. Thus a complete system engineering approach was realised through this project.

10 Conclusion

The implementation of autonomous trucks in a distribution center shows promising advancements in transportation. Enhancing safety, reducing pollutant emissions and tackle personnel shortages are some of the many benefits of automated trucks. However, the advancements face many challenges. From technical developments to moral acceptance.

This project aims to contribute to the development of a Digital Twin system for automated truck driving. By starting from a pre-developed framework, the team implemented motion controllers, human machine interfaces and communication formats to build further on the framework. Besides this, a detailed systems engineering approach is followed in quartile 3 which consisted of understanding the problem, creating a solution and working out all relevant documentation to create a strong basis. Within this period, the group focused on TRIZ for generating creative solutions, project and system context to define the scope and linked externals and user stories to capture the user's needs and desires. To convert this information to useful engineering terms, the requirements were generated and captured in use cases. Moreover, to ensure safety of the system, functional safety engineering was applied according to the ISO 26262 standard. A functional safety assessment was made using ASILs. This qualifies whether a use case is safety-critical or not.

The project continued in Q4 where the implementation phase started. This phase concerned the development of a motion controller, HMI dashboards, a Matlab-IBM integration and a common software structure for communication between trucks. Using a virtual kinematic model, the virtual entity and physical entity of the digital twin, the motion controller was created. The HMI dashboards were created using the IBM Rhapsody tool whereas a common communication platform was created using Matlab Simulink. Moreover, each group member individually focused on a certain research topic.

In the end, the group was able to create a working Pure Pursuit forward controller which was validated using a three-tiered simulation process. The HMI dashboard including real-time radar mapping also showed promising results. Moreover, the successful integration of Matlab and IBM Rhapsody enabled the group to create a communication between operator input and truck controller. Lastly, a common software structure was successfully created by defining common in- and outputs for communication between trucks.

The group is happy about the achievements and concludes the project to be a success in many ways. In terms of knowledge, experience, contributions to the vision of autonomous driving in distribution centers and more.

Appendices

A TRIZ

The following are the different project domains and their corresponding TRIZ parameters numbers (refer Figure 4) within the brackets. These parameters play a crucial role in the design and development process in each of these domains, as explained below.

1. Virtual entity [12, 18, 27, 28, 29, 33, 34, 36, 39]:

   The problem with current generation Physical Prototypes is that they are expensive, with respect to both Time and Money. Prototyping is an iterative process where the Industries or Manufacturers build models to test the feasibility and compliance with the requirements and aesthetics of the product. As the number of these prototypes increases substantially, it becomes time-consuming and economically burdensome. Implementation of Virtual Prototyping can be a viable solution to this problem. By using CAD Models, engineers can visualize a product in a 3-Dimensional view, which can further be applied to building an entire system and visualising it before implementing it in the real world.
   The virtual entity being considered is the Digital Part of the Distribution Center. By developing a digital system, the objective of the project can be visualised in advance in order to make the necessary changes. In this context, TRIZ Parameters such as Shape and Brightness are considered, since they can be utilized to create a simulated environment. The simulated environment examines the measurement and manufacturing aspects of the system. By introducing the Digital Twin, the prototyping and testing of the system can be done Virtually and check for all possible scenarios. This in turn increases the Productivity of building the Physical System and also facilitates better reliability.

2. Automotive Human Factors (HMI) [17, 18, 19, 24, 27, 29, 34, 39]:

   Safety and comfort are a few important aspects when it comes to Trucks. The long distances that the drivers are required to travel, along with the need for rest periods within the trucks are some of the issues that require the Truck Cabin to be designed to ensure a comfortable and user-friendly environment.
   For designing the HMI of a truck, TRIZ Parameters like Temperature, and Brightness are taken into consideration. Nevertheless, there is always a compromise between Comfort and Energy Consumption. It is essential to find the balance between them since the design has to be efficient but also minimize energy consumption. Another important factor is the reliability and consistency of the information provided by the Human Machine Interface (HMI). Finally, a well-built cabin can increase the driver's productivity while ensuring their comfort during longer journeys, while also providing the opportunity to rest whenever possible.

3. Centralized Control (Path Planning, and Communication) [2, 3, 4, 6, 9, 15, 20, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]:

   The ability to reach a desired target and to avoid surrounding obstacles in the environment are crucial aspects of autonomous vehicles, especially in complex and dynamic environments like a distribution center. A robust path-planning algorithm has to be implemented, which generates a sequence of points and segments that enable autonomous trucks to navigate safely through the working area toward the destination while considering collision avoidance with the surrounding environment. Moreover, the path planning algorithm should also optimize other goals like minimizing the travelled distance, energy consumed, and the time to reach the destination.
   On the other hand, Cooperative Autonomous Maneuvering (CAM) enables the truck to communicate and cooperate with other trucks, sensors, and the infrastructure to achieve an efficient and safe operation. By combining all the features mentioned above in the contradiction matrix, a path-planning algorithm can be developed that takes into consideration various factors such as the weight, length, area, speed of the truck, the presence of surrounding objects, and the duration of action by moving objects.
   Additionally, TRIZ parameters such as the capacity and reliability of the truck are also included, as

the algorithm must generate a path that optimizes the transport of goods and minimizes the risk of breakdowns or failures. The accuracy and precision of the truck's sensors and components must be taken into account to ensure safe and efficient navigation. Furthermore, the algorithm must optimize the energy usage of the truck and consider the energy consumption of stationary objects in the working area when generating the path.

4. Smart Planning [16, 25, 27, 28, 29, 31, 33, 35, 36, 39]:

   Smart planning in autonomous distribution trucks involves considering various TRIZ parameters to optimize the use of trucks and reduce idle time. By analyzing data, ensuring accurate and efficient navigation, and adapting to changing circumstances, a smart planning system can improve efficiency, reliability, and productivity in the distribution process. To reduce idle time and improve efficiency in autonomous distribution trucks, a smart planning approach is to strategically assign the trucks to suitable docking/undocking stations. The use of charging points at these stations can also be optimized, as the trucks often remain stationary for over an hour, to enable the vehicles to be charged during this time. By ensuring the system is easy to operate, drivers can focus on monitoring the truck's condition and ensuring safe and efficient navigation. By incorporating real-time data and analytics, a smart planning system can adjust the distribution process according to changing demands. Ultimately, the goal of smart planning is to improve productivity, reduce costs, and enhance customer satisfaction by optimizing the use of trucks and reducing idle time.

5. Smart trucks (Sensing and Communication) [2, 3, 4, 5, 6, 9, 15, 18, 24, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39]:

   For a truck to be identified as being a smart truck, multiple features must be implemented compared to conventional trucks. A smart truck should perform or assist driving, communicate with other vehicles and/or infrastructure, and other technologically advanced functionalities. The addition of features increases the complexity of a truck, further enhancing the existing system. When applying the TRIZ methodology to this domain, quite a lot of engineering parameters are relevant to smart trucks. This domain focuses in particular on the sensing and communication part of the smart trucks.
   Firstly, since there is a dependency on sensors and communication devices, a particular selection of TRIZ parameters should be selected for sure. This selection of parameters consists of Brightness, Loss of information, Reliability, Accuracy of measurements, and Accuracy of manufacturing, to include awareness and dependency on the boundaries and limitations of sensors and communication devices in the development process. The second aspect of the sensing and communication domain regards general information about the vehicle in order to process incoming data from the sensors and communication devices. In order for a smart truck to make a certain manoeuvre, for example, it needs to know its length, width speed and weight. Therefore, the TRIZ parameters Weight of non-moving objects, Length, Area and Speed are of importance.
   Other relevant parameters, like Level of automation, Complexity of device and control and Harmful side effects need to be included due to the features the truck needs to provide/is assumed to provide. Subjects like complexity, safety and system usage are subdomains that flow out of the conceptual idea of a smart truck's sensing and communication. Based on these subdomains, the relevant aforementioned TRIZ parameters were selected.

6. Driver-less smart docking [2, 3, 4, 5, 6, 9, 16, 27, 28, 29, 32, 33, 36, 38]:

   One of the subjects to focus on when increasing a DC's automation is driver-less smart docking. This automation principle is closely related to the sensing and communication domain of smart trucks. Therefore, multiple TRIZ parameters are overlapping between these domains. This selection of parameters represents the dependency on truck dynamics, sensors and communication devices. Therefore, TRIZ parameters Weight, Length, Area, Speed, Measurement accuracy and Manufacturing accuracy were selected.
   Another important component of driver-less smart docking is the dock itself. In order to achieve high reliability and minimize risks, the current docks might not be suitable for auto-docking. To include this, the TRIZ parameters Area, Durability of the non-moving objects, Reliability and Manufacturability are important.

The system complexity and usage of driver-less smart docking require other TRIZ parameters like the Complexity of the device and the level of automation.

7. Powertrain [1, 9, 15, 19, 21, 22, 27, 29, 32]:

   The powertrain whether it consists of a conventional internal combustion engine, a hybrid engine or an electric one, helps deliver power from the engine to move the wheels of the vehicle. Modelling a powertrain that meets the requirements of the user, the whole vehicle and the system takes into consideration various parameters. A few of these TRIZ parameters include the force, speed, power, ease of manufacturing, use of energy and weight of the truck.
   The truck, including the powertrain, plays a major role in the considered system (smart distribution centers). It is designed in a way to not only be efficient and reliable but also reduce energy consumption, in turn reducing the carbon footprint of the vehicle. A few ways to achieve this is by using alternative fuels in a conventional engine or changing the engine to be hybrid or fully electric.

8. Safety [1, 2, 3, 4, 5, 6, 9, 10, 15, 24, 29, 31, 35]:

   The safety of any system depends on a number of factors. In this system, many of the major functions like path planning, collision avoidance and driver-less smart docking all depend on the aspect of the system safety. Various TRIZ parameters like the weight of the truck, dimensions of the truck, area, speed, force, loss of information, manufacturing precision, adaptability, and the duration of action by the moving objects are all taken into account.
   A system is said to be safe when it can adapt or respond positively to any expected or unexpected external changes. This also involves the occurrence of little to no errors or accidents caused by the objects in the system. There should always be a fail-safe for any unpredictable incidents that the system may encounter to make sure that safety is not compromised.

9. Infrastructure [2, 3, 5, 9, 18, 20, 29, 32]:

   The infrastructure within the distribution center includes the parking area of the trucks, loading and unloading docks, storage areas, buildings within the area of the distribution center and also the autonomous driving trucks. The infrastructure might also include virtual components like the Cooperative Autonomous Maneuvering (CAM) Server and the communication between different entities.
   In conclusion, various TRIZ parameters such as weight and dimensions of the trucks, speed of the trucks, light intensity, and energy consumption within the distribution centre are considered for this domain. The construction and planning of the distribution center are key factors in increasing efficiency while reducing the time and energy consumed, all while reducing the total costs.

B Requirements

B.1 Business Requirements

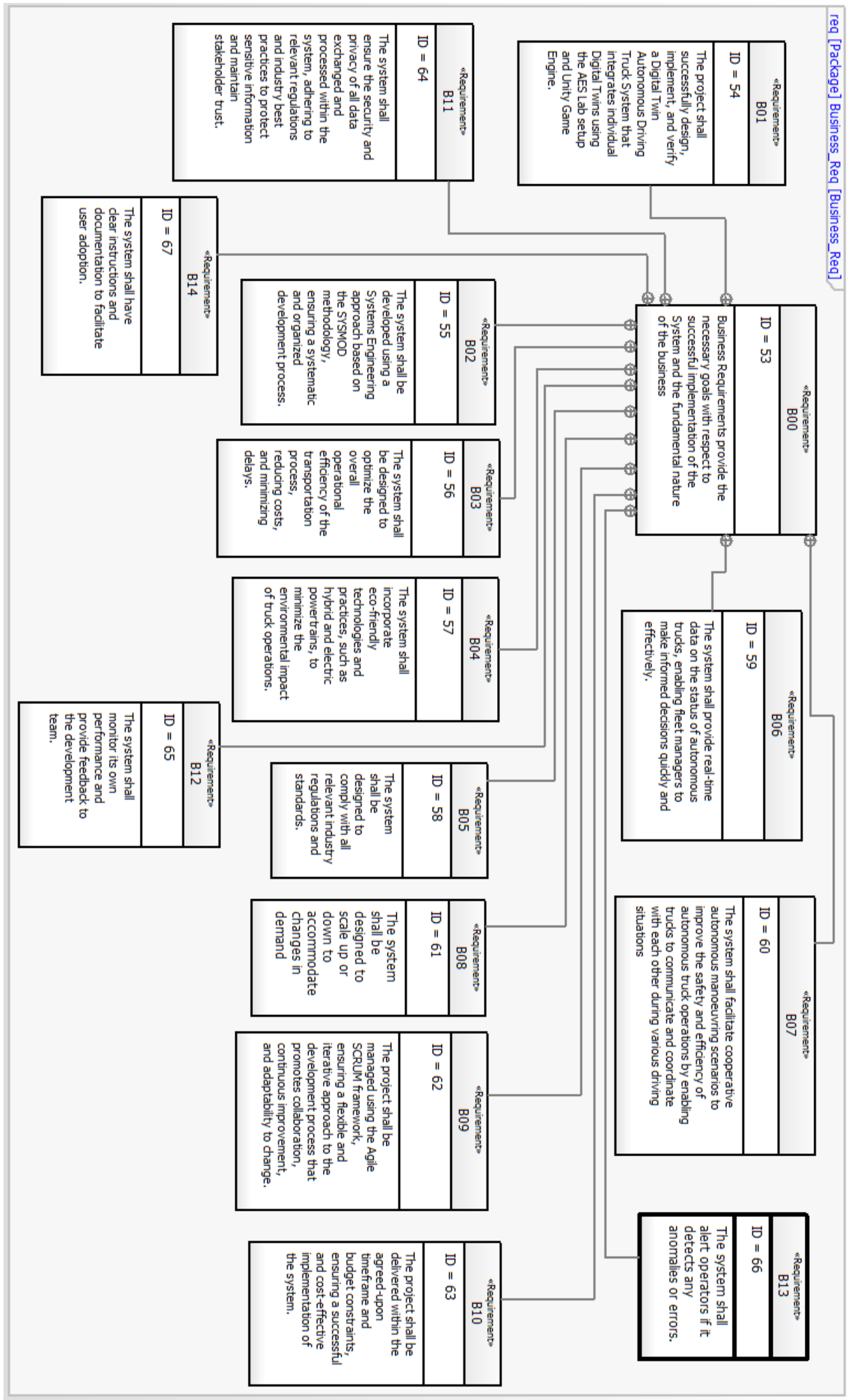


Figure 41: Business Requirements Diagram

B.2 General Requirements

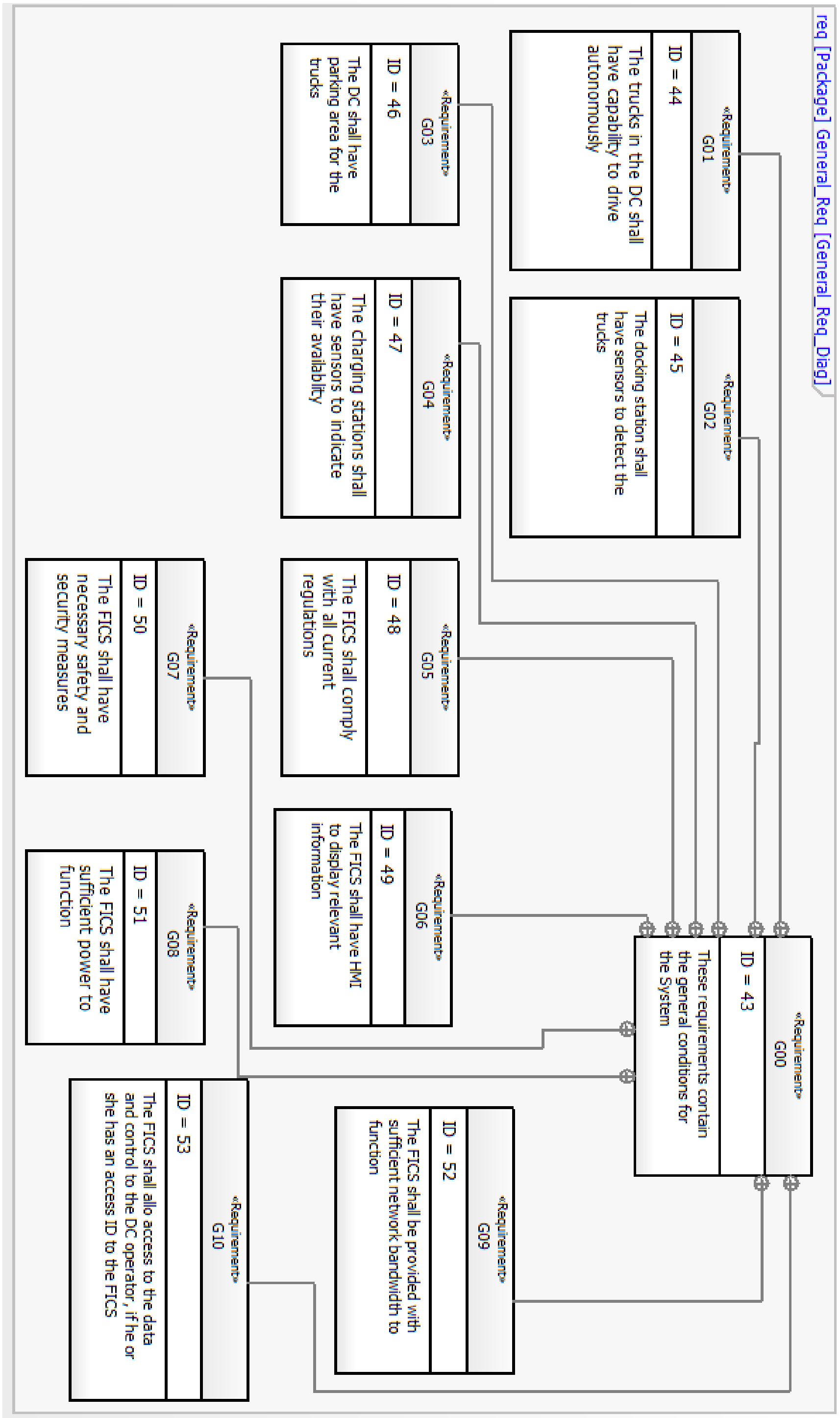


Figure 42: General Requirements Diagram

B.2.1 Requirements - Network and Communication

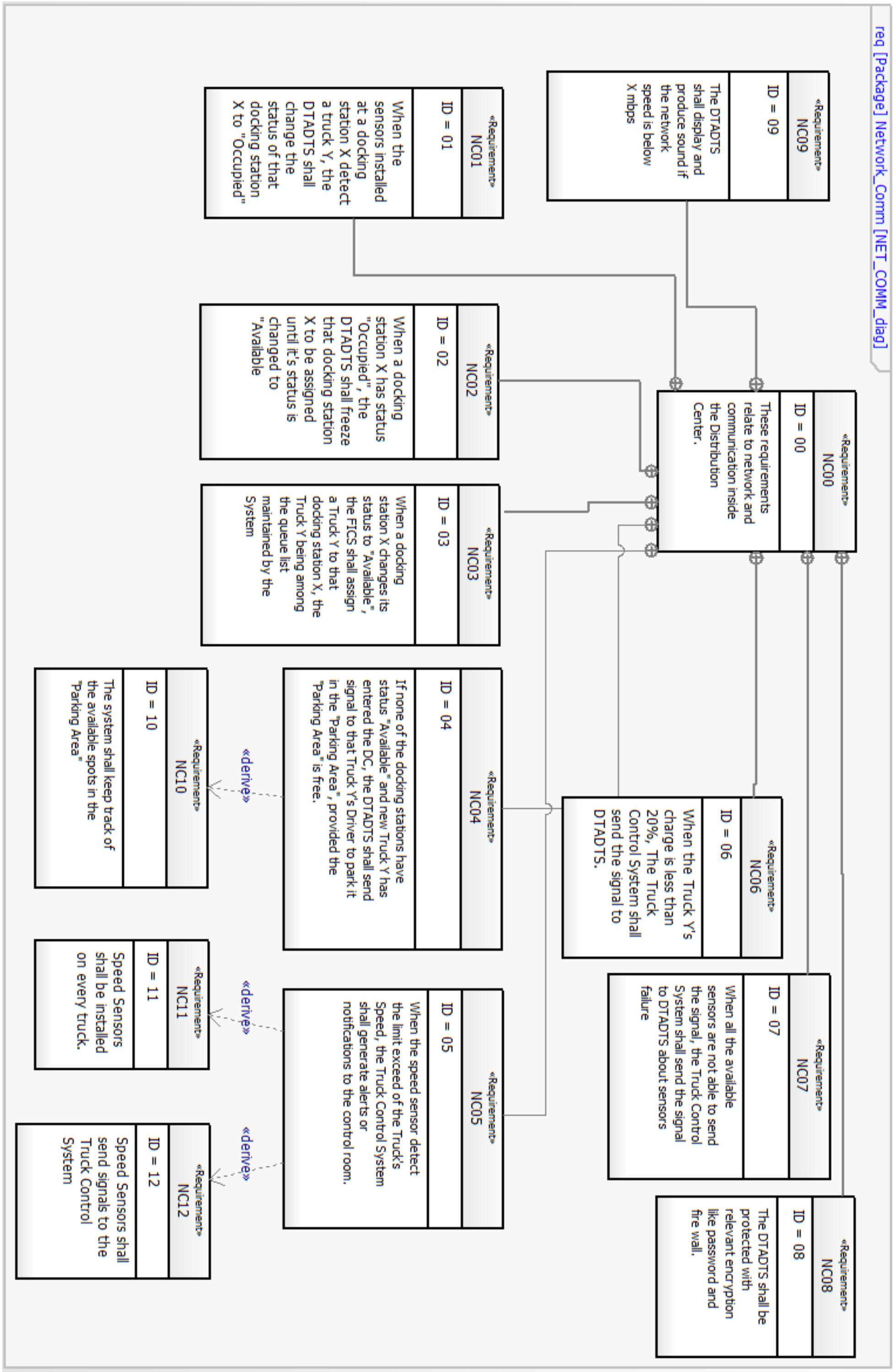


Figure 43: Network and Communication Requirements Diagram

B.2.2 Requirements - Safety

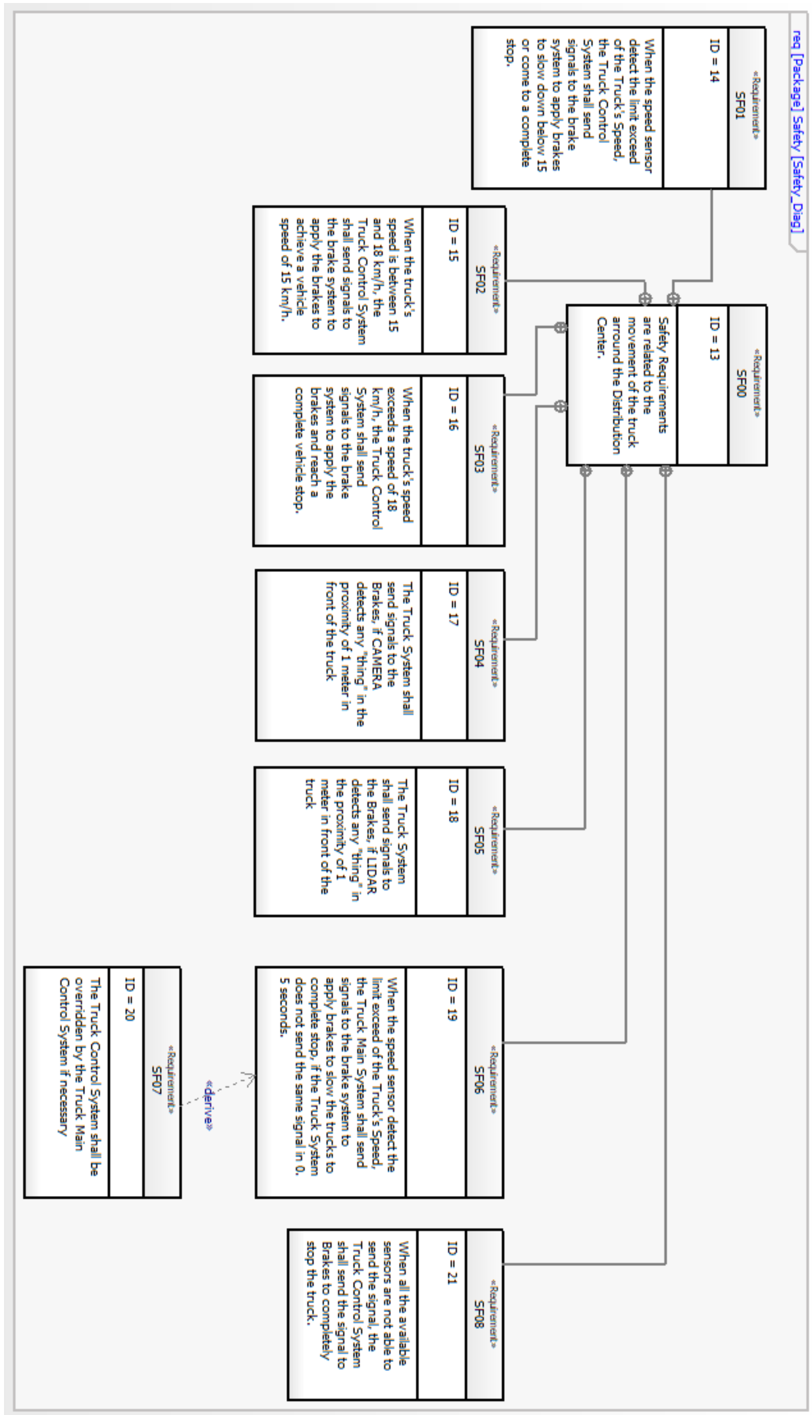


Figure 44: Safety Requirements Diagram

B.2.3 Requirements - Vehicle Dynamics

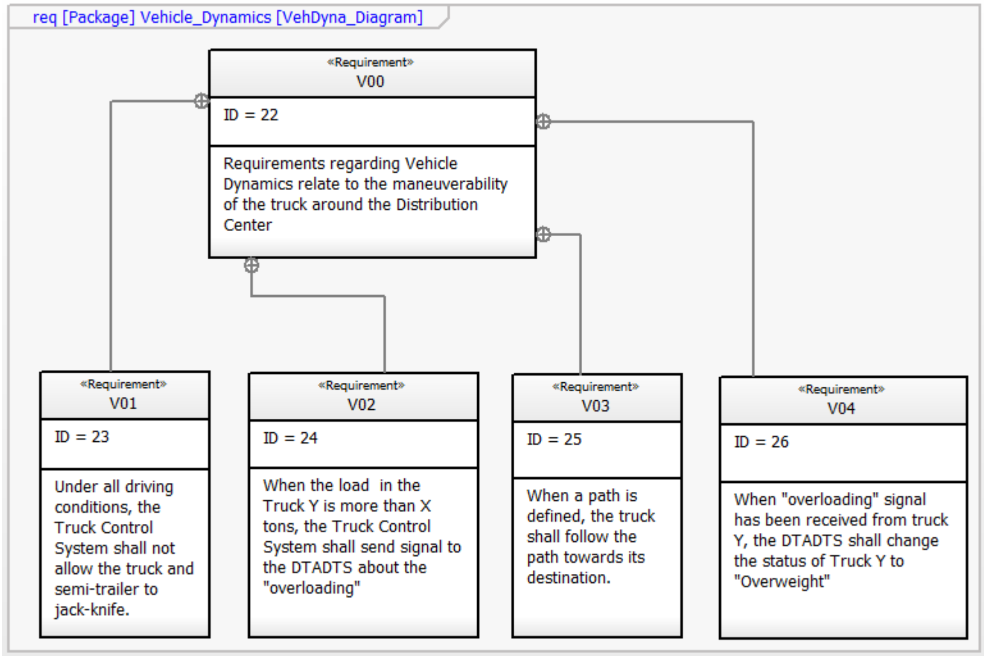


Figure 45: Vehicle Dynamics Requirements Diagram

B.2.4 Requirements - Sensing

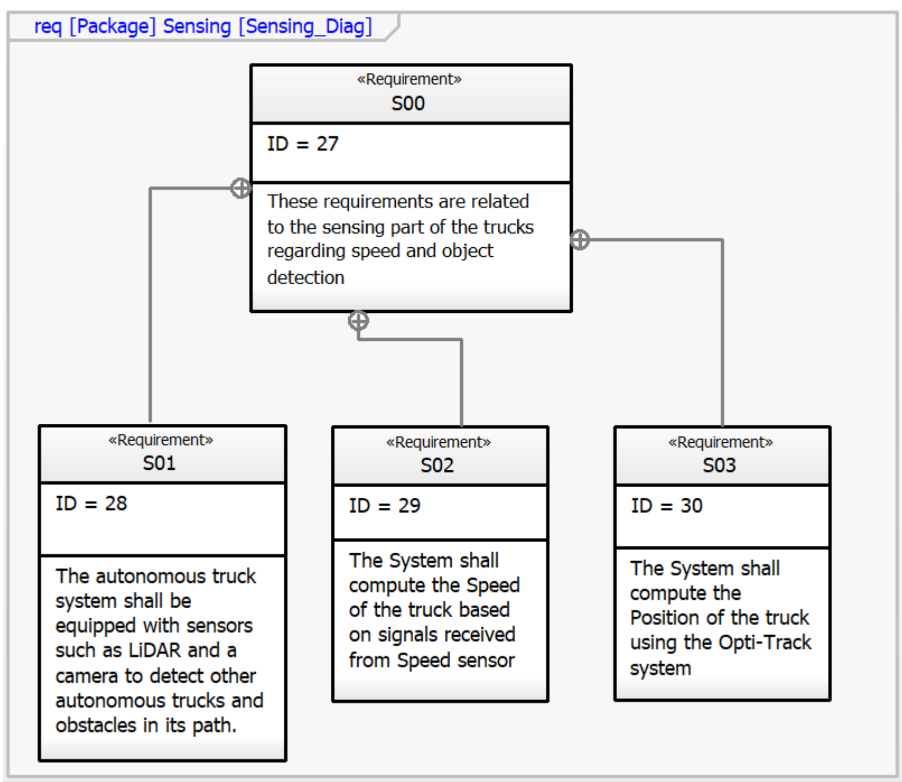


Figure 46: Sensing Requirements Diagram

B.2.5 Requirements - Powertrains

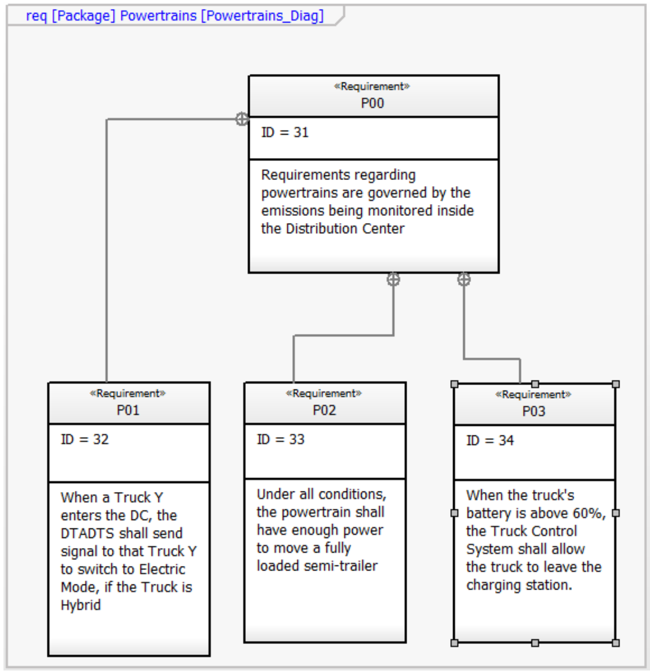


Figure 47: Powertrain Requirements Diagram

B.2.6 Requirements - Path Planning

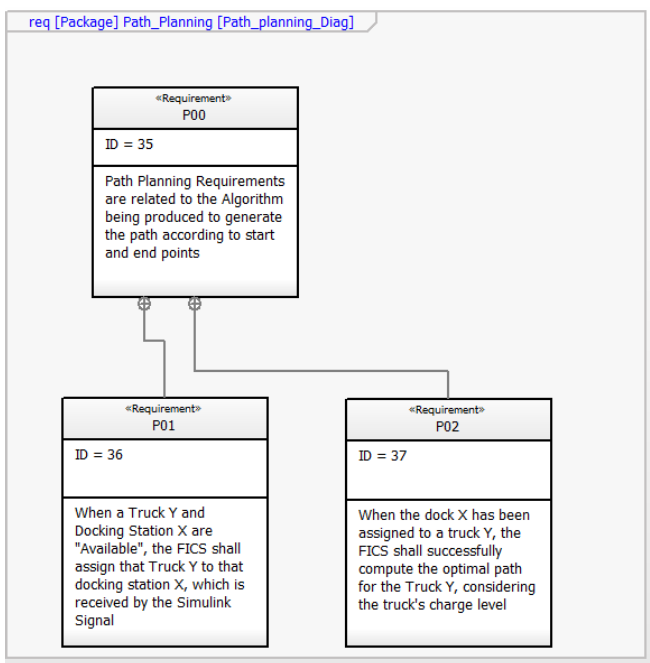


Figure 48: Path Planning Requirements Diagram

B.2.7 Requirements - HMI

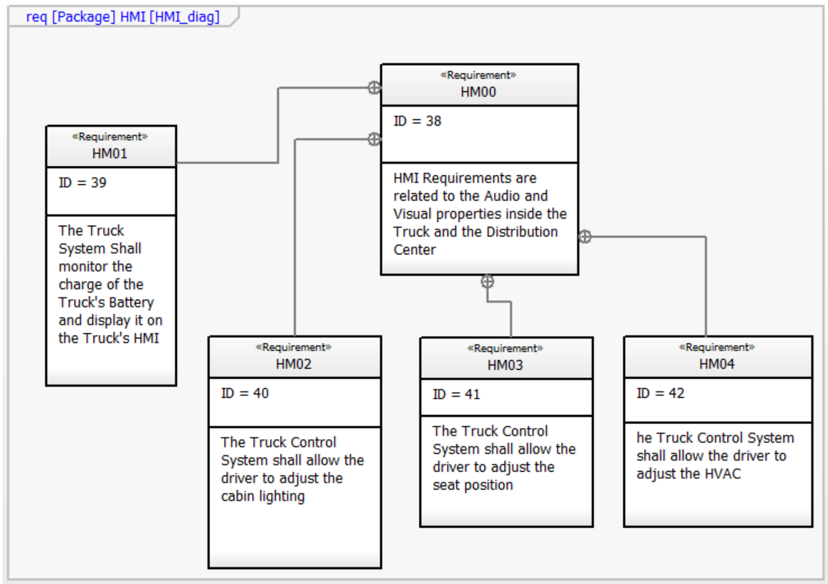


Figure 49: HMI Requirements Diagram

C Use cases

C.1 Safety Rating

The safety rating of each use case is shown in Table 5. The table contains ratings for the functional safety elements Severity, Exposure and Controllability, denoted by S, E and C respectively. Based on the combined ratings for S, E and C, there is a combined safety rating and a provided motivation. The ratings are based on the criteria shown in Figure 11.

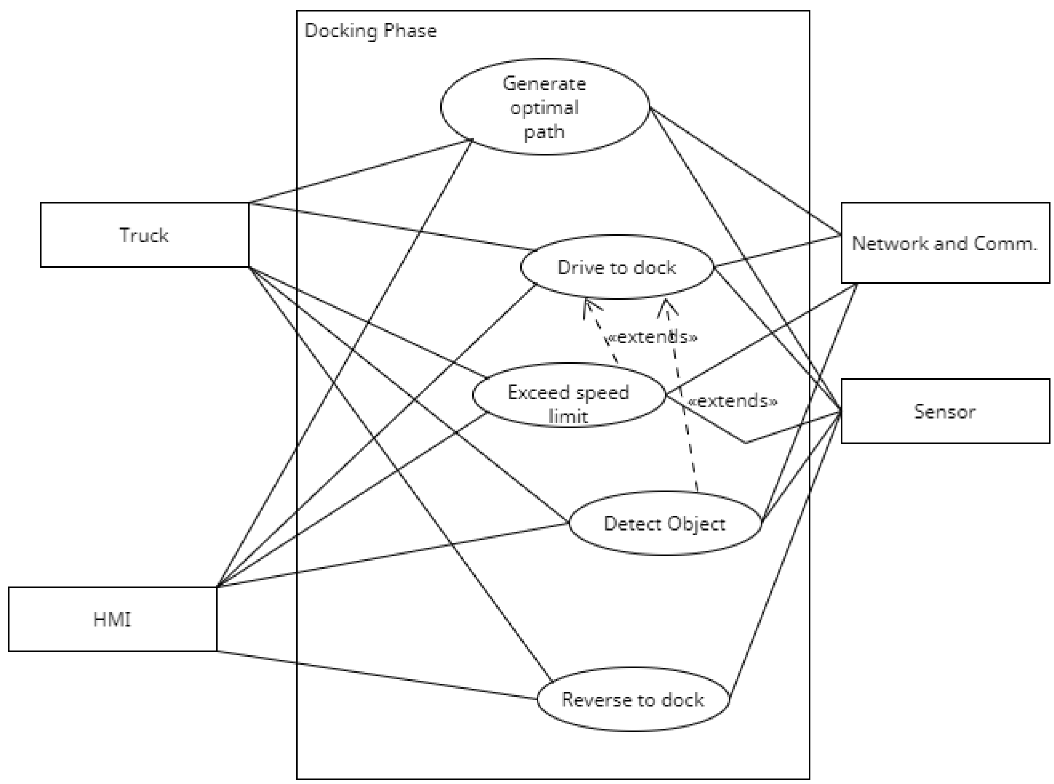


Figure 50: Use case diagram: Docking phase

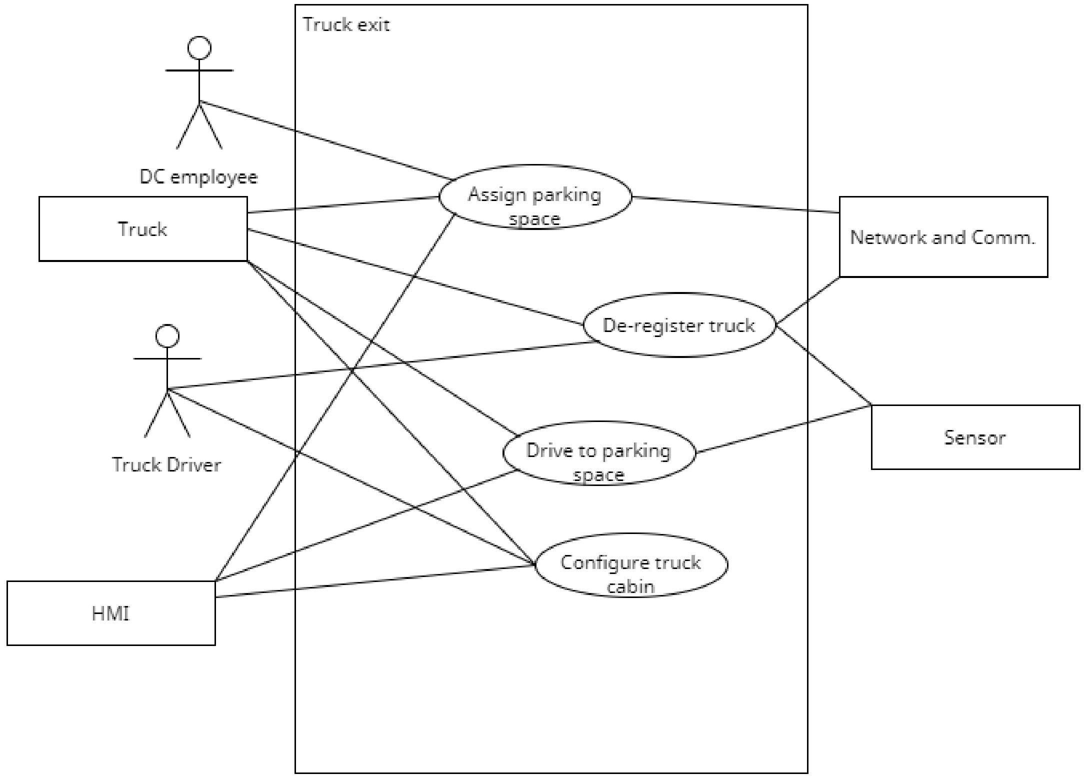


Figure 51: Use case diagram: Truck exit phase

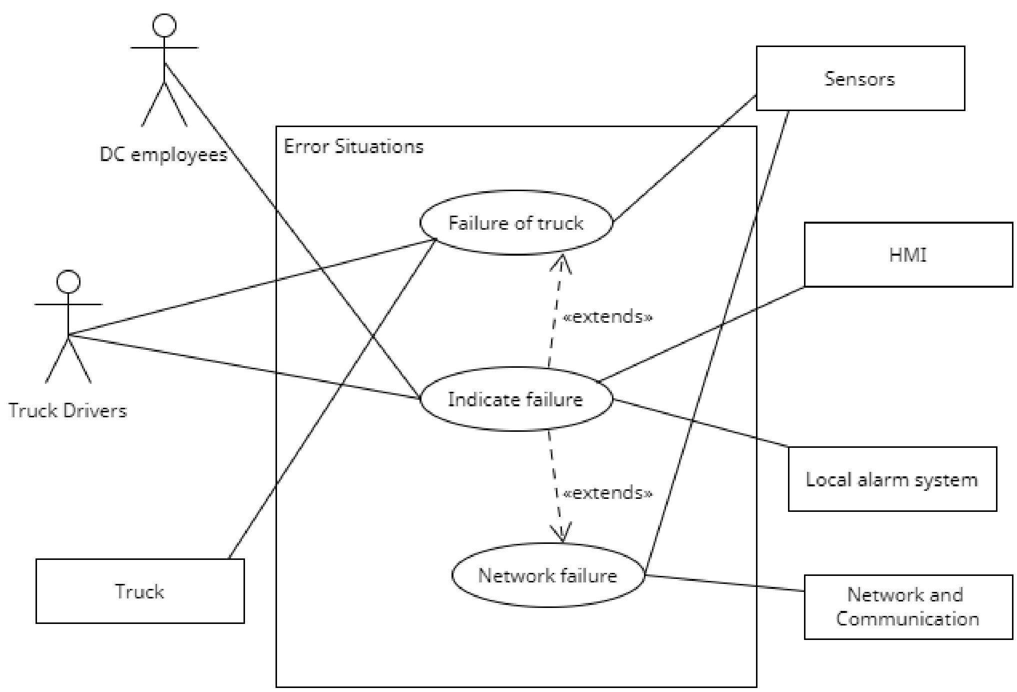


Figure 52: Use case diagram: Error situations

Table 4: Use case descriptions

no. 1	**Use Case Name**	**Register the truck**
	Short description	The INTELLION establishes a connection with the incoming truck
	Primary Actors	Truck, Truck Driver
	Secondary Actors	Network and Communication
	Preconditions	Truck enters the DC area
	Normal flow	1. The INTELLION sends the request signal to the truck 2. Truck receives and accepts the request 3. INTELLION adds truck to database 4. INTELLION sends an acknowledgement back to the truck 5. The truck receives the acknowledgement
	Postconditions	There is a connection between INTELLION and truck
	Error situations	1. INTELLION does not receive the request signal 2. The truck does not receive the acknowledgement
	Alternative flow(s)	1.1 The truck driver initiates the request in the truck HMI 1.2 The INTELLION receives the request 1.3 Go to 3
no. 2	**Use Case Name**	**Park the truck**
	Short description	The truck driver drives the truck from DC entrance to an available parking space
	Primary Actors	Truck, Truck driver
	Secondary Actors	Sensors, Network and Communication, HMI
	Preconditions	Truck is registered to the INTELLION
	Normal flow	1. Driver drives to parking space 2. HMI displays the truck's battery level 3. Truck driver plugs the truck into a charger 4. Driver leaves the truck 5. Parking spot becomes unavailable
	Postconditions	Truck is parked
	Error situations	
	Alternative flow(s)	1.1 Truck switches to electric mode (if hybrid)
no. 3	**Use Case Name**	**Check dock availability**
	Short description	INTELLION senses the presence of the trucks and determines dock availability
	Primary Actors	Truck, Sensors
	Secondary Actors	Network and Communication, HMI
	Preconditions	none
	Normal flow	1. Present truck leaves dock X 2. Sensor detects the truck that's leaving the dock 3. INTELLION processes the sensor signal 4. INTELLION changes the status of dock X to 'available' 5. HMI displays that dock X is available
	Postconditions	The dock status of the docks is up to date
	Error situations	1. INTELLION does not receive the signal of the sensor 2. HMI does not update accordingly
	Alternative flow(s)	1.1 DC employee changes the status of Dock X by manually pressing a button from Dock X 2.1 **Network failure**

no. 4	**Use Case Name**	**Assign dock**
	Short description	Assign a truck to a dock
	Actors	Truck, Network and Communication
	Secondary Actors	HMI
	Preconditions	Truck A is available at the parking area
	Normal flow	1. INTELLION checks the dock ID of the available dock X 2. INTELLION sends the available dock ID to truck A 3. INTELLION changes the status of dock X to 'occupied' 4. HMI displays that dock X is 'occupied'
	Postconditions	Truck A is assigned to dock X
	Error situations	1. No dock is available 2. The truck does not receive the assignment signal
	Alternative flow(s)	1.1 INTELLION rechecks the availability of other docks docks 1.2 Goto 2
no. 5	**Use Case Name**	**Generate optimal path**
	Short description	INTELLION generates the optimal path for a truck
	Actors	Truck
	Secondary Actors	Network and Communications, HMI
	Preconditions	The start and end point of the path is known
	Normal flow	1. INTELLION considers the start and end point of the path 2. INTELLION runs the optimization algorithm 3. INTELLION checks the charge level of the truck 4. INTELLION sends the optimal path to the truck 5. HMI displays the path the truck will drive
	Postconditions	The truck has received its optimal path to drive
	Error situations	1. No optimal path is found 2. The charge level of the truck is below 20%
	Alternative flow(s)	1.1 Goto 1 2.1 HMI displays that the battery level is too low 2.2 Truck driver checks if the Truck is connected to a charger
no. 6	**Use Case Name**	**Drive to dock**
	Short description	Drive truck A to dock X
	Actors	Truck, Sensors
	Secondary Actors	Network and Communications, HMI
	Preconditions	The truck has received its optimal path to drive
	Normal flow	1. Truck checks if all driving conditions are met 2. Truck drives the optimal path 3. Truck reaches the destination 4. Truck sends a confirmation to INTELLION 5. HMI shows that the truck successfully reached its destination
	Postconditions	The truck is positioned at the dock
	Error situations	1. Not all driving conditions are met 2. The sensors detect an object in front of the truck 3. the speed limit is exceeded
	Alternative flow(s)	1.1 HMI displays that the truck is unable to drive 1.2 Goto 1 2.1 **Detect object** 3.1 **Exceed speed limit**

no. 7	**Use Case Name**	**Reverse to dock**
	Short description	Truck A reverses into dock X
	Actors	Truck, Sensors
	Secondary Actors	Network and Communications, HMI
	Preconditions	The truck is driving to dock X
	Normal flow	1. Truck manoeuvres into the dock 2. Dock sensor detects the presence of truck A
	Postconditions	The truck is positioned at the dock
	Error situations	1. The sensors detect an object in front of the truck 2. the speed limit is exceeded
	Alternative flow(s)	1.1 **Detect object** 2.1 **Exceed speed limit**
no. 8	**Use Case Name**	**Exceed speed limit**
	Short description	Truck A exceeds its speed limit
	Actors	Truck, Sensors
	Secondary Actors	
	Preconditions	The truck is driving
	Normal flow	1. Sensor detects over-speed of the truck 2. Truck decelerates 3. Sensor checks if the correct speed is maintained
	Postconditions	The truck drives at a safe speed
	Error situations	
	Alternative flow(s)	
no. 9	**Use Case Name**	**Detect object**
	Short description	The path of a truck is adapted to obstacles
	Actors	Truck, Sensors
	Secondary Actors	Network and Communications, HMI
	Preconditions	The truck is driving
	Normal flow	1. Sensor detects an object in front of the truck 2. Truck decelerates 3. HMI displays that an object is detected 4. **Generate optimal path** 5. Truck continues to drive the new optimal path
	Postconditions	The truck avoids the obstacle
	Error situations	1. The optimal path cannot be generated
	Alternative flow(s)	1.1 Truck stops 1.2 Truck sends error message to INTELLION

no. 10	**Use Case Name**	**Assign parking space**
	Short description	Parking space X is assigned to truck A
	Actors	Truck, Network and Communications
	Secondary Actors	HMI, Sensors
	Preconditions	The truck is at the dock
	Normal flow	1. INTELLION checks if truck A is finished loading/unloading 2. INTELLION checks which parking spot is available 3. **Generate optimal path** 4. INTELLION sends optimal path to truck A
	Postconditions	The truck received the optimal path to drive
	Error situations	1. Truck A is not finished loading/unloading 2. No parking spot is available
	Alternative flow(s)	1.1 Wait for truck A to finish loading/unloading 1.2 Go to 2 2.1 Wait for a parking spot to become available 2.2 Go to 3

no. 11	**Use Case Name**	**Drive to parking space**
	Short description	Truck A drives to an assigned parking space
	Actors	Truck, Sensors
	Secondary Actors	Network and Communications, HMI
	Preconditions	The truck has received its optimal path to drive to the parking space
	Normal flow	1. Truck checks if all driving conditions are met 2. Truck drives the optimal path to the parking space 3. Truck arrives at the assigned parking space 4. Truck sends a confirmation to INTELLION 5. HMI shows that the truck successfully reached its assigned parking space
	Postconditions	The truck is positioned in the parking space
	Error situations	1. Not all driving conditions are met 2. The sensors detect an object in front of the truck 3. The speed limit is exceeded
	Alternative flow(s)	1.1 HMI displays that the truck is unable to drive 1.2 Goto 1 2.1 Detect object 2.2 HMI alerts the driver about the detected object 2.3 Truck stops or slows down to avoid the object 2.4 Generate new optimal path (if necessary) 2.5 Goto 1 3.1 Exceed the speed limit 3.2 HMI alerts the driver about the exceeded speed limit 3.3 Truck slows down to the allowed speed limit 3.4 Go to 1

no. 12	**Use Case Name**	**Configure Truck cabin**
	Short description	Driver configures the truck cabin for their comfort
	Actors	Truck, Truck Driver, Environment
	Secondary Actors	HMI
	Preconditions	Driver is in the truck cabin
	Normal flow	1. The driver checks the seat position 2. The driver checks the cabin lighting 3. The driver checks the HVAC temperature
	Postconditions	The driver checks the above settings for his comfort
	Error situations	
	Alternative flow(s)	1.1 The driver changes seat position according to his comfort 2.1 The driver changes cabin lighting accordingly 3.1 The driver changes the HVAC temperature according to the ambient temperature and his comfort

no. 13	**Use Case Name**	**Deregister truck**
	Short description	Deregistering a truck from the system while exiting the DC
	Actors	Truck, Truck Driver
	Secondary Actors	Network and Communication, Sensor
	Preconditions	The truck is exiting the DC
	Normal flow	1. Truck driver sends the request signal to INTELLION 2. INTELLION receives and accepts the request 3. INTELLION deregisters truck from database 4. INTELLION sends an acknowledgement back to the truck 5. The truck driver receives the acknowledgement
	Postconditions	There is a connection between INTELLION and the truck
	Error situations	1. INTELLION does not receive the request signal 2. The truck driver does not receive the acknowledgement
	Alternative flow(s)	1.1 INTELLION initiates the request to the truck driver for exiting 1.2 The Truck driver receives the request 1.3 Go to 3

no. 14	**Use Case Name**	**Failure of truck**
	Short description	A truck fails to execute an intended action or performs unintended actions
	Actors	Truck
	Secondary Actors	Truck driver, Sensor
	Preconditions	The truck has entered the DC area
	Normal flow	1. The truck is performing driving actions 2. The truck fails to perform its intended action 3. The sensors located on the truck detect the failure 4. The truck enters safe mode 5. The truck applies the brakes and comes to a complete stop 6. The truck indicates the failure to the truck HMI 7. The truck indicates the failure to INTELLION
	Postconditions	The truck has come to a complete stop
	Error situations	1. The sensors fail to detect failure 2. The truck fails to enter safe mode 3. The truck fails to come to a complete stop
	Alternative flow(s)	1.1 The truck performs unintended actions 1.2 The sensors detect unintended truck behaviour 1.3 Go to 4

no. 15	**Use Case Name**	**Failure of network**
	Short description	The network is unable to perform intended actions
	Actors	Truck, Sensors
	Secondary Actors	Network and Communications, HMI
	Preconditions	The truck is driving and connected to the network
	Normal flow	1. Network failure is detected by the truck or INTELLION 2. HMI alerts the truck driver about the network failure 3. Truck safely slows down and stops 4. Truck sends a status update to INTELLION 5. INTELLION logs the network failure incident
	Postconditions	The truck is stopped safely due to network failure
	Error situations	1. HMI fails to alert the truck driver about the network failure 2. The truck is unable to send a status update to INTELLION
	Alternative flow(s)	1.1 Truck driver notices the network failure through other means (unresponsive HMI, loss of GPS signal) 1.2 Go to step 3
no. 16	**Use Case Name**	**Indicate failure**
	Short description	Indicate to the truck and INTELLION in the case of a failure
	Actors	HMI, Network and Communication, Local Alarm system, Sensors
	Secondary Actors	DC Employees, Truck, Truck Driver
	Preconditions	There is a connection between INTELLION and the truck
	Normal flow	1. Failure is detected by the truck or INTELLION 2. CAM server alerts the truck and INTELLION about the failure 3. Audio and Visual signal sent to the truck and the system 4. The truck and the system accept the signal 5. The truck slows down to work on the failure aspect 6. The truck sends an update to INTELLION
	Postconditions	The failure is fixed
	Error situations	1. HMI fails to alert the truck driver about the network failure 2. The truck is unable to send a status update to INTELLION
	Alternative flow(s)	1.1 INTELLION or the truck notices the failure and sends a signal to the other 1.2 Go to 5

Table 5: Funtional safety ratings

Use Case Name	S	E	C	Motivation	ASIL rating
Register the truck	S0	E4	C1	1. S0 - No damage done if registration fails. 2. E4 - Every incoming truck is registered. 3. C1 - Simple exchange of signals for registration.	QM
Park the truck	S2	E4	C2	1. S2 - Possibility of collision with other trucks. 2. E4 - Frequent parking of trucks before and after docking. 3. C2 - Moderate complexity in executing parking procedure.	B
Check dock availability	S0	E4	C2	1. S0 - No damage done if dock availability check fails. 2. E4 - Frequent checking of availability of docks. 3. C2 - Simple process to check dock availability.	QM
Assign dock	S0	E4	C2	1. S0 - No damage done if dock assignment fails. 2. E4 - Frequent assignment of the docks to the trucks. 3. C2 - Moderate complexity in dock assignment.	QM
Generate Optimal Path	S0	E4	C2	1. S0 - No damage done if dock assignment fails. 2. E4 - Frequent assignment of the docks to the trucks. 3. C2 - Moderate complexity in dock assignment.	QM
Drive to dock	S3	E4	C2	1. S3 - Significant damage when failure 2. E4 - Every incoming truck will drive to a dock. 3. C2 - Moderate complexity to drive a specified path.	C
Reverse to dock	S2	E4	C2	1. S2 - Moderate damage potential if reversing to dock fails. 2. E4 - Reversing has to be done whenever the truck reaches the dock. 3. C2 - Moderate complexity for truck reversing.	B
Exceed speed limit	S3	E1	C1	1. S3 - Major damage potential if speed limit exceeds. 2. E1 - Probability of over-speeding is very low. 3. C1 - Simple process to detect and control speed.	QM
Detect object	S2	E4	C2	1. S2 - Possible injury to DC employee and/or truck. 2. E4 - Frequent object detection to prevent collisions. 3. C2 - Actuation of brake or new path generation.	B
Assign parking space	S0	E4	C2	1. S0 - No damage done if park assignment fails. 2. E4 - Frequent assignment of the parking space to the trucks. 3. C2 - Moderate complexity in parking space assignment.	QM
Drive to parking space	S3	E4	C2	1. S3 - Significant damage when failure. 2. E4 - Every truck exiting the dock will drive to a parking space 3. C2 - Moderate complexity to drive a specified path.	C

Configure truck cabin	S0	E2	C0	1. S0 - No damage is done if going wrong. 2. E2 - Configuration is often used for a long period. 3. C0 - Highly controllable.	QM
De-register truck	S0	E4	C1	1. S0 - No damage done if de-registration fails. 2. E4- Every outgoing truck is de-registered. 3. C1 - Simple exchange of signals for deregistration.	QM
Failure of Truck	S2	E1	C2	1. S2 - Severe damage to the Person, Goods or Trucks. 2. E1 - Very low probability of truck system error. 3. C2 - Actuation of brakes to stop the truck.	QM
Failure of Network	S2	E1	C3	1. S2 - Major damage potential if the system crashes. 2. E1 - Very low probability of network failure. 3. C3 - High complexity in resolving network errors.	QM
Indicate failure	S3	E1	C3	1. S3 - Possible damages if Indication fails. 2. E1- Very low possibility of Indication failure. 3. C3 - High complexity in resolving the issue.	A

D Individual papers

Path-Following Control for Autonomous Trucks: A Pure Pursuit Approach for Autonomous Maneuvering

Adwaith Gopichand

Fig. 1: TruckLab Environment in the TU/e Automotive Lab.

***Abstract*–This paper presents an in-depth exploration of a novel pure pursuit path-following control strategy for autonomous trucks. The primary aim is to enhance autonomous manoeuvring in confined environments, such as distribution centres. In particular, this study has two main contributions. Firstly, a detailed examination of the pure pursuit controller is presented, emphasizing its working principles and a comprehensive geometric interpretation of the algorithm. We also establish a kinematic model that provides a more comprehensive understanding of the vehicle's dynamics. Lastly, we conduct simulations and real-world experiments to validate the efficacy of our proposed control strategy. The results show a significant improvement in path-following precision and stability, suggesting that our pure pursuit control algorithm has significant potential in real-world autonomous truck applications. Thus, this study underpins the potential of the pure pursuit approach in enhancing the reliability and safety of autonomous truck navigation, paving the way for further research in this area.**

***Index Terms*–Autonomous vehicles, Path-tracking, Lateral controller, Pure Pursuit algorithm, Distribution centre**

I. Introduction

As the dynamics of the automotive and transportation industries continue to evolve, a paradigm shift is taking place. What was once an industry driven solely by mechanical systems has now been merged into electrical systems, software, and mechatronics, paving the way for sophisticated and highly integrated solutions. The ongoing multidisciplinary evolution is driving a significant transition, leading us into an era dominated by autonomous systems. Autonomous systems redefine the driving experience and greatly enhance safety, efficiency, and sustainability. As such, they have garnered substantial interest from leading companies such as Waymo, Amazon, and Uber, who are pioneering the development and public testing of autonomous vehicle technologies [3]. However, the application of autonomous systems is not only confined to public roads. The increasing necessity for sustainable and efficient logistics and freight transport systems has triggered significant research interest in Autonomous Driving Trucks (ADTs), especially in controlled environments like distribution centres.

In contrast to public roads riddled with intricate technical and legal challenges, logistic warehouses present a more suitable environment for adopting autonomous systems [9]. They provide an optimal setting, free from the unpredictable elements typically encountered on public roads. The controlled, low-speed operations within the distribution centre, coupled with a well-defined layout and short stopping distances, make it an optimal setting for the safe and efficient deployment of autonomous vehicles.

The main challenge with autonomous vehicles is guaranteeing that the vehicle can accurately track and follow a preset course, executing the movement strategy devised in the planning module. This is essentially the objective of lateral control, which must choose the necessary steering angle to correct any mounting errors and adapt to any changes in the direction of the path as they emerge [5]. In order to construct the lateral controller, the discrepancy between the actual vehicle location and the optimal path coordinates needs to be identified. Subsequently, choosing a control design strategy that mitigates errors to the slightest possible degree and duly adheres to the steering angle limitations becomes essential.

There are two categories of lateral control design. The initial category comprises geometric controllers. These controllers utilize the principles of geometry, working with the specified path coordinates and the vehicle's kinematic models. Notable examples of geometric controllers include the Pure Pursuit and Stanley controllers, which leverage these principles to establish vehicle control [1]. The second category, dynamic controllers, adopts a more advanced approach. The model predictive controller (MPC), an example in this category, employs finite horizon optimization to ascertain the most effective control command, considering the prevailing errors and potential future disturbances, offering a compre-

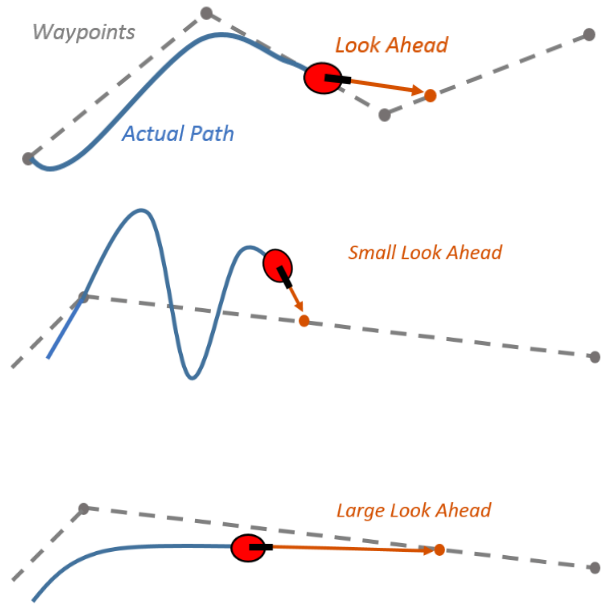


Fig. 2: Effect of varying look ahead distance on pure pursuit path tracking.

hensive and anticipatory control strategy [6].

This paper will focus on implementing the pure pursuit controller as an effective method of lateral control for Autonomous Driving Trucks (ADTs) in a distribution centre. Furthermore, it is essential to highlight that the development of this controller was done within TruckLab, which is a part of the TU/e Automotive lab, and is shown in Fig. 1.

Related Literature: The foundation of pure pursuit controller is grounded in the study of geometry and vehicle kinematics. An early example is the work by Coulter (1992), who introduced the Pure Pursuit path-tracking algorithm for autonomous vehicles [4]. The author proposed a simple, geometrically intuitive approach to path tracking, which has since become the foundation for many subsequent studies. Another noteworthy contribution is by Snider (2009), who extended Coulter's model by incorporating the dynamics and kinematics of different types of vehicles, thus enhancing the adaptability of the pure pursuit controller [8]. In terms of practical applications, pure pursuit controller has been implemented in various settings. For instance, Varundev Sukhil et al. applied the controller to an autonomous vehicle operating in a controlled environment, demonstrating its effectiveness in achieving accurate path tracking [10]. Similarly, Hans Andersen et al. successfully used the pure pursuit controller for an autonomous vehicle navigating in an autonomous golf cart through complex pedestrian environments, underlining its robustness [2]. In the context of Autonomous Driving Trucks (ADTs), studies have begun to explore the application of the pure pursuit controller. For instance, Kresimir Petrinec et al. implemented this controller in an autonomous forklift in a warehouse environment, providing empirical evidence of its viability for ADTs [7].

Organization: The remainder of this paper is structured as follows: Section II discusses the fundamental principles and operational mechanics of the pure pursuit controller. This section further elaborates on implementing the pure pursuit controller, detailing the step-by-step process, including controller design and simulation setup. Section III presents the experimental results, focusing on the performance of the pure pursuit controller in terms of accuracy and adaptability. Conclusions are drawn in Section IV, together with an outlook on future research.

II. Methodology

This section will focus on four crucial aspects: The working principle of the pure pursuit controller, the kinematic vehicle modelling, the geometric interpretation of the pure pursuit control algorithm, followed by the design of the Controller, and the configuration of the simulation setup. These topics will offer a deeper understanding of the controller's operation, underlying principles, and integration with the vehicle's physical attributes.

A. *Pure pursuit controller*

The primary functionality of this controller revolves around its ability to follow a reference path, relying solely on the vehicle's kinematic geometry and the characteristics of the desired path. In doing so, it simplifies the complexity of real-world vehicle dynamics by disregarding dynamic forces exerted on the vehicle. This assumption is primarily based on the premise that wheels maintain a no-slip condition, effectively adhering to the contact surface without skidding. The controller operates by continuously adjusting the vehicle's steering angle to ensure it is always directed towards a "goal point" on the reference path. This point is also known as the "Look Ahead" point. The algorithm continually shifts the look-ahead point along the path based on the vehicle's current position and continues until the vehicle reaches the final point on the path. The positioning of this Look Ahead point is determined by the "Look Ahead Distance" parameter, which dictates the distance between the vehicle and the look-ahead point and this is the main tuning parameter of the controller [11]. As shown in Fig. 2., the path is oscillatory and accurate for a smaller look-ahead distance and, for a larger look-ahead distance, the path is less oscillatory, but the tracking is poor.

B. *Kinematic Bicycle model*

The proposed analysis uses a simplified bicycle model, shown in Fig. 3., to represent the vehicle, where the vehicle's front and rear wheels are each represented by a single wheel. This model provides a useful tool for exploring the vehicle's kinematics. Within this model, we establish a reference point, denoted as X, Y, which can be variably positioned on the vehicle. This point can be positioned at various locations,

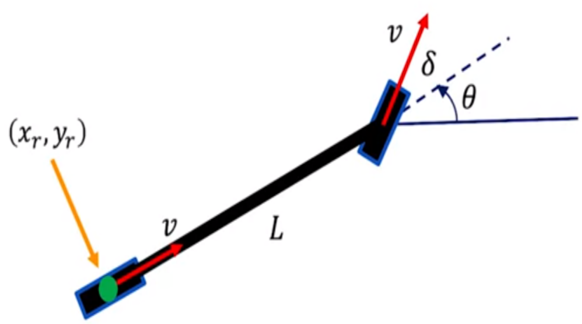


Fig. 3: Rear axle kinematic bicycle model.

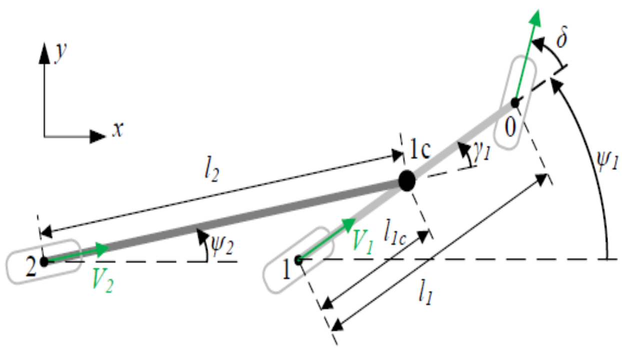


Fig. 4: Tractor semi-trailer kinematic model.

such as at the centre of the rear axle, the centre of the front axle, or the centre of gravity, often referred to as 'cg'. For the pure pursuit controller, the centre of the rear axle is selected as the reference point, denoted by 'x_r' and 'y_r'. 'θ' refers to the heading angle. Another key parameter is the wheelbase, denoted as 'L', which is the distance between the front and rear axles. The steering angle, denoted as 'δ', is measured in relation to the forward direction of the bicycle model. The velocity 'v' corresponds with the direction of each wheel, adhering to the 'no slip' condition. From the no-slip condition,

$$\dot{\theta} = \omega = \frac{v}{R}, \tag{1}$$

where ω is the bicycle's rate of rotation. From the similarity of triangles formed by R and L, and δ and v,

$$tan(\delta) = \frac{L}{R}, \tag{2}$$

From (1) and (2),

$$\dot{\theta} = \omega = \frac{v}{R} = \frac{v \cdot tan(\delta)}{L}, \tag{3}$$

From this configuration, the components of velocity for the reference point in the x and y directions are,

$$\dot{x_r} = v \cdot cos(\theta), \tag{4}$$

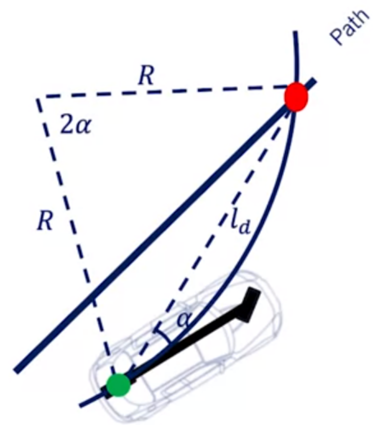


Fig. 5: Pure Pursuit geometry indicating the look ahead point and look ahead distance.

$$\dot{y_r} = v \cdot sin(\theta), \tag{5}$$

These equations, along with the equation for the bicycle's rate of rotation derived earlier, collectively constitute the rear axle bicycle model and is shown below,

$$\dot{\theta} = \frac{v \cdot tan(\delta)}{L}, \tag{6}$$

This model provides a concise mathematical representation of the vehicle's motion, which is crucial for designing the path-following controller.

For implementing the pure pursuit controller for ADTs, this research considers the dynamics of a Tractor Semi-Trailer model shown in Fig. 4. However, the existence of three non-steered trailer axles adds complexity. During a turning manoeuvre, side slip angles inevitably manifest on multiple trailer axles, irrespective of the speed. The consequential forces may induce side slip angles on the tractor as well. In the context of low-speed manoeuvring, it is reasonably assumed that no side slip angle occurs on the tractor and the second axle of the semi-trailer. Under these circumstances, the first and third trailer axles are disregarded for simplicity. One significant challenge with the tractor-semi-trailer configuration is the occurrence of jackknifing. In this situation, the articulation angle between the tractor and the semi-trailer becomes excessively large, potentially leading to a collision between the tractor cabin and the semi-trailer. Constraints are set on the maximum steering angle to avoid this scenario, especially during forward motion.

C. *Geometric interpretation of pure pursuit algorithm*

For the Pure Pursuit control strategy, the key reference point on the vehicle is the centre of the rear axle. The line connecting this reference point to the target point on the planned path is defined as l_d, commonly referred to as the look-ahead distance, represented by the blue dashed line

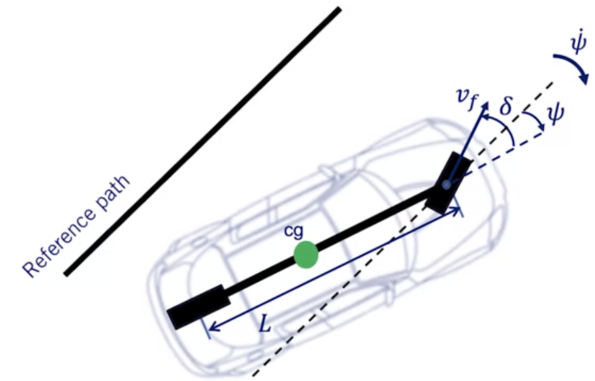


Fig. 6: Bicycle model showing the vehicle heading angle 'ψ'.

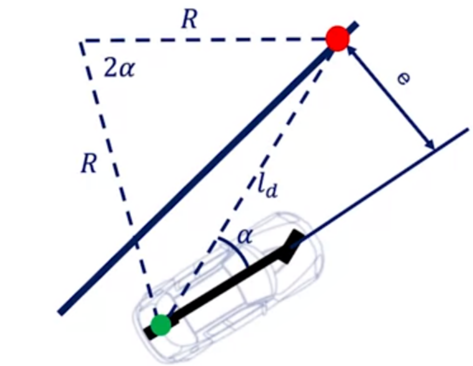


Fig. 7: Bicycle model showing the vehcile cross track error 'e'.

in Fig. 5. The angle 'α' represents the angle between the heading of the vehicle and the look-ahead line. The interpretation of Pure Pursuit control from a geometric perspective relies on applying the instantaneous centre of rotation (ICR) principle. The ICR, the centre of the rear axle, and the target point on the path together form a triangle with two sides of length R (radius of the ICR circle) and one side of length 'l_d' (the look-ahead distance). The arc the vehicle needs to follow to reach the target point from its current reference point can be defined within this triangle. This arc lies on the circumference of the ICR circle and spans an angle of 2α. By utilizing fundamental trigonometric identities and applying the law of sines, the angle 2α can be obtained as shown below,

$$\frac{l_d}{sin(2\alpha)} = \frac{R}{sin(\frac{\pi}{2} - \alpha)}, \tag{7}$$

$$\frac{l_d}{2 \cdot sin(\alpha) \cdot cos(\alpha)} = \frac{R}{cos(\alpha)}, \tag{8}$$

$$\frac{l_d}{sin(\alpha)} = 2 \cdot R, \tag{9}$$

$$\kappa = \frac{1}{R} = \frac{2 \cdot sin(\alpha)}{l_d}, \tag{10}$$

where κ is the curvature of the circular arc, which is the inverse of the radius of the ICR circle (R). Combining (2) and (10), the steering angle needed to track the arc is given by,

$$\delta = \tan^{-1}\left(\frac{2 \cdot L \cdot sin(\alpha)}{l_d}\right). \tag{11}$$

D. *Pure pursuit algorithm- error dynamics*

The controller ensures that the autonomous truck aligns precisely with the intended route, minimizing deviations. The variable 'ψ', shown in Fig. 6. represents the relative heading angle of the truck in relation to the desired path. Two critical types of errors come into play in this scenario: heading and cross-track errors. Heading error, the discrepancy between the intended path direction and the truck's actual heading, is a primary indicator of how well the vehicle aligns and moves in the direction of the desired path. The rate of change of the heading error, denoted as '$\dot{\psi}$', provides insight into the evolution of the heading error over time. The equation is shown below,

$$\text{Rate of heading error} = \dot{\psi}_{des}(t) - \dot{\psi}(t), \tag{12}$$

where $\dot{\psi}_{des}(t)$ is the desired rate of change of heading angle. The second form of error, the cross-track error, represents the displacement between the vehicle's reference point and the nearest point on the planned path. This is shown in Fig. 7. A flawless path-tracking mechanism would require the heading and cross-track errors to be reduced to zero. In the context of the pure pursuit controller for an autonomous truck, the cross-track error is the gap between the truck's directional vector and the target point on its path, represented with the symbol 'e'. The following equation is obtained,

$$sin(\alpha) = \frac{e}{l_d}. \tag{13}$$

Integrating this with the previously derived expression for curvature, from (10) and (13),

$$\kappa = \frac{2}{l_d^2} \cdot e. \tag{14}$$

Equation (14) indicates that the path's curvature dictated by the pure pursuit controller is directly proportional to the cross-track error at the look-ahead point on the path.

E. *Pure pursuit controller design and simulation setup*

A kinematic vehicle model of a tractor-semitrailer truck was provided in MATLAB/Simulink, enabling the simulation of low-speed manoeuvring. The model allows for the prescription of velocity and steering as a function of distance, facilitating the computation of the resulting vehicle motion. In the pure pursuit controller, critical parameters

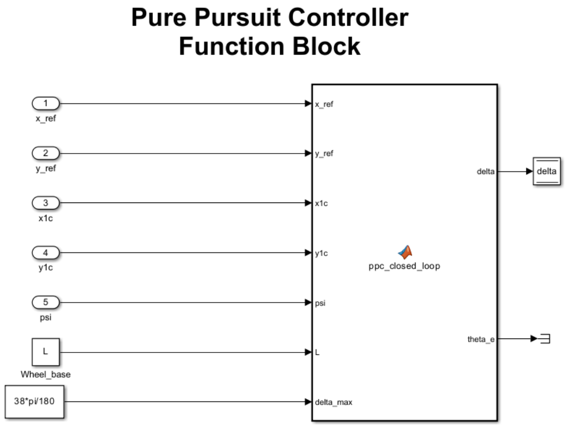


Fig. 8: Pure pursuit controller function block implemented in Simulink.

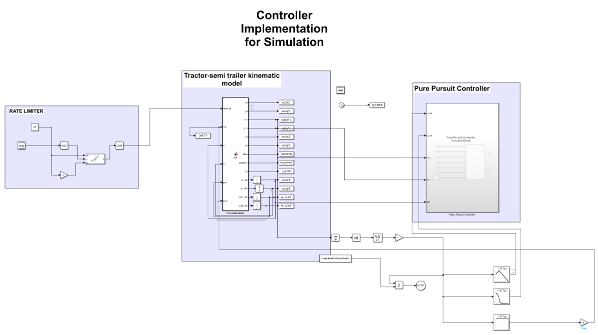


Fig. 9: An overview of the kinematic tractor-semitrailer Simulink block paired with its controlling entity. The system also features the integration of a rate limiter block to prevent jackknifing.

derived from the kinematic model, including the rear axle coordinates (x1, y1), the truck's current position (x1c, y1c), and the heading angle of the truck 'ψ', serve as the input variables. The controller's function block shown in Fig. 8., processes these inputs and outputs the required steering angle accurately guide the truck along the planned path. The system diagram can be visualized in Fig. 9. A series of 1-D lookup tables implemented in Simulink handle the dynamic update of waypoints as a function of the distance travelled. The simulation is designed to halt once the vehicle reaches or is within a set proximity threshold of the last point in the waypoint.

Upon successfully validating the pure pursuit controller using the provided kinematic model, the next stage involves its integration and validation within a 3D virtual environment called the Virtual Entity. This environment has been designed utilizing the Unity Game Engine. The practical application of the controller, translated into MATLAB/Simulink blocks for driving the tractor within the Virtual Entity, is detailed in Fig. 10. For a more comprehensive understanding of the controller's operational code used in the virtual simulation,

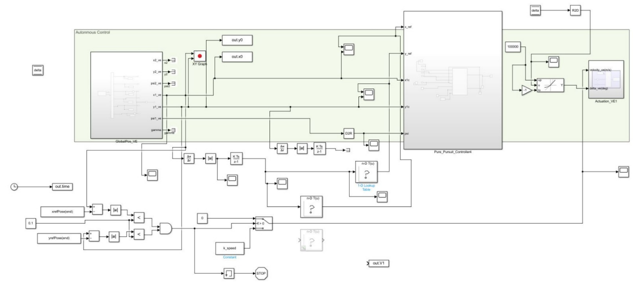


Fig. 10: Depicts the integration of the pure pursuit controller with the Virtual Entity blocks.

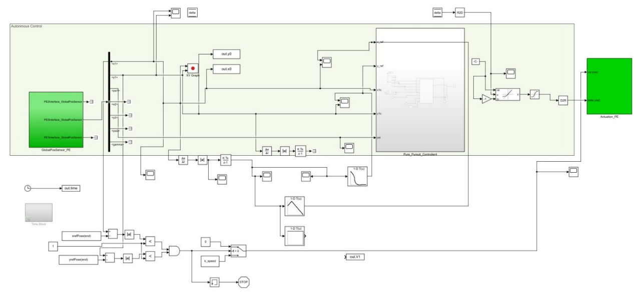


Fig. 11: Depicts the integration of the pure pursuit controller with the Physical Entity blocks.

readers are referred to Appendix B. This approach ensures a thorough validation process, extending from abstract model simulations to an immersive 3D environment, thus underlining the potential for practical deployment of the pure pursuit controller in real-world applications.

In the final stage of methodology, the pure pursuit controller is subjected to testing and validation using the AES Lab's physical truck. To facilitate this physical testing, the Virtual Entity blocks within the simulation setup are replaced with Physical Entity blocks. These new blocks interface with the global position sensor and actuator of the physical truck, ensuring a direct link between the controller and the hardware of the truck. This configuration is outlined in Fig. 11.

Through this multi-tiered approach, progressing from kinematic modelling, through a virtual environment, to actual physical testing, our methodology ensures a comprehensive validation of the pure pursuit controller, reinforcing its suitability for real-world ADT applications.

III. Experimental results

In this section, the key findings from the three-tiered simulation process are showcased, highlighting the performance of the pure pursuit controller under various simulation conditions. In the first stage, the kinematic model simulation, the plot shown in Fig. 12. demonstrates the truck's ability to almost precisely track the given path, which highlights the accuracy and reliability of the pure pursuit controller. An interesting observation from the plot pertains to a series of oscillations in the path. These oscillations can be attributed to

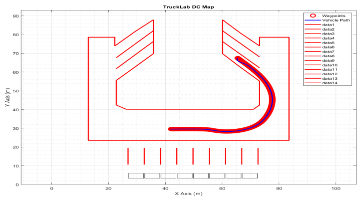


Fig. 12: Illustration of the TruckLab map.

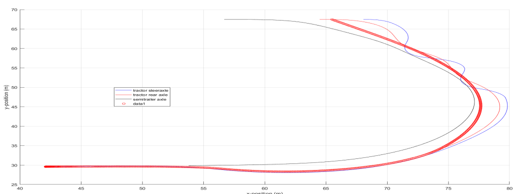


Fig. 13: Truck navigating through a set of waypoints.

the truck's starting position being proximate to the beginning of the waypoint. This proximity leads to the computation of a relatively small look-ahead distance, which subsequently introduces oscillations into the truck's trajectory. The TruckLab map is also provided in Fig. 13, illustrating the truck's path along the designated waypoints. This graphic representation affirms the practicality and feasibility of our controller when deployed in real-world distribution centre scenarios.

In the second stage, the findings from the virtual simulations are shown in Fig. 15 and Fig. 17. For these simulations, two calibrated path-following tasks were conducted involving the truck operating within a virtual representation of the distribution center. The first task involved navigating from the entrance of the distribution center to the docking area. The second task, involved navigating the truck as it made its way from the docking area back towards the exit of the distribution center. In both scenarios, shown in Fig. 14 and Fig.16, the simulation plots reveal that the truck maintained a high degree of accuracy in following the pre-defined path.

IV. Conclusion

This study has significantly advanced our understanding of the potential and limitations of the pure pursuit controller in the context of Autonomous Driving Trucks (ADTs). By operating at a constant velocity, the controller can accurately follow a predetermined course, executing a planned movement strategy precisely. The controller's performance was evaluated across three scenarios: tractor semi-kinematic model simulation, virtual environment realized through Unity Game Engine, and, a real-world trial using AES Lab's physical truck. These tests underscore the controller's adaptability across multiple settings, highlighting its robustness and reliability. The pure pursuit controller has shown to be particularly effective in controlled environments like distribution centres, where its advantages can be fully realized.

This work calls for the following possible extensions: First, the static velocity model could limit the controller's

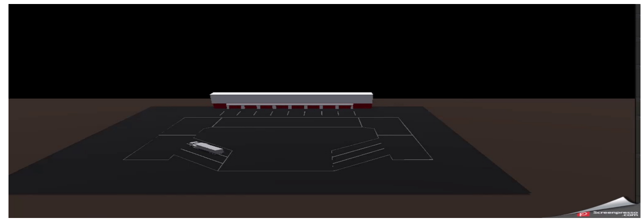


Fig. 14: Scenario 1: Truck located at the entrance.

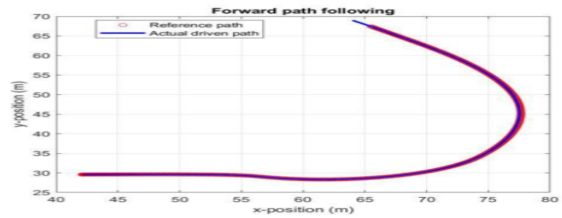


Fig. 15: Truck precisely following calibrated path 1.

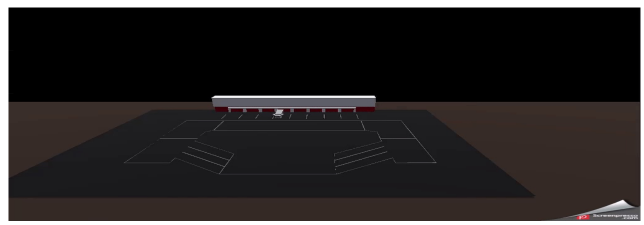


Fig. 16: Scenario 2: Truck located at the docking area.

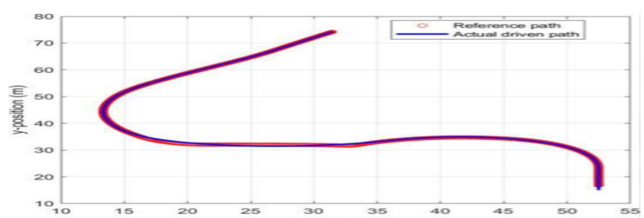


Fig. 17: Truck precisely following calibrated path 2.

adaptability to dynamic environments. Thus, a significant implication of our findings for practitioners and researchers alike is the potential performance enhancement that might be achievable through a variable velocity model. Second, an intriguing prospect would be to expand the controller's capability to include reverse motion. This development could significantly increase the versatility of ADTs, especially in scenarios where reverse movements are essential, such as rectifying path errors, and loading and unloading goods. However, implementing a reverse controller brings challenges, such as the risk of jackknifing. Therefore, a deeper investigation into the dynamics of the trailer would provide a comprehensive understanding of the factors influencing jackknifing, thereby informing the development of more sophisticated control strategies.

Acknowledgment

I am very grateful to Dr. Ion Barosan for proofreading this paper and countless papers and documents before this, teaching the authors something new every time. Moreover, this paper draws inspiration from ”Guidelines for Writing Papers and Reports” by Dr. Mauro Salazar and Olaf Borsboom, which has taught me much about writing scientific papers.

Appendix A
Matlab code for the controller function

```
function [delta, theta_e]  =
   ppc_closed_loop(x_ref, y_ref, x1c,
   y1c, psi, L, delta_max)

%Obtaining the waypoint co-ordinates and
     the current position of the truck
waypoint = [x_ref, y_ref];
current_position = [x1c, y1c];

% Compute the desired heading angle
theta_d = atan2(waypoint(2) -
   current_position(2), waypoint(1) -
   current_position(1));

% Compute the heading error
theta_e = (theta_d - psi);

% Calculate the look ahead distance
ld = vecnorm(waypoint - repmat(
   current_position, size(waypoint, 1),
   1), 2, 2);

% Compute the steering angle based on
   the heading error and lookahead
   distance
delta = atan((2*L*sin(theta_e))/ld);
delta = sign(delta) * min(abs(delta),
   delta_max);

end
```

Appendix B
Matlab code for running the simulation

```
clear all; close all;

TractorSemitrailer_parameters;   % load
                     % vehicle parameters

tmax= 100;      % simulation time [s]
fs  = 10;       % sample frequency for
                % data storage [Hz]
freq = fs;

%% Load Trajectory
path = load('SampleTrajTemp.mat') ;

%%
L = 3.6;        %Wheel base

%% Determine speed profile
xrefPose = path.m.fwd_x1c;
yrefPose = path.m.fwd_y1c;

dx = diff(xrefPose);
dy = diff(yrefPose);

cumulativeDistanceX = [0; cumsum(abs(dx)
   )];
cumulativeDistanceY = [0; cumsum(abs(dy)
   )];

k_speed = 0.1; % Gain factor for speed
v = k_speed*ones(size(
   cumulativeDistanceX,1),1);
v = [v(1:end-1); 0];

figure(99)
plot(cumulativeDistanceX,v,'black','
   LineWidth',1.5)
grid on
xlabel('Distance')
ylabel('Velocity')

%% For running the Simulink Simulation
tic
s=sim('control_temp_kin.slx');
toc

%% Plot
waypoints = [xrefPose, yrefPose];
relative_waypoints = waypoints;
figure
hold on
plot(s.x0,s.y0,'b') % Steer axle
   position
plot(s.x1,s.y1,'r') % Drive axle
   position
plot(s.x2,s.y2,'k') % Trailer axle
   position
legend('tractor steeraxle','tractor rear
    axle','semitrailer axle','Location',
   'best')
xlabel('x-position (m)')
ylabel('y-position (m)')
plot(relative_waypoints(:, 1),
   relative_waypoints(:, 2), 'ro', '
   MarkerSize', 5, 'LineWidth', 0.1);
grid on;
```

References

[1] Omead Amidi and Chuck E Thorpe. Integrated mobile robot control. In *Mobile Robots V*, volume 1388, pages 504–523. SPIE, 1991.

[2] Hans Andersen, Zhuang Jie Chong, You Hong Eng, Scott Pendleton, and Marcelo H Ang. Geometric path tracking algorithm for autonomous driving in pedestrian environment. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1669–1674. IEEE, 2016.

[3] Ion Barosan, Arash Arjmandi Basmenj, Sudhanshu GR Chouhan, and David Manrique. Development of a virtual simulation environment and a digital twin of an autonomous driving truck for a distribution center. In *Software Architecture: 14th European Conference, ECSA 2020 Tracks and Workshops, L'Aquila, Italy, September 14–18, 2020, Proceedings 14*, pages 542–557. Springer, 2020.

[4] R Craig Conlter. Implementation of the pure pursuit path'hcking algorithm. *Camegie Mellon University*, 1992.

[5] Jingjing Jiang and Alessandro Astolfi. Lateral control of an autonomous vehicle. *IEEE Transactions on Intelligent Vehicles*, 3(2):228–237, 2018.

[6] A Ollero and O Amidi. Predictive path tracking of mobile robots. application to the cmu navlab. In *Proceedings of 5th International Conference on Advanced Robotics, Robots in Unstructured Environments, ICAR*, volume 91, pages 1081–1086, 1991.

[7] Krešimir Petrinec, Zdenko Kovacic, and Alessandro Marozin. Simulator of multi-agv robotic industrial environments. In *IEEE International Conference on Industrial Technology, 2003*, volume 2, pages 979–983. IEEE, 2003.

[8] Jarrod M Snider et al. Automatic steering methods for autonomous automobile path tracking. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.

[9] Qunying Song, Emelie Engström, and Per Runeson. Concepts in testing of autonomous systems: Academic literature and industry practice. In *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, pages 74–81. IEEE, 2021.

[10] Varundev Sukhil and Madhur Behl. Adaptive lookahead pure-pursuit for autonomous racing. *arXiv preprint arXiv:2111.08873*, 2021.

[11] SHIFA SULAIMAN and AP SUDHEER. Implementation of pure pursuit algorithm based controller for a mobile platform.

Individual Paper - Integration of SysML Model with Simulation and Visualisation Tools

4AT100 Course

Team 2

Full Name	Student ID	Study
A. Gali	1660004	Automotive Technology

Eindhoven, July 13, 2023

Contents

Abstract

System Architecture Design is important for digital transformation and verification. The problems pertaining to Industry 4.0 are becoming increasingly complex. On the other hand, technology is also being developed at a faster rate to deal with such problems. The system integration consisting of different technologies is ever more important. A system architecture design consisting of this integration can provide a common Business Process and Visibility into whats going on within the project of the organization.The primary step would be to recognise the suitable technologies that are applicable to the ongoing project. Followed by the data and control flow diagram which will be the subset of the system architecture. This can give an overview and complexity involved in this project. Model Based Systems Engineering can provide a platform to build the System Architecture. SysML (System Modelling Language) diagrams help the organization to understand the processes involved and with the capability to convert these SysML models into codes through the softwares will facilitate in the data and control flow to the different technologies.This research paper aims to provide an architecture for the system and software integration focusing on the Automotive Field. The objectives of this research paper are to find an appropriate Architecture Style that can be utilised for automotive applications, to verify the reliability of tools communicating with each other to provide desired outputs and the opportunity to verify the requirements laid out by the stakeholders through visual representation.

1 Motivation for the Individual Paper

The 4AT100 Project is a large-scale undertaking that spans several areas. In order for the team to complete the project effectively, it will require substantial research and teamwork. The project's domains cover a broad range of subjects covered in the Automotive Technology Master's Degree program. Each domain denotes a particular area of experience and knowledge, and it is essential that team members thoroughly research the domains they have been given.

The project gains from a wide pool of experts who each have in-depth knowledge of their respective disciplines by dividing the domains across team members. This method makes sure that every important focal area is covered and that every potential angle is addressed. The team members can bring important insights, ideas, and conclusions to the project thanks to their specific knowledge. The collaborative aspect of the project encourages a setting where team members can benefit from one another's knowledge and experience while also working together to accomplish the project's goals.

The 4AT100 Project's research and collaboration not only help to build interdisciplinary connections but also a thorough understanding of each topic. Team members can discover connections, overlaps, and potential synergies between various project components as they delve deeper into their own expertise. Through the integration of many viewpoints and knowledge, a coherent final product is produced, which improves the project's overall coherence. The team's collaborative efforts and comprehensive understanding of each domain help to ensure extensive study, analysis, and coverage of the numerous focal areas within the Automotive Technology Master's Degree course, which contributes to the project's success.

2 Introduction

Prototyping of the products has been implemented from many years. Prototypes help in understanding the product and its limitations. Rectifying these limitations is the next task before releasing the product to the market. During the early days of prototyping, mainly consisted of physical models which were either made from sheet metal or clay in case of automobiles. These require significant amount of time and labour and the product can be visualised once the prototypes were fully built. With the introduction of CAD Modelling and Programming, there has been a shift from Physical Modelling to Virtual Modelling. The advantages of Virtual Modeling range from decreased cost of prototyping, speeding the process of concept and prototype stages, real time customisation to be able to test a product or system works before building it.With the introduction of Systems Modelling Language (SysML) which supports specification, design, analysis and verification, enhanced the effect of virtual prototyping. SysML has now been emerged as an important tool for model based system engineering applications.

A strong method for designing and analyzing complex systems can be made possible by the integration of SysML modeling, simulation, virtual visualisation and verification. While simulation modeling provides the investigation of system performance under various scenarios, SysML modelling enables the building of system-level models that capture the structure, behavior, and needs of a system. Before the system is physically implemented, the system design can be tested and validated via digital visualization. Engineers may more effectively design and test these complex systems, which help in decreasing the development time and cost. This integrated approach also makes it possible to analyze the system design in more detail. Simulations help in finding potential issues or constraints in the system design. Engineers can evaluate system performance by visualising the product or system, which enables them to spot possible problems that would not be obvious in a static model. Engineers can work together more efficiently thanks to the integration of these three technologies since they can share models, simulations, and visualisations, which enhances communication and lowers the possibility of mistakes. Overall, the combination of simulation modeling, SysML modeling, virtual visualisation and verification, is a potent method that may greatly enhance the efficacy and efficiency of complex system design and analysis.

3 SysML modelling

The Object Management Group (OMG) has developed the Systems Modeling Language (SysML) as a standard to assist in the design, analysis, and verification of systems that may contain both software and hardware components. SysML incorporates features of UML while also adding new ones, such as value types and quantity types, as well as the ability to describe the behavior of continuous systems. SysML provides engineers the tool to build the system without intensive programming. Due to this simplicity, SysML has been used in many applications. Systems are using more software than ever before along with equal focus on hardware, such as Internet of Things (IoT), Cyber Physical Systems (CPS), and Cyber Physical Production Systems (CPPS), have increased their adoption of Industry 4.0 principles in the industrial sector. Due to these requirements, updates to SysML have been made several times regarding the redefinition of physical flows and architectural alignments with UML. These changes may have had the additional effect of making SysML more appealing to software engineering and systems engineering. Increasingly, SysML is required for concurrent design processes and for mechanical concept designs.

SysML applies the semantic of UML 2.0 diagrams as class or object diagrams to eliminate software vocabularies (e.g. class and object are replaced by blocks). In addition, new diagrams are introduced to prepare simulation-based design and to streamline requirement declarations. Both languages can be used for system engineering, but SysML provides unique properties for studying systems reliability. Multi-technological system modeling is made possible by both languages' object-oriented approach, hierarchy, and composition options.Also, it is well known that their graphical explanation is simple to understand and minimizes misunderstandings. A significant benefit of these languages is that numerous software tools support them. SysML makes use of diagrams and visual elements such as blocks, classes, flow connections etc., to describe the model of the system. SysML modelling is possible with different tools which provide easy application of this Modelling Language while maintaining the language specific syntax. IBM Rhapsody is one such tool which is being used extensively in the industries and academics.

SysML facilitates requirement modelling. The Business, Functional and Non-Functional Requirements can be documented in the tool and referred whenever necessary during the phase of the project. These requirements help to verify the end product for successful implementation of all the necessary features. Major features of the SysML are the Block Definition Diagrams (BDD) and behaviour diagrams. The former helps in laying out the context and overview of the system, with the interaction of the system with other components and systems. Figure 1 shows an example of BDD of Distribution Center Control System with the interactions.

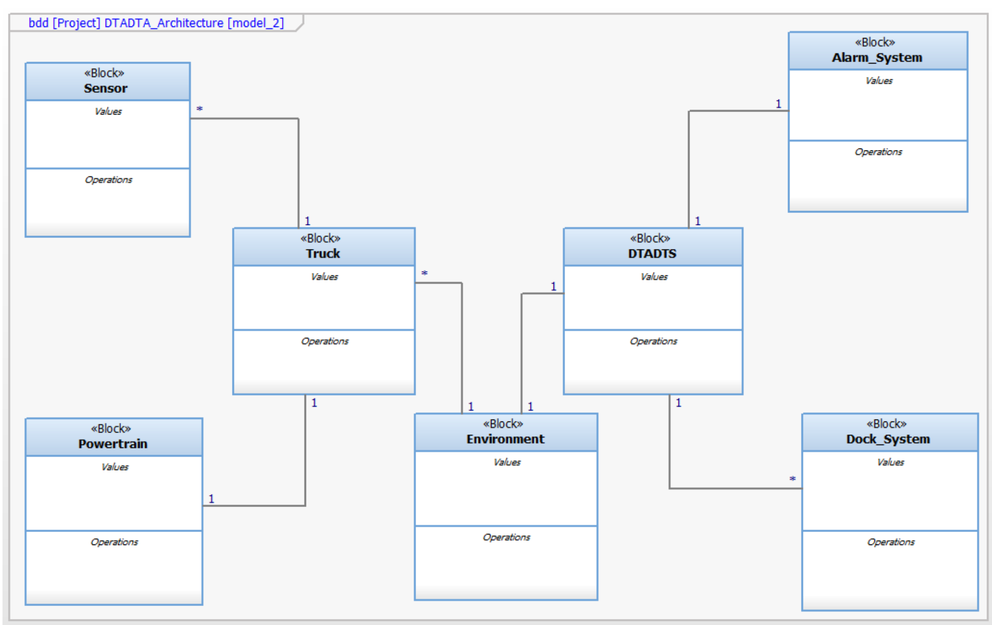


Figure 1: Block Definition Diagram of Distribution System in IBM Rhapsody Tool (System Context)

Behavior Diagram as the name suggests, facilitate in designing the behavior i.e simulation of the system. There are different type of Behavior Diagrams which are suitable for different applications and understanding. While Sequence Diagram provides line by line working of the system, State Machine Diagram (SMD showcases the change of States between the system according to the inputs and conditions. Figure 2 shows an example of SMD of the distribution center where depending on the X and Y coordinates of the truck, the status of the Truck is changed inside the Distribution Center Control System.

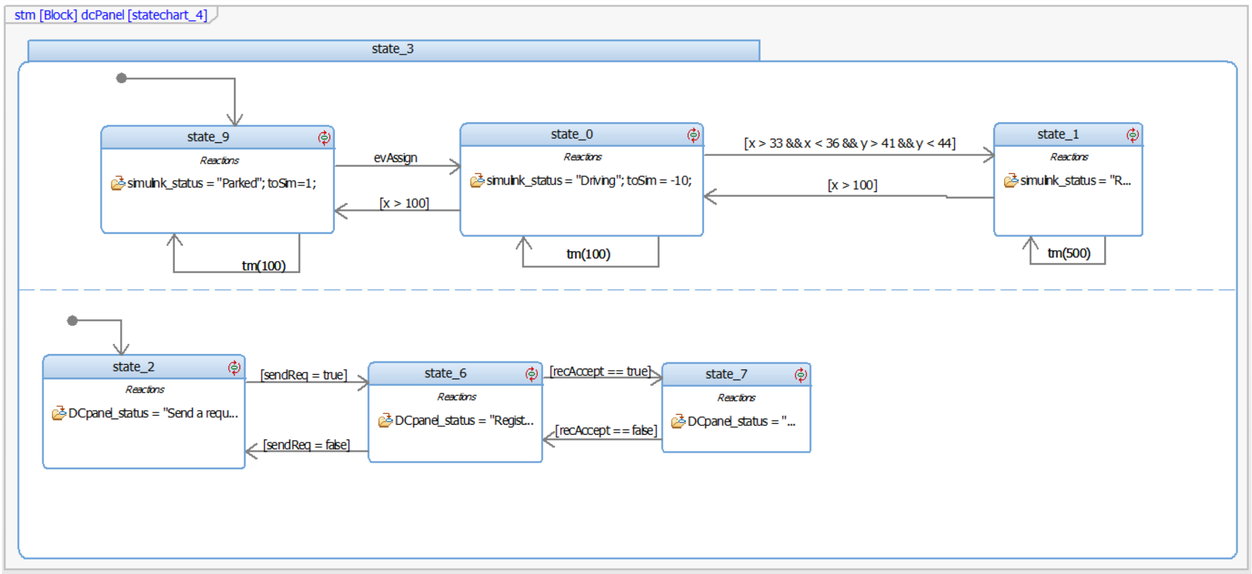


Figure 2: State Machine Diagram of Distribution Center in IBM Rhapsody Tool

4 Mathematical Simulation

Matlab and Simulink are tools that are widely used in a variety of industries and academic subjects due to their capabilities for complex problem solving, data analysis, modeling, and simulation. Because of their user-friendly interfaces, and large libraries. These softwares are critical tools for engineers, scientists and researchers.
Simulink is best known for modeling, simulating, and analyzing dynamic and non-linear systems. A block diagram technique is used to portray system components as blocks connected by signal lines. Simulink allows users to create and simulate a wide range of phenomena, including electrical circuits, control systems, robotics, and communication systems. Simulink provides an intuitive visual depiction of complex systems. Simulink's extensive collection of pre-built blocks and customizable components simplifies the modeling process and allows users to easily translate practical concerns into mathematical models. Engineers and researchers can gain insights, validate designs, improve performance, and make well-informed decisions by simulating the behavior of these models before putting them into effect in the actual world.

5 3-Dimensional Visualization

Engineers use 3D visualization software, such as Unity and Unreal Engine, to precisely construct, prototype, and visualize designs. Unreal Engine delivers cutting-edge rendering technology for incredibly realistic designs, while Unity gives a user-friendly platform for creating dynamic 3D worlds. Both engines enable real-time team collaboration and seamlessly connect with engineering applications, revolutionizing engineering techniques and producing ground-breaking results.
Significant improvements and savings have been made in engineering thanks to the integration of game engines. Engineers may produce realistic 3D simulations using Unity and Unreal Engine, which improves design comprehension and evaluation. Engineers build virtual prototypes and run extensive tests using potent rendering and physics engines, cutting costs and spotting flaws early. Teamwork and continuous design improvement are fostered via collaborative features. Effective engineering concept communication is aided by immersive experiences. Traditional methods have been altered by game engines, which have improved workflows, visualization, and delivered better results.

6 Integration of SysML, Simulation and Visualisation

6.1 SysML and Simulation tool

In general, a system can be considered as a product comprising of various elements or components to achieve a common goal which cannot be achieved by a single component. Current systems are a combination of mechanical and electronic components making them the mechatronic systems, also consisting of intelligent control. Model-based software engineering (MBSE), which depicts system engineering problems from several perspectives and turns stakeholder requests into particular design specifications through model evolution or transformation, has increasingly gained recognition as an approach for building complex systems.SysML is frequently used in mechatronic system design to provide a straightforward, transferable model for these mechatronic systems by modeling system requirements, structures, and behaviors using its rich semantics. Where SysML lags behind is the fact that it is not capable of describing continuous dynamic behavior of the system. That is why simulation tools are necessary for the system's behavior to be verified.
By establishing a mapping relationship between the SysML model produced by the system modeling tool into the system architecture model in the simulation tool based on the architecture modeling capability of the simulation tool. The transformation to the system architecture model can represent the content in the early design phase of the system model more thoroughly than the conventional simulation model transformation based on the extraction of the system model skeleton. The system architecture model can be developed into the executable simulation model that corresponds to the physical architecture in the process of synchronization for further verification because of the natural integration between the transformed system model elements and the simulation model elements in the simulation tool. The communication between the design groups is further strengthened, the number of design iterations is decreased, and the cost of learning is decreased through this approach, which is more natural and practical than other model transformation processes. Figure 3 provides an overview of different tools that are used for the integration between the SysML modelling and Simulation.

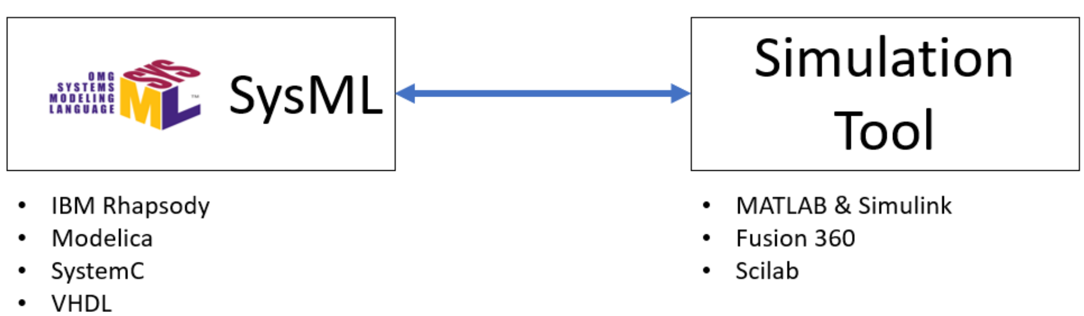


Figure 3: Overview of tools for the integration

The integration of the two fields starts with the fact that SysML will be the Modelling language for behaviour of the system which consists different states of activities involved in the design, where as, the simulation tool helps in representing the system in terms of mathematical expressions. The SysML field consists of meta-model, requirements, Use Cases, System Architecture, Simulation Execution Profiles and Model Libraries. The function of executing the simulation on the established model and producing the result set in accordance with the supplied input parameters is provided by the simulation execution solver. Figure 4 shows an overview of the elements of integration.

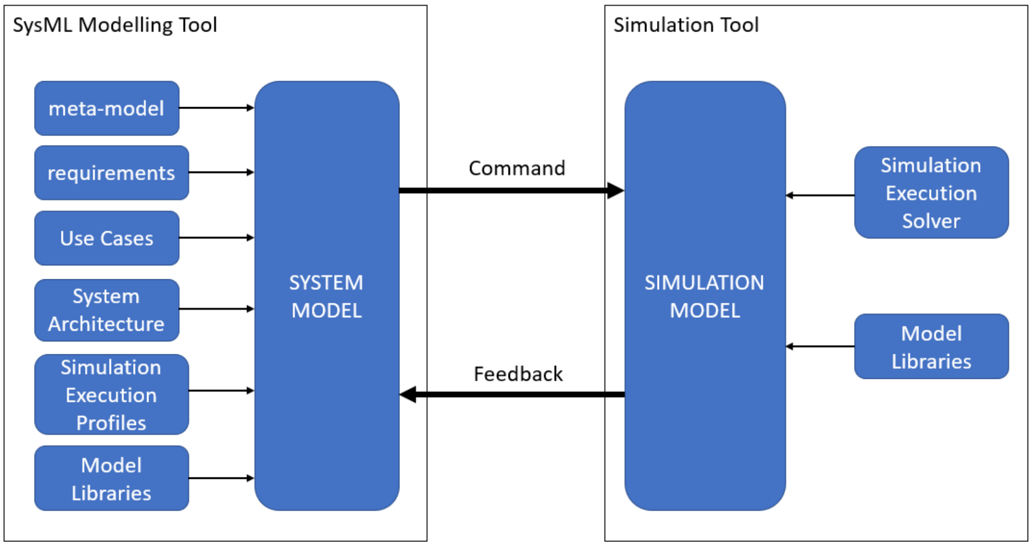


Figure 4: Integration overview between SysML Modelling Tool and Simulation Tool

Information about simulation setting settings, the initial values of system design parameters, and whether parameters are limited are all provided by the SysML model. The simulation tool will receive the information to process. The transformation is generally implemented in two ways namely, Model-To-Text (M2T) and Model-To-Model (M2M). For any simulation environment, the expanded SysML model can create the executable code directly from the code template that derives from the source meta-model, such as MATLAB code. In order to guarantee the consistency of the system model in a more uniform manner, the M2M technique sets the mapping and transformation rules based on the meta-model of the source model and the target model. Yet this approach's underlying assumption is that both the source model's and destination model's meta-models are accessible.

The SysML model consists of blocks, Input and Output Ports and Links between the blocks. A block is regarded as the modular unit of structure. IBD is one of the SysML diagrams used to describe a block's internal structure by showing how its components are related. Components can interact through Input and Output Ports, which have variables indicating various aspects of the energy or information transferred with other components. Links stand for the connections between parts, or ports, via which energy or data is transferred. They resemble the notion of SysML connectors. They enable describing the connections between the components of a block. The simulation model consists of mathematical expressions or equations, functions, blocks, connections etc. After the mapping is done from the SysML model onto the simulation tool, the tool will display similar structure of the system as proposed in SysML model.

6.2 Simulation Tool and Visualisation tool

Simulink's integration with Unity Game Engine, a potent simulation and modeling tool, opens up new avenues for engineering applications. Engineers may build highly realistic and accurate simulations by integrating Simulink's computational capabilities with Unity's real-time rendering and interactive features. Engineering professionals can now see and evaluate complicated systems in a virtual environment thanks to this connection, which enables the easy transmission of data and settings between Simulink and Unity.

Engineers can, for instance, model and simulate a mechanical system's dynamics using Simulink, and then import the simulation results into Unity to produce a lifelike 3D representation of the system. By bridging the gap between simulation and visualization, this integrated technique offers engineers a singular opportunity to learn important lessons about system behavior, carry out virtual testing, and improve designs before physical implementation.
Interactive learning opportunities and interdisciplinary cooperation are also made possible by the integration of Simulink and Unity Game Engine. To produce interesting and interactive simulations, engineers can collaborate with other team members like graphic designers or game developers. In areas like robotics, autonomous driving, and industrial automation, where precise and realistic simulations are essential for design validation and performance assessment, this integration is especially advantageous. Additionally, Unity's interactive features enable user involvement with the simulated system, creating a realistic setting for training, instruction, and demonstrations. The combination of Simulink and Unity Game Engine improves engineering workflows, fosters innovation, and creates new opportunities for simulation-driven engineering applications, whether it be for displaying intricate control algorithms or developing virtual environments for training operators.
The Robot Operating System (ROS) network and Simulink's integration improve robotics development. Engineers can create intricate control algorithms and behavior models in Simulink thanks to this combination, which also makes use of ROS for smooth communication and coordination between different systems. This connectivity simplifies testing and prototyping, enabling the development of sophisticated robotic systems in fields like industrial automation and driverless vehicles. Engineers may create and visualize robotics simulations utilizing a strong framework by integrating Simulink with Unity Game Engine over the ROS network. Engineers may create intricate control algorithms and behavior models in Simulink while observing and engaging with the simulation in a lifelike 3D environment in Unity by utilizing the ROS network and Simulink models. By combining the computational power of Simulink, the connectivity of ROS, and the immersive visualization of Unity, engineers can now build sophisticated robotics simulations and assess system performance in a way that is both highly interactive and aesthetically pleasing. Figure 5 shows an overview of the flow of data and control between the Simulation tool and the 3D Visualization Tool.

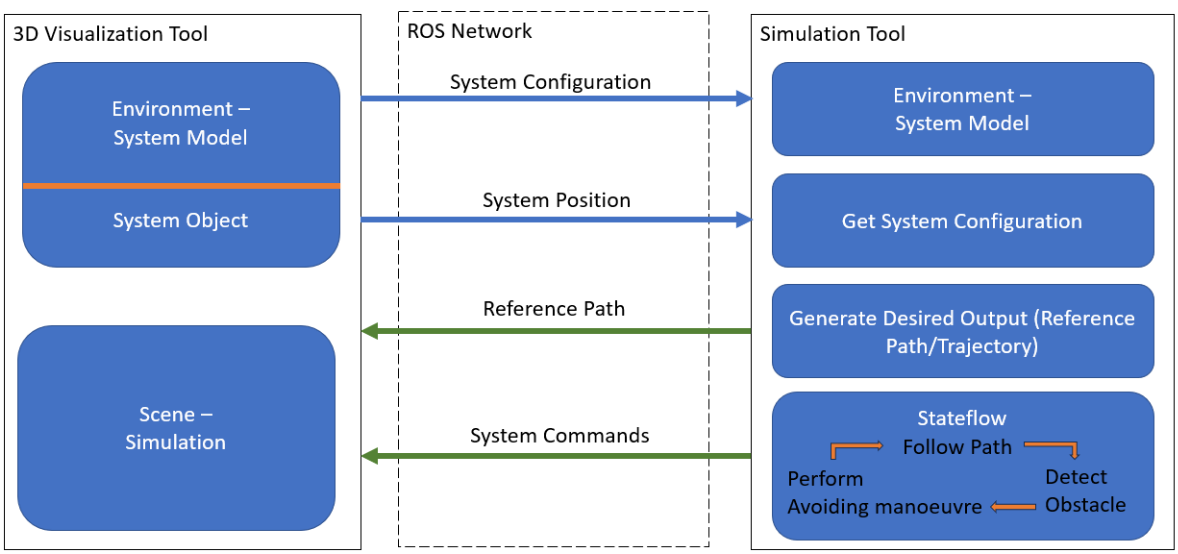


Figure 5: Integration overview between 3D Visualization Tool and Simulation Tool

7 Conclusion and Implementation in the 4AT100 Project

The integration of different software fields is performed to showcase the usability and flexibility of the these softwares. It facilitated in designing an interactive system which helped in realising the true potential of these softwares and the technology.

For the 4AT100 project, IBM Rhapsody is used as Control Panel of the Distribution Center which consisted the functionality of Assigning docks to the trucks, keeping the number of trucks inside the Distribution Center in check and also to shutdown the system if any error is about to or has occurred. The counter part is Simulink which consisted the Control algorithm for the trucks to move autonomously. The control panel is designed in IBM Rhapsody using the Block Definition Diagram (BDD), Internal Block Diagram (IBD) and State Machine Diagram (SMD). "Structured Simulink Block" is chosen inside the BDD which acts like a platform on which the system can be built. "Simulink Block" is chosen as one of the block which constitutes for the Simulink file. Figure 6 shows the BDD designed for the project. Once the BDD is complete and the ports are defined, the connections are done in IBD and the stateflow is designed in SMD. Figure 7 shows the complete connection along with the Simulink file.

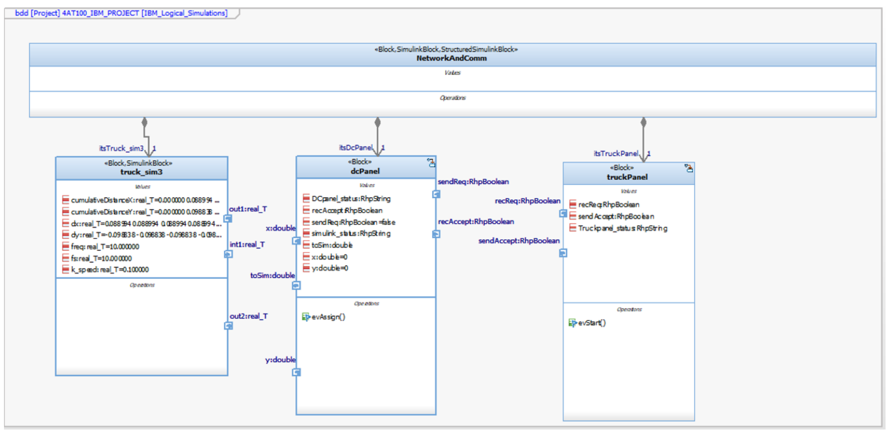


Figure 6: BDD for 4At100 Project

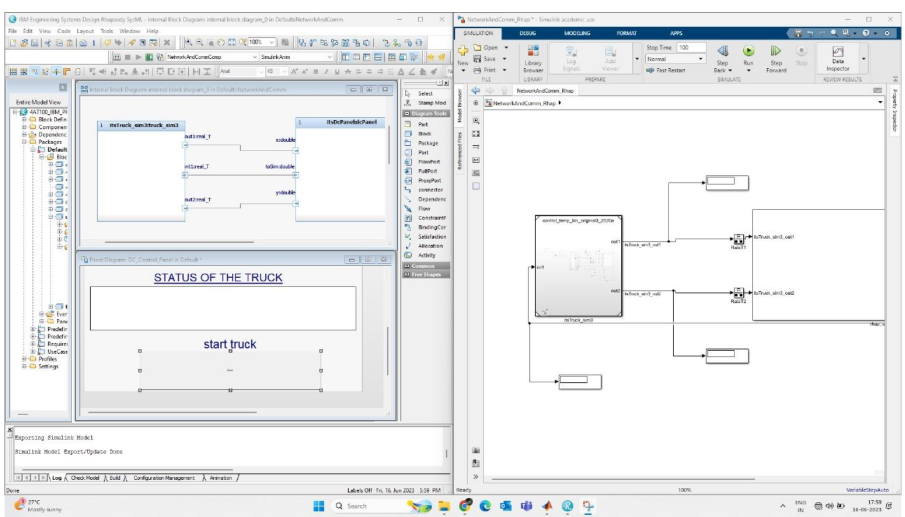


Figure 7: Integration of IBM Rhapsody (SysML Tool) and Simulink (Simulation Tool) for 4AT100 Project

References

1. David, Pierre, Idasiak, Vincent, Kratz, Frédéric. (2009). Improving reliability studies with SysML. Proceedings - Annual Reliability and Maintainability Symposium. 527 - 532. 10.1109/RAMS.2009.4914731.

2. Hause, Matthew. "The SysML modelling language." Fifteenth European Systems Engineering Conference. Vol. 9. 2006.

3. Chu, C.; Yin, C.; Su, S.;Chen, C. Synchronous Integration Method of System and Simulation Models for Mechatronic Systems Based on SysML. Machines 2022, 10, 864. https://doi.org/10.3390/machines10100864.

4. Chabibi, Bassim, Anwar, Adil, Nassar, Mahmoud. (2015). Towards an alignment of SysML and simulation tools. 1-6. 10.1109/AICCSA.2015.7507216.

Controller design for forward manuevering of an Autonomous truck in a Distribution Centre

Aryan Kumar, Automotive Technology, Eindhoven University of Technology, Eindhoven, THE Netherlands
e-mail: (a.kumar@student.tue.nl).

***Abstract*—This paper presents a controller design framework for the forward dynamics of an autonomous truck. The forward dynamics of a vehicle refer to its motion and behavior in response to external forces and inputs. In the context of an autonomous truck, the controller design plays a critical role in ensuring safe and efficient operation.**
The proposed controller design framework leverages advanced control techniques, such as Pure pursuit control to achieve accurate and robust control of the autonomous truck's forward dynamics.
Simulations and experiments are conducted to validate the effectiveness of the proposed controller design framework. Performance metrics such as trajectory tracking accuracy, stability, and response time are evaluated to assess the controller's performance in various scenarios.

I. Introduction

A Model is constructed to simulate the motions of the scaled truck in multiple scenarios. The model is restricted to kinematic model beacuse there are too many unknown variables to make a detailed model. The kinematic model will be compared, verified and adjusted to the TruckLab vehicles to give the best possible representation.

A. Kinematic Vehicle Model

Normally the side slip angles of the tires have to be considered in vehicle dynamics, as they have a major impact on the handling characteristics (e.g. understeer/oversteer). However when the lateral accelerations are low and the accompanying lateral tire forces are small, nearly zero, we may assume that the side slip angles remain zero. This allows to create a kinematic vehicle model, which can be used to describe vehicle motions for low lateral accelerations.

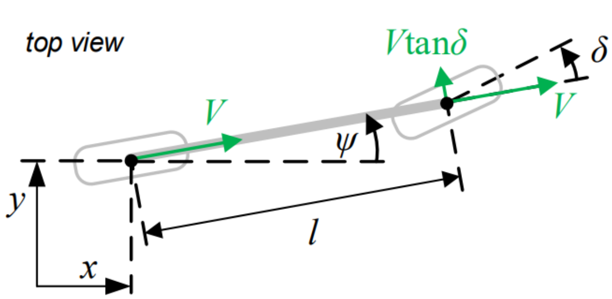


Fig. 1. Single track kinematic vehicle model

V : vehicle forward velocity at the rear axle
δ : front wheel steering angle Both V and δ can be prescribed as a function of time The lateral velocity of the front wheel, expressed in the chassis frame, equals $V\tan\delta$, ensuring that the front tire has zero side slip angle.
Governing equations:

$$v_x = V\cos\psi$$
$$v_y = V\sin\psi$$
$$\dot{\psi} = \frac{V\tan\delta}{l}$$

Where v_x and v_y are the linear velocities of the rear axle in the global frame, ψ is the yaw velocity. The velocities have to be integrated to obtain the global x and y coordinates of the rear axle and yaw angle ψ.

$$x = \int v_x dt, y = \int v_y dt, \psi = \int \psi dt$$

B. Tractor semi-trailer kinematic model

A tractor semi-trailer can also be modelled with a kinematic model, however the three non-steered trailer axles pose a problem. When making a turn (at any speed!)side slip angles will occur on multiple trailer axles. The accompanying forces will also result in side slip angles on the tractor.
The simplifying assumption is that during low speed maneuvering no side slip angle occurs on the tractor and 2nd axle of the semitrailer. The 1st and 3rd trailer axles are simply neglected.

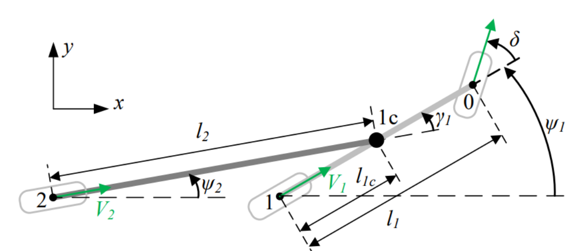


Fig. 2. Tractor semi-trailer kinematic model

Governing equations:

$$v_{x1} = V_1\cos\psi_1$$
$$v_{y1} = V_1\sin\psi_1$$
$$\psi_1 = \frac{V_1\tan\delta}{l_1}$$
$$V_2 = V_1\cos\gamma_1 - \psi_1 l_{1c}\sin\gamma_1$$
$$\psi_2 = \frac{V_1\sin\gamma_1 + \psi_1 l_{1c}\cos\gamma_1}{l_2}$$

The velocities have to be integrated to obtain the global x_1 and y_1-coordinates of the rear axle and yaw angles ψ_1 and ψ_2.

$$x_1 = \int v_{x\leq} dt, y_1 = \int v_{y1} dt, \bar{\psi}_1 = \int \psi_1 dt, \psi_2 = \int \psi_2 dt$$

The articulation ange γ_1 equals:

$$\gamma_1 = \psi_1 - \psi_2$$

Once the velocities have been integrated, the global coordinates of various points on the tractor semi-traller can be calculated, using simple goniometry: front axile

$$x_0 = x_1 + l_1 \cos \psi_1$$
$$y_0 = y_1 + l_1 \sin \psi_1$$

articulation point (("5^{th} wheel")

$$x_{1c} = x_1 + l_{1c} \cos \psi_1$$
$$y_{1c} = y_1 + l_{1c} \sin \psi_1$$

2^{nd} trailer axle

$$x_2 = x_{1c} - l_2 \cos \psi_2$$
$$y_2 = y_{1c} - l_2 \sin \psi_2$$

Note that four differential equations have to be integrated, the pose of the vehicle combination is uniquely deficed by x_1, y_1, ψ_1, ψ_2.

C. Controller Design

The Pure Pursuit controller is a popular path tracking algorithm used in autonomous vehicle navigation. It calculates the steering commands necessary to track a predefined trajectory by continuously estimating the vehicle's position and determining the desired look-ahead point on the trajectory.

II. Method

The following steps are used for the controller design-

Defining Waypoints: Specifying the waypoints that define the desired trajectory. These waypoints can be dynamic, meaning they change over time. Then the waypoints are stored as a sequence of (x, y) coordinates in an array.

Vehicle Model:A single track kinematic vehicle model or a tractor semi-trialer kinematic model is used.

Controller Initialization: Initialize the Pure Pursuit controller parameters, such as the look-ahead distance and the maximum steering angle.

Control Loop: In a loop, continuously update the vehicle's position based on global and local sensor measurements in Trucklab. The distance between the current vehicle position and each waypoint was calculated to determine the closest waypoint.

Look-ahead Point Calculation: Determining the look-ahead point by projecting a point from the vehicle's current position along the trajectory. The projection point is selected based on the desired look-ahead distance.

Steering Angle Calculation: Calculating the desired steering angle using the Pure Pursuit algorithm. This involves computing the curvature of the trajectory between the vehicle's position and the look-ahead point and applying a control law to determine the required steering angle.

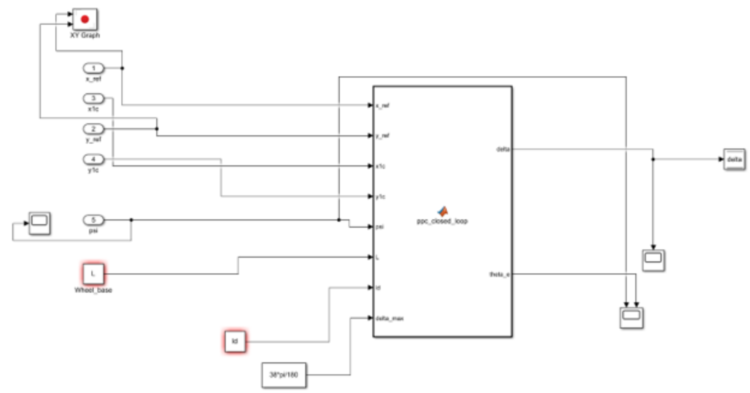


Fig. 3. Pure pursuit controller

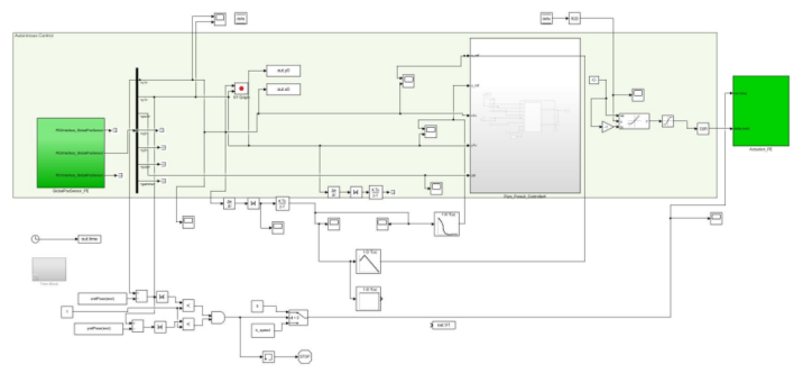


Fig. 4. Simulink model

III. Results

A path following controller has been designed and has been tested on various test paths.The proposed path following controller is suitable for parking and docking maneuvers.

After the implementation of pure pursuit controller along with the vehicle model, we got the plots indicating the robust design of the controller for forward maneuvering of the tractor with semi trailer.

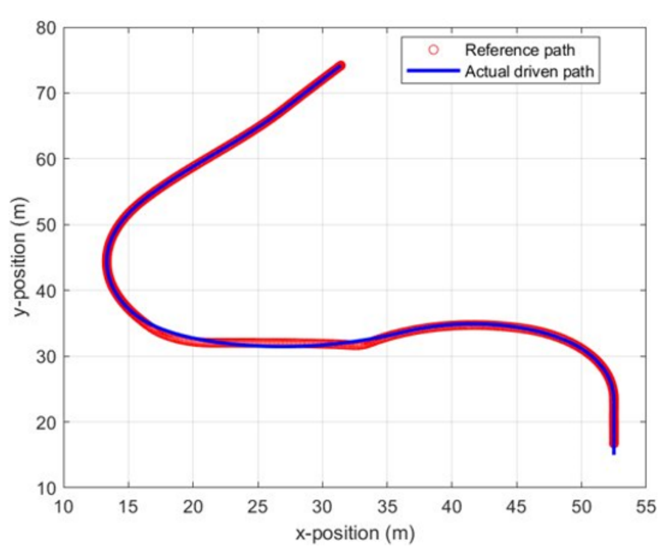


Fig. 5. Forward path following- path(1)

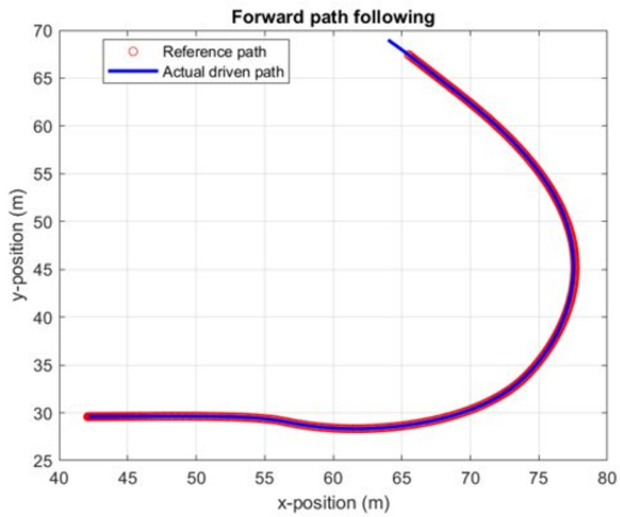


Fig. 6. Forward path following- path(2)

IV. Conclusion

In this paper, the Pure Pursuit controller design for the forward dynamics of an autonomous truck offers an effective and robust solution for accurate path tracking. By continuously estimating the vehicle's position and determining the desired look-ahead point on the trajectory, the controller enables the autonomous truck to navigate along predefined paths with precision.

Through the implementation of the Pure Pursuit algorithm, the controller accounts for the dynamic nature of the waypoints and adjusts the vehicle's steering commands accordingly. This adaptability allows the autonomous truck to effectively track the desired trajectory even when the waypoints change over time.

By successfully implementing the Pure Pursuit controller, autonomous trucks can navigate complex environments, follow designated paths, and safely interact with other vehicles and obstacles on the road. This controller design contributes to the advancement of autonomous transportation systems, leading to more efficient, reliable, and safe operations.

V. References

1. Smith, J.,and Johnson, A. (Year). "Advanced control techniques for autonomous truck trajectory tracking." Journal of Robotics and Automation, 25(4), 123-140.

2. Lee, S., Park, H., and Kim, C. (Year). "Design and implementation of a Pure Pursuit controller for autonomous truck navigation." IEEE Transactions on Intelligent Transportation Systems, 15(2), 456-467.

3. Chen, L., Wang, X., and Li, Y. (Year). "Modeling and control of forward dynamics for autonomous trucks." Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 789-796.

4. Zhang, Q., Liu, X., and Wang, Z. (Year). "Optimal parameter tuning for Pure Pursuit controller in autonomous truck systems." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1-6.

5. Johnson, M., Anderson, R., and Brown, K. (Year). "Adaptive Pure Pursuit controller for dynamic waypoint tracking in autonomous truck applications." IEEE Conference on Decision and Control (CDC), 3000-3005.

6. Kim, J., Lee, D., and Park, S. (Year). "Real-time trajectory generation and Pure Pursuit controller for autonomous truck systems." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 789-796.

7. Li, Z., Zhang, H., and Wu, F. (Year). "Collision prediction and avoidance algorithm for Pure Pursuit controller in autonomous truck navigation." IEEE Transactions on Intelligent Transportation Systems, 18(3), 789-800.

8. Wang, Y., Zhang, J., and Yang, H. (Year). "Pure Pursuit controller with obstacle avoidance for autonomous truck systems." Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 100-107.

9. Liu, C., Liu, Y., and Zhang, G. (Year). "Pure Pursuit controller with dynamic waypoint prediction for autonomous truck navigation." IEEE Transactions on Vehicular Technology, 67(8), 654-667.

10. Xu, H., Chen, W., and Li, S. (Year). "Design and simulation of a Pure Pursuit controller for autonomous truck trajectory tracking." IEEE International Conference on Mechatronics and Automation (ICMA), 400-407.

Assessing the Environmental Impacts of Electric Vehicles and Hydrogen Fuel Cell Vehicles: A Comparative Life Cycle Analysis using the GREET Model

Sai Kirutheka Peram
Automotive Technology
Eidhoven University of Technology
Eindhoven, The Netherlands

***Abstract*—This paper concerns the Life Cycle Assessment review of Electric Vehicles and Hydrogen Fuel Cell Vehicles using the GREET Model approach, and comparing their characteristic advantages and disadvantages. In particular, we first discuss the working principle of the GREET Model, Electric Vehicles, and Hydrogen Fuel Cell Vehicles. Secondly, we analyse the cradle-to-grave life cycle for these powertrains and lastly, compare the results, along with their characteristics. Our results show that on strategically choosing the hydrogen production pathway, Hydrogen Fuel Cell Vehicles are a cleaner choice than Electric Vehicles in terms of overall energy consumption and life cycle emissions. The quantitative results show that Electric vehicles consume 4% higher energy than Fuel Cell Vehicles, whereas the latter emits 36% lower greenhouse gases.**

***Index Terms*—Hydrogen Fuel Cell Vehicle, Electric Vehicle, Life Cycle Assessment, GREET Model**

I. Introduction

In recent years, the automotive system has witnessed an increasing shift towards alternate and sustainable powertrains. This is primarily driven by the detrimental environmental impacts of conventional Internal Combustion Engines (ICEs). The disadvantages include having limited reserves, and an increase in prices. Statistics show that petroleum accounts for 93% of all transportation energy sources, contributing to the majority of greenhouse emissions [1]. As a result, Electric Vehicles (EVs) and Hydrogen Fuel Cell Vehicles (FCVs) are increasingly being used as a solution to tackle the problems of emissions and dependence on fossil fuels [2]. Hydrogen fuel provides a clean, safe, reliable, and affordable energy solution, with characteristic advantages such as energy density, ease of transportation, abundance, and zero or minimal emissions production methods [3]. Further, the current EV technologies, with a battery pack as the primary energy source, are advancing to extend the driving range, increase the powertrain efficiency, and reduce maintenance requirements, and costs by using induction motor drives and permanent magnet brushless motor drives to improve the electric propulsion system, along with the adoption of advanced and fast charging, and power steering systems [4]. Moreover, there have been advancements in Phase Change Materials (PCM) for thermal management, as well as Battery Management System (BMS) for improved performance [5].

Life Cycle Analysis (LCA) has become a commonly used method for evaluating the environmental performance of different vehicle technologies. LCA analyses the overall sustainability of different powertrain options by considering the environmental impacts of the vehicle throughout its entire life cycle, from the raw materials extraction stage to end-of-life disposal. Fig. 1 shows the phases of LCA – goal and scope definition, inventory analysis, life cycle impact assessment, and interpretation, as suggested by the Joint Research Centre of the European Commission. Moreover, the International Organization for Standardization has developed ISO14040 and ISO14044 for LCA. While ISO14040 describes the basic principles and framework for product LCA, ISO14044 specifies the requirements and guidelines for LCA [6].

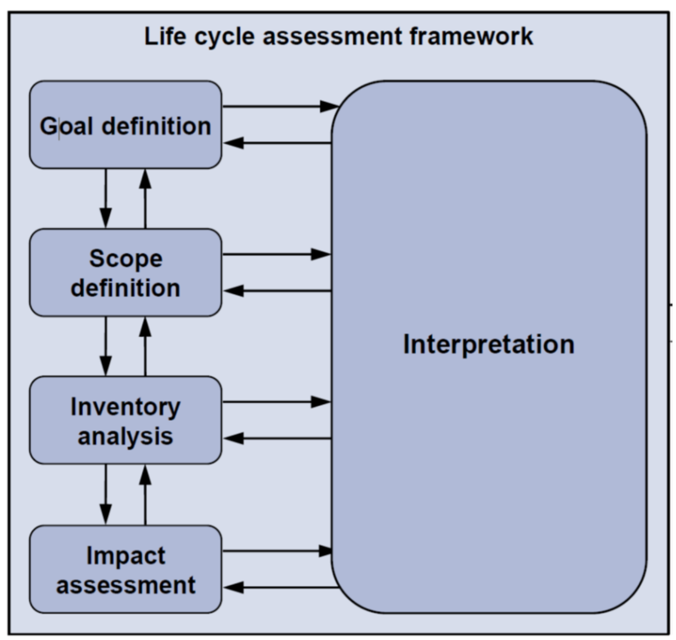


Fig. 1. The Phases of Life Cycle Analysis

In this paper, we aim to compare the LCA results of EVs and FCVs, while also highlighting the key differences between the two powertrain technologies with the conventional ICEs. Toward this end, we will use the GREET (Greenhouse Gases, Regulated Emissions, and Energy Use in Transportation) model, a comprehensive and widely-used LCA tool developed by Argonne National Laboratory. The GREET model provides a robust framework for measuring the environmental impacts associated with vehicle manufacturing, energy production, and fuel production. GREET.Net provides users with a user-friendly, open-access, graphical interface toolbox to perform life cycle analysis simulations of alternative fuels. The GREET model calculates fuel-cycle emissions of volatile organic compounds (VOCs), CO, NO_x, SO_x, and particulate matter measuring 10 μm or less, and three greenhouse gases – carbon dioxide, methane, and nitrous oxide. Furthermore, it also calculates the total fuel-cycle energy consumption, and fossil fuel consumption using various transportation fuels [7]. This tool is therefore able to provide a reliable comparison of EVs and FCVs in order to identify their strengths and weaknesses in terms of environmental sustainability.

In conclusion, the purpose of this research paper is to provide a comprehensive Life Cycle Assessment using the GREET model of EVs and FCVs. By evaluating the environmental performance of these powertrain technologies, we aim to provide policymakers, industry stakeholders, and researchers with valuable insights to inform their decisions regarding sustainable transportation options. Through this research, we hope to contribute to the ongoing shift towards a greener and more sustainable transportation system. Finally, the sections in the remainder of this paper discuss the methodology used, followed by results, and conclusions.

A. Hydrogen Fuel Cell

A Hydrogen Fuel Cell is an electrochemical device that converts the chemical energy of the reduction-oxidation (redox) reactions between Hydrogen and Oxygen into electricity, water and a small amount of heat. The produced electricity is then used as the energy source for FCVs. A fuel cell consists of two electrodes - cathode and anode, immersed in an electrolyte. At the anode, hydrogen undergoes an oxidation reaction,

$$H_2 \rightarrow 2H^+ + 2e^-. \tag{1}$$

This reaction generates positively charged ions, called cations, that migrate towards the cathode through the electrolyte and the free electrons flow through the external circuit. Conversely, oxygen is reduced to water at the cathode by the cations and free electrons,

$$4H^+ + O_2 + 4e^- \rightarrow 4H_2O \tag{2}$$

Comparing a hydrogen fuel cell to a conventional combustion engine, fuel cells can have an energy conversion efficiency of more than 60%, with lower noise during operation, a fast start-up time, and a wide range of operating temperatures [8]. Based on the electrolyte used, fuel cells can further be divided into alkaline fuel cells (AFCs), proton exchange membrane fuel cells (PEMFCs), direct methanol fuel cells (DMFC), phosphoric acid fuel cells (PAFCs), molten carbonate fuel cells (MCFCs), and solid oxide fuel cells (SOFCs). These fuel cells and their characteristics have been listed in Fig. 2

Type	AFCs	PEMFCs	PAFCs	MCFCs	SOFCs
Electrolyte	KOH	Perfluorosulfonic acid ionic exchange membrane	H_3PO_4	Li_2 CO_3-K_2CO_3	Y_2 O_3–ZrO_2
Conductible ions	OH^-	H^+	H^+	CO_3^{2-}	O^{2-}
Fuel	H_2	H_2, CH_3OH	Reformed fuel (CH_4, CO, H_2)	Purified coal gas, natural gas, and reformed fuel (CH_4, CO, H_2)	Purified coal gas and natural gas (CH_4, CO)
Oxidant	O_2	Air	Air	Air	Air
Catalyst	Pt/Ru	Pt/Ru	Pt	NiO	Ni
Operating temperature	65–220 °C	−40–90 °C	150–200 °C	650–700 °C	600–1000 °C
Theoretical voltage	1.18 V	1.18 V	1 V	1.116 V	1.13 V
System efficiency	60%–70%	43%–68%	40%–55%	55%–65%	55%–65%
Application	Special ground and aerospace	Electric vehicle, submarine, and mobile power source	Regional power supply (e.g., power plant)	Power station	Power station
Development	Rapid development at 1–100 kW	Rapid development at 1–300 kW with high cost	Rapid development at 1–200 kW with high cost	Mainly development at 250–2000 kW with short life	Mainly development at 1–200 kW with high preparation technology cost

Fig. 2. Classification and characteristics of fuel cells

B. Electric Vehicles

The main components of an electric vehicle are a potentiometer, Direct Current (DC) Motor, DC Controller, Inverter and a Rechargeable Battery [9]. It operates on the principle of conversion of electrical energy to mechanical energy. As shown in Fig. 3, batteries provide power to the electric motor by converting the current from DC to Alternate Current (AC) using an inverter/power converter. The motor then uses the received power to rotate a transmission and the transmission turns the wheels. Potentiometers are connected between the accelerator and controller to monitor and control the power to be delivered.

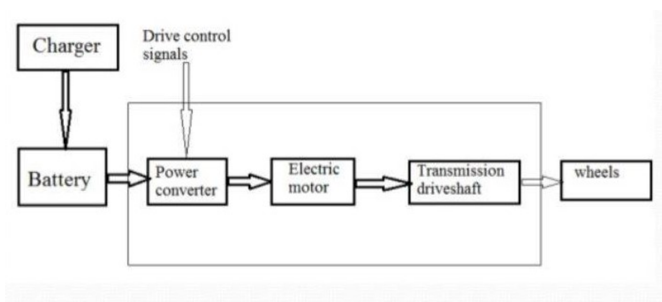


Fig. 3. Architecture of an Electric Vehicle

Comparing EVs to ICEs, EVs have fewer moving parts than ICEs, making them more reliable, increasing efficiency, and reducing noise.

II. Methodology

In this section, we highlight the GREET Model methodology and its advantages, followed by a brief Life Cycle Assessment considerations and assumptions for Electric Vehicles and Hydrogen Fuel Cell Vehicles.

A. The GREET Model

The GREET Model, as mentioned earlier, analyses the emissions associated with both the fuel cycle and the vehicle

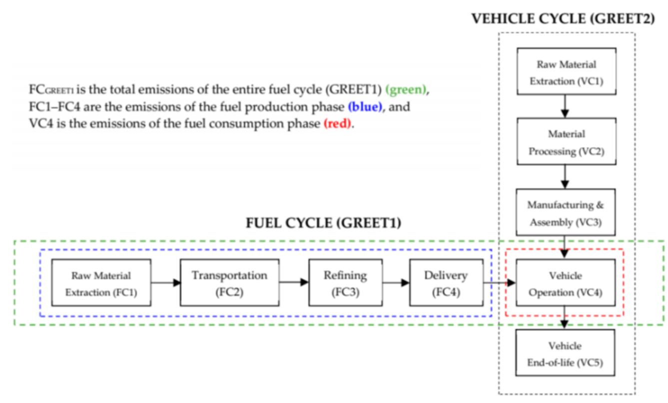


Fig. 4. GREET Model

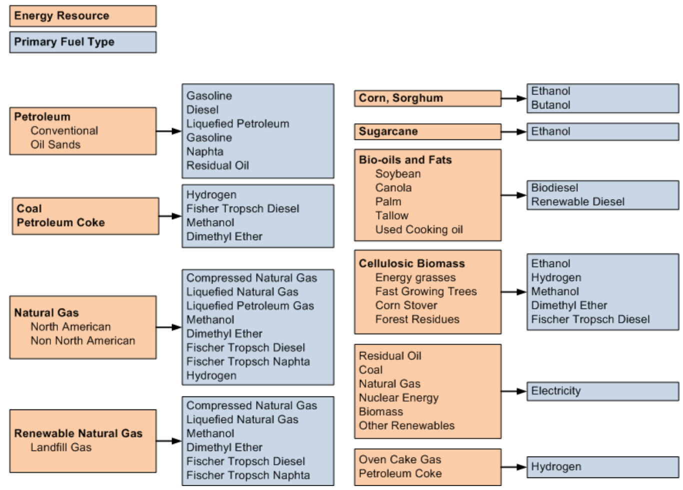


Fig. 5. GREET Fuel Pathway Grouping

cycle. Fig. 4 shows the Fuel Cycle and Vehicle cycle, termed GREET1 and GREET2, respectively.

GREET1 evaluates the well-to-wheel (WTW) energy use while calculating the emissions associated with the recovery of primary feedstock (FC1), transportation of the said feedstock (FC2), production of the fuel from the feedstock and transportation (FC3), distribution of the fuel (FC4), and use of the fuel during vehicle operation (VC4) (or tank-to-wheels [TTW]). The Fuel Cycle part of the model can further be summarised in the form of Eq. 3 below,

$$\begin{aligned} FC_{GREET1} &= FC_{FuelProduction} + FC_{FuelConsumption} \\ &= FC1 + FC2 + FC3 + FC4 + VC4. \quad (3) \end{aligned}$$

Additionally, GREET2 evaluates the energy use while calculating the emissions associated with the production (VC1) and processing of vehicle materials (VC2), manufacturing and assembly of the vehicle (VC3), and the end-of-life disposal and recycling of vehicle components (VC5) [10]. The Vehicle Cycle part of the model can further be summarised in the form of Eq. 4 below,

$$\begin{aligned} VC_{GREET2} &= VC_{VehicleProduction} + VC_{VehicleOperation} + \\ & VC_{VehicleEndOfLife} \\ &= VC1 + VC2 + VC3 + VC4 + VC5. \quad (4) \end{aligned}$$

Moreover, the GREET Model contains more than 100 fuel pathways including petroleum fuels, natural gas fuels, biofuels, hydrogen and electricity, all produced from various energy sources. Fig. 5 shows the grouping of pathways by feedstock, used in the model.

B. Analysis of Electric vehicles

The main focus for the Life Cycle Assessment of electric vehicles in this paper is on Tesla Model 3, setting the lifetime driving distance to 150,000km. This is due to the New European Driving Cycle (NEDC) consideration. Fig. 6 shows the vehicle life cycle system boundary for Tesla Model 3 [11]. It consists of Raw Material Production, Vehicle Production, Power Battery Production, and the Usage Stage. A total mass of 1860.3 kg of the vehicle's material is covered in this study, including the materials of the main system, lead acid battery, liquids, and tires for the raw materials acquisition stage. Additionally, the production stage includes the production of the vehicle from the raw materials until the delivery of the finished product. Furthermore, the paper considers the energy consumption during the assembly process, such as welding, painting, and stamping, as well as the consumption of energy during battery production. Finally, the use stage of the vehicle includes energy and material consumption. The electricity consumption of a Tesla Model 3 is known to be around 16kWh/100km with zero carbon dioxide emissions during this stage. The use stage analysis also includes the replacements of tires, batteries, etc.

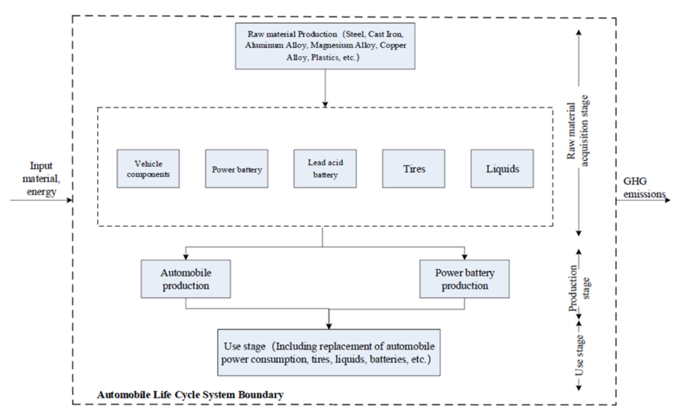


Fig. 6. Tesla Model 3 Life Cycle System Boundary

C. Analysis of Hydrogen Fuel Cell Vehicles

Toyota Mirai is taken to be the hydrogen fuel cell counterpart for the Life Cycle Assessment in this paper. Although the assessment was done for both the Japanese driving cycle (JC08) and the NEDC, only the NEDC of 150,000km is considered in this study. Fig. 7 shows the system boundary for this study, which includes material production, vehicle production, maintenance and replacement of battery, tires and engine oil, and end-of-life stages. However, note that the remnant raw materials disposal and recycling processes are not included in the analysis [12]. Lastly, the processes involved in the end-of-life recycling and disposal of the vehicle include the transportation of the vehicle, destruction of freon, dismantling of parts fuel cell stack, airbags and tires, shredding, thermal energy recovery, incineration and landfill. The hydrogen production pathways used in the UK, Germany and Denmark, where the vehicle was first launched, are used in this study. These pathways include central reforming from natural gas piped from the North Sea in Russia, and electrolysis using renewable energy from wind power.

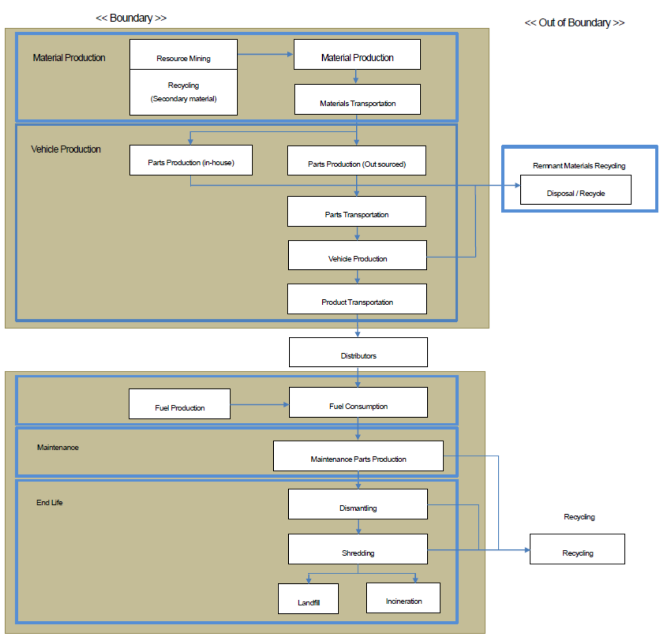


Fig. 7. Toyota Mirai Life Cycle System Boundary

III. Result

The results of the GREET analysis for both the electric vehicles and the hydrogen fuel cell vehicles are highlighted and discussed in this section of the paper. The comparison between various powertrains of multiple vehicles, including Gasoline, Compressed Natural gas (CNG), Electric Vehicle (EV), and Fuel Cell (FC) have been concluded as in Fig. 8 and Fig. 9. The later sections in this paper have been used to discuss the results of the main focus of this paper, i.e., EVs and FCVs in further detail.

A. Results of Electric Vehicles

Through the GREET Model analysis, results for Tesla Model 3, the Electric Vehicle representative in this paper is

Vehicle Type	Energy consumption (J/m)
Gasoline-fueled	3912.11
CNG-fueled	4554.19
FCV	699.99
EV	725.51

Fig. 8. Powertrain-wise energy consumption for similar driving conditions

Pollutant Particles	Vehicle			
	CNG	EV	Gasoline	FC
CO_2	29.927	12.334	32.964	7.953
GHG	34.923	13.328	35.012	8.542
CO	0.106	0.011	0.097	0.002
NOx	0.022	0.010	0.018	0.003
SOx	0.006	0.026	0.007	0.003

Fig. 9. Total emitted particles per 100km of vehicles' travel (kg).

visualized in Fig. 10 and Fig. 11. The Greenhouse Gas (GHG) Emissions for the Usage Stage are higher, accounting for about 81% of the total GHG emissions, followed by the Raw Material Acquisition Stage of 16%, and only the remaining 3% is attributed to the Production Stage. As a result, the emissions per unit distance are mainly dependent on the Usage Stage for Electric vehicles [11]. To conclude, the Tesla Model 3 life cycle GHG emissions are about 376gCO_2e/km.

Life cycle stage	GHG emissions	Unit
Raw material acquisition stage	62	gCO_2e/km
Production stage	10	gCO_2e/km
Use stage	304	gCO_2e/km
Total	376	gCO_2e/km

Fig. 10. Greenhouse Gas Emissions of Tesla Model 3 at each Life Cycle Stage

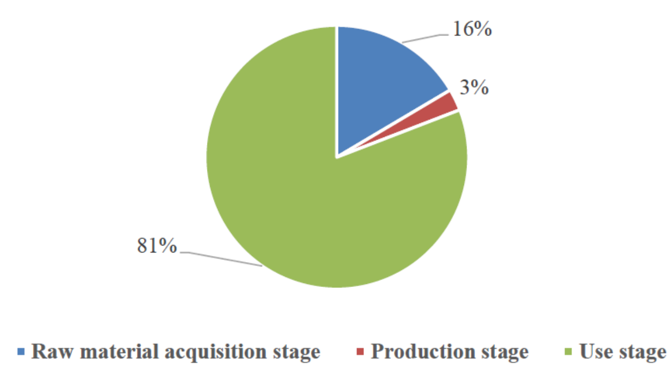


Fig. 11. Proportion of Greenhouse Gas Emissions of Tesla Model 3 at each Life Cycle Stage

B. Results of Hydrogen Fuel Cell Vehicles

The GREET analysis for Toyota Mirai, the Hydrogen Fuell Vehicle representative in this paper, can be illustrated as done in Fig. 12, Fig. 13 and Fig. 14. Fig. 12 shows the consumption of three major nonrenewable energy sources in each stage of the life cycle for Toyota Mirai. It is seen that the consumption of raw coal is much higher than the other sources (around 94% of total raw coal consumption), especially in the Usage Stage and the Scrap Recycling Stage [13]. This is due to the large amount of electricity required in the production of Hydrogen and in recycling the end-of-life product. Further, the most energy-intensive parts during the material acquisition stage are the vehicle body and chassis. Moreover, the fuel cell stack consumes the most energy during the production stage due to a very complex multi-step manufacturing process. To conclude, Fig. 13 suggests that the highest fossil fuel energy consumption for an FCV is at the Use Stage in the life cycle of the vehicle.

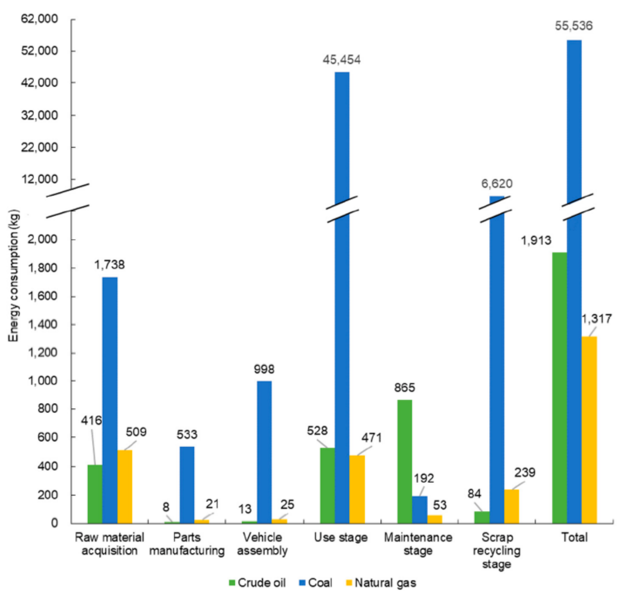


Fig. 12. Energy consumption of Toyota Mirai at each Life Cycle Stage

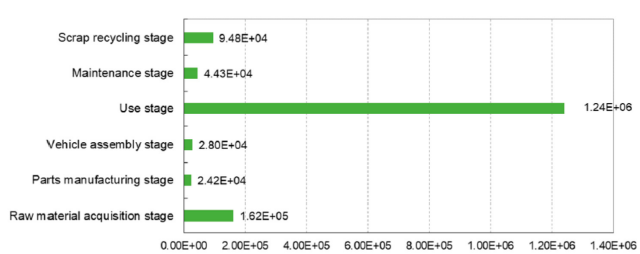


Fig. 13. Fossil Fuel Energy Consumption of Toyota Mirai at each Life Cycle Stage

Fig. 14 show the emissions of Toyota Mirai, including Non-methane volatile organic compounds (NMVOC), Methane (CH_4), Particulate Matter smaller than 10 microns, Particulate Matter smaller than 2 microns, etc. The data suggest that CO_2 is the largest of all emissions. This may be due to the heavy dependence on coal combustion to produce power in China, where this study has been conducted [13]. The NMVOC emissions are the highest during the raw material acquisition stage due to the various processes required to obtain Platinum for the fuel cell stack. In addition, the negative values in the figure show the positive benefits of recycling on emissions. Moreover, a majority of the emissions are due to the vehicle body during the raw material acquisition, and due to the fuel cell stack during the manufacturing stage.

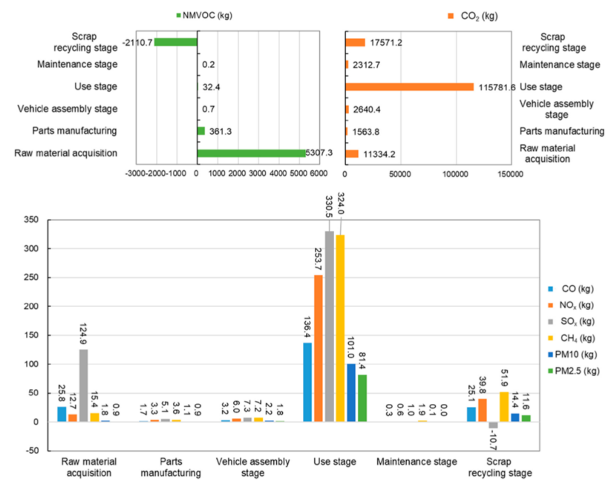


Fig. 14. Emissions of Toyota Mirai at each Life Cycle Stage

IV. Conclusion

In this paper, the Life Cycle Assessment review of Electric Vehicles and Hydrogen Fuel Cell vehicles is carried out, using a cradle-to-grave GREET Model analysis approach. Comparing various studies for EVs and FCVs, it can be presumed that the overall energy consumption is 4% higher for EVs as compared to FCVs, which are both lower than conventional Gasoline-powered vehicles. Moreover, the overall emissions are 36% lower for FCVs than EVS, with FCVs having the least CO_2 among all the powertrains [5]. The emissions for both, are again, much fewer than gasoline-powered and CNG-powered vehicles. Furthermore, FCVs have close to zero acid rain causing NO_x and SO_x emissions. These results can be attributed to the emissions related to electricity and hydrogen production for EVs and FCVs, respectively. Although, it is to be noted that the energy consumption and emissions of hydrogen production depend on the chosen production pathway of the hydrogen used as fuel [13]. Therefore, the CO_2 emissions in the fuel consumption phase for FCVs is zero, whereas it depends on the fuel production mechanism in the fuel production phase. Emissions from the natural gas production pathway for Hydrogen are 6 times higher than the renewable energy production pathway.

In conclusion, Hydrogen Fuel Cell Vehicles stand to be the cleaner powertrain option as compared to Electric Vehicles, if it is strategically produced from a low-carbon energy source or solar-thermal processes [14]. For future scope, advancements in engine optimisation and onboard hydrogen storage will make Hydrogen the perfect decarbonisation solution in the automotive industry [15]. The task at hand is to extend this study to further include not only passenger vehicles but also heavy-duty vehicles.

REFERENCES

[1] S.C. Davis, S.W. Diegel, and R.G. Boundy. Transportation Energy Data Book, U.S. Department of Energy, Oak Ridge National Laboratory, vol. 31., 2012.

[2] C. Acar and I. Dincer, "Testing and performance evaluation of a hybrid photoelectrochemical hydrogen production system." International journal of hydrogen energy, vol. 42., pp. 3605-3613, February 2017.

[3] C. Acar and I. Dincer, "The potential role of hydrogen as a sustainable transportation fuel to combat global warming." International journal of hydrogen energy, vol. 45., pp. 3396-3406, January 2020.

[4] C.C. Chan and Y.S. Wong, "Electric vehicles charge forward." IEEE Power and Energy Magazine, vol. 2., pp. 24–33, November 2004.

[5] A. Teimouri, K.Z. Kabeh, S. Changizian, P. Ahmadi and M. Mortazavi, "Comparative lifecycle assessment of hydrogen fuel cell, electric, CNG, and gasoline-powered vehicles under real driving conditions." International journal of hydrogen energy, vol. 47., pp. 37990-38002, 2022.

[6] M. Finkbeiner, A. Inaba, R. Tan, K. Christiansen and H.J. Klüppel, "The new international standards for life cycle assessment: ISO 14040 and ISO 14044." The international journal of life cycle assessment, vol. 11., pp.80-85, 2006.

[7] M.Q. Wang, "GREET 1.0 – Transportation fuel cycles model: Methodology and use." United States, June 1996.

[8] L. Fan, Z. Tu and S.H. Chan, "Recent development of hydrogen and fuel cell technologies: A review." Energy Report, vol. 7., pp. 8421–8446, August 2021.

[9] E.Y.C. Wong, D.C.K. Ho, S. So, C. Tsang and E.M.H. Chan, "Life Cycle Assessment of Electric Vehicles and Hydrogen Fuel Cell Vehicles Using the GREET Model—A Comparative Study." April 2021.

[10] D. Barapatre, C. Kanfade, A. Bari, R. Nile, R. Wagh and R. Nimbalkar, "Electric Vehicle." International Journal of Engineering Research and Technology, 2016.

[11] G. Gui, "Carbon Footprint Study of Tesla Model 3." E3S Web of Conferences, vol. 136, 2019.

[12] T. Ozawa, T. Yoshida, S. Aida, L. Oya and J. Tokieda, "The MIRAI Life Cycle Assessment Report." Toyota Motor Corporation, June 2015.

[13] Y. Chen, X. Hu and J. Liu, "Life Cycle Assessment of Fuel Cell Vehicles Considering the Detailed Vehicle Components: Comparison and Scenario Analysis in China Based on Different Hydrogen Production Schemes". Energies, August 2019.

[14] M. Miotti, J. Hofer and C. Bauer, "Integrated environmental and economic assessment of current and future fuel cell vehicles." Int J Life Cycle Assess, vol. 22., pp. 94–110, 2017.

[15] D. Candelaresi, A. Valente, D. Iribarren, J. Dufour and G. Spazzafumo, "Comparative life cycle assessment of hydrogen-fuelled passenger cars." International Journal of Hydrogen Energy, vol. 46., pp. 35961-35973, October 2021.

Requirements engineering of power electronics modelling in electric vehicle applications

M. den Besten
m.d.besten1@student.tue.nl, 1494503

***Abstract*—In this paper a literature study is done on the requirements for power electronics modelling in electric vehicle applications. The main focus will be on the battery to electric machine interface. The main current practices in power electronics modeling are discussed. Important parameters which are relevant for the simulation of the power converter, when it comes to machine control, battery to machine interactions, manufacturer requirements and component level interactions, are defined. These parameter requirements are implemented and validated in an H-bridge inverter simulation model and a simple and accurate model for this circuit is presented, which fulfills most of the requirements for engineering power electronics.**

I. Introduction

As the demand for electrification of the automotive industry is growing and vehicle manufacturers are focusing more and more on the production of electric vehicles [1], the demand of power electronics in this discipline is growing. With this demand, the demand for models and simulation methods suitable for electric vehicle application is growing too [2]. Therefore, in this paper research into the requirements for modelling practices for electric vehicle application is done. First, the current simulation practices are discussed after which the parameter requirements from literature are presented. This is implemented on a model of a full bridge circuit, after which conclusions are drawn.

II. Current Simulation Practices

To get proper requirements for the simulation of power electronics in electric vehicle applications, first the best simulation method and software has to be selected. When it comes to the simulation of electronic circuits, and specifically power electronic circuits, there are two main practices currently used. First, there are circuit level simulators, which try to take into account as much of the dynamics of the components as possible, from which SPICE (or derivatives from that like PSpice, ltSpice, etc.) is the most used. Second, there are simulation tools which simplify the behaviour of the components somewhat, like PLECS, Simscape toolbox for MATLAB and plain Simulink.

A. SPICE

Software based on SPICE can perform nonlinear dc, nonlinear transient, and linear ac analyses. Next to that there are a lot of detailed models available for individual components such as semiconductors. This give SPICE the possibility to accurately capture all the transient events in a circuit. This makes SPICE a very good method to simulate small size power systems.

The disadvantage from using SPICE is however, that, because all the transients are captured accurately, the simulation becomes rather slow. This means that large circuits or longer simulation periods are hard to simulate or will take a lot of time. Next to that, there are components, especially power components, which do not have SPICE models available. [3] [4] [5]

B. Simplified modelling

Other software uses simplified models of switches and components. While with this method ideal switches are used, often models are available with which switching losses and temperatures in components can be modelled. With this method of simulating power electronic circuits large and complex topologies can be simulated for extended simulation periods. When MATLAB is used as the Simulation engine interactive control and extensive post-processing options are offered. Next to that Matlab gives a lot of opportunities to implement controllers for the circuit. These advantages do come at the cost of reduced accuracy of the modelling of transient events. [3] [4] [5] [6]

C. Conclusions

It can be concluded that for small size power converters, where transient events have to be known accurately, SPICE is the best method for simulating a circuit. But for larger or more complex circuits, where longer simulation periods are needed to, for example, test controllers, simplified modelling methods are more suited. This is because the simplified modelling gives still an accurate insight in the behaviour of the circuit and (when MATLAB is used) gives a very broad range of options for designing controllers and using libraries and premade blocksets.

III. Parameter requirements

To design power electronics for electric vehicle applications, first the requirements have to be determined these requirements are divided in manufacturer, control, component and interface requirements. These will be presented below and after that some annotations on the reliability of electronics in automotive applications will be given.

A. Manufacturer requirements

The major part of the requirements for the design of power electronics for a vehicle will be given by the manufacturer as that is the stakeholder with the most interest in the characteristics of the vehicle. First off all, the cost needs to be made as low as possible to make the product attractive for the customer. Second, the volume and weight of the electronics have to be made as small as possible, as a vehicle has only a limited amount of space and additional weight will reduce the efficiency of the vehicle. Next, the manufacturer will strive for electronics to be as efficient as possible. [7] Also there are a lot of regulations (i.e. [8], [9] and [10]) which have to be taken into account and influence the design of the power electronics and their electromagnetic compatibility (EMC)(The EMC will however, not be taken into account into this paper as this requires a whole different modelling approach). These safety regulations also call for safety mechanism in the electronics like over-current, over-voltage and over-temperature protections and also protections for the devices connected to the power converters (which is discussed in more detail in Sections III-D and III-E). Further, the manufacturer will desire a long lifetime of the electronics, which means that they (especially the semiconductors) have to be tested with different drive-cycles and the thermal- and load-cycling has to be limited, such that the lifespan of the semiconductor can be extended [11] [12]. [13] Lastly the measurability in real vehicles of the parameters of interest needs to be taken into account. It needs to be thought out if the currents, voltages and temperatures which needs to be known indeed can be measured (while still taking into account all the requirements) in a real system.

B. Control

To achieve the high efficiency which is required as discussed in Section III-A the control strategy has to be adapted upon that. One of the main strategies to reduce the losses in a power converter is to use soft-switching. It will give a huge reduction in switching losses [14] [15] and it will therefore also require less heatsinks, which on its turn reduces the size of the converter. So for an efficient and power-dense converter, soft-switching hast to be utilized as much as possible. This might however come at the cost of more complex control to be required, but there are, on the other hand, plenty of control schemes available to achieve soft-switching in a wide range of topologies.

C. Component-level

To achieve the requirements for a high efficiency, small volume and weight and a long lifetime it is needed that all components are optimized for this. This means that semiconductors, which are a key component in vehicle applications [7], with small forward voltages and on-resistances have to be chosen. Also the bandwith of the transistor is very much of interest, as with an increased switching frequency the efficiency of a converter can be increased a lot and magnetic components (which determine for a large part the size of a converter) can be made much smaller [1]. This means that new and promising technologies such as silicon carbide or gallium nitride tranistors have to be considered for the design of power electronics. Also the design of busbars and printed circuit boards (PCB's) are very much of interest, because they have a large influence on the EMI behaviour of a converter.

D. PE-to-battery

Currently the mainly used battery type is a Li-ion or lithium based batteries, because of their high energy density, low maintenance, and a variety of shapes, chemistries, and performances available for electrical applications [16] [17]. As these batteries are heavily affected by charge and discharge currents and operating temperature [18], the power converters connected to these batteries need to have strict limits to operate in, to not damage the battery. In practice that means that there has to be an upper limit on both the voltage and current supplied and requested from the battery. And also that the amount of charge supplied to the battery does not exceed the maximum capacity of the battery, which means there has to be another path for the current to flow when the battery is fully charged and the vehicle still needs to brake.

E. Power electronics to machine interactions

The requirements for the interactions with the electric machine of the vehicle, depend mainly on the type of machine that is used. It can be assumed that a permanent magnet synchronous machine (PMSM) will be used, as these give a significant higher efficiency and higher torque than induction or DC machines. [19] With this higher torque also a higher current demand comes. This means that the power converter driving this machine has to be able to handle the currents drawn by the machine and also supplied by the machine when regenerating. Next to voltage, the inverter has to be able to handle voltages supplied and requested by the machine. Lastly there should also be a current path in case the inverter fails, such that the machine is slowed down and the vehicle comes to a quick stop in a fault scenario. [11]

F. Reliability

With all these different requirements stated it has to be taken into account that in automotive applications the reliability of the electronic components will be less than in most other industries [20] and therefore larger margins of all components have to be taken. This is because in automotive industry parts are placed in an operating environment where the temperature can swing between -40 °C and 155 °C [21] and humid conditions can be present.

IV. Validation

Based on the findings from Sections II and III a model of a inverter is made. The circuit modelled is a H-bridge with an inductor and resistor in series connected to its output. The H-bridge is modelled using plain Simulink and the model is shown in Figure 1. The inductance L is taken as 19.6 H, the resistance R as 470 Ω and the input voltage U_{DC} as 560 V.

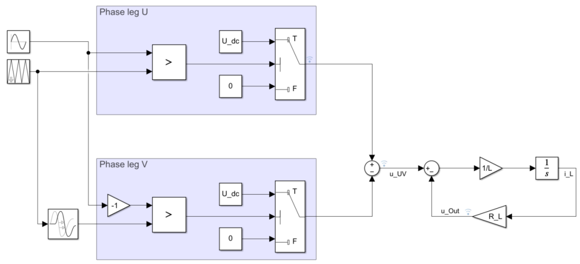


Fig. 1. Simulink model of the full bridge circuit

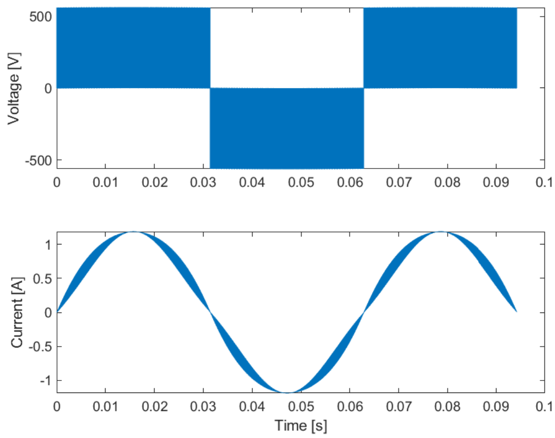


Fig. 2. Simulation output using unipolar switching

Using this model which is simplified with ideal switches, the peak currents and voltages of both the input and the output of the converter and all the components can be evaluated. In Figure 2, the node voltage of the H-bridge and the current through the load is shown with a sinusoidal input. It can be clearly seen that the ripples in the signals are simulated well and this gives a good insight on the loading of the components.

The model was simulated for a period of 1 second and it took only a few seconds to finish the simulation, which means that with the presented model a controller can also be easily implemented. After state space averaging was applied to the circuit, the transfer function of the plant was obtained and a PI controller was tuned for this plant (with gains $K_P = 0.141$ and $K_I = 1579.2$). The step response of the closed loop system is shown in Figure 3.

Next, the controller was applied to the real circuit and again the step response was measured. The step response of the physical system is shown in Figure 4. If this is compared with the response shown in Figure 3 it can be seen that both the rise time and the ripple of the current are very well matched. Also, even though the switches are modelled as ideal switches, no difference between the currents at the switching moment can be noticed.

A flaw of the modelling approach for this full bridge circuit is that there is no information of temperatures present. And

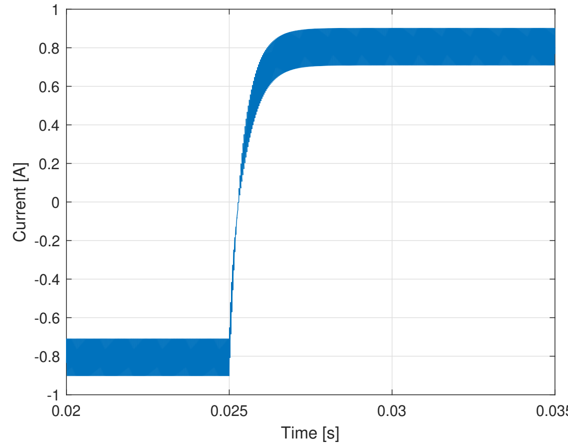


Fig. 3. Simulated step response of the controller with a PI-controller

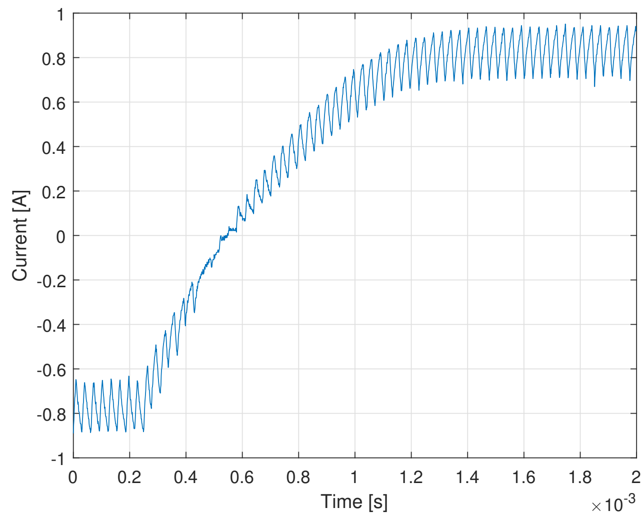


Fig. 4. Measured step response of the controller with a PI-controller

because the switches are modelled as ideal switches, their losses are unknown and it is therefore very hard to estimate their temperatures and with that their wear and aging.

V. Conclusions

In conclusion, the most adequate method for simulating power electronics in electric vehicle applications is the simplified modelling approach. After simulating a simple full bridge circuit it appeared that currents and voltage can be accurately simulated and all the transients are accurately captured. With this information a choice for the components in the power converter can be made. The only important parameter which the simulation does not capture is the temperature, which therefore requires another simulation method (like PLECS) or testing on the physical system.

As this is only tested on a very simple circuit with an inductive-resistive load, more testing has to be done if this method for simulating power electronics is still accurate in more complicated circuits and if the transient events in the circuit are still captured accurate enough to make a good design for a power converter.

REFERENCES

[1] A. Emadi, Y. J. Lee, and K. Rajashekara, "Power electronics and motor drives in electric, hybrid electric, and plug-in hybrid electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2237–2245, 2008.

[2] M. Amrhein and P. Krein, "Dynamic simulation for analysis of hybrid electric vehicle system and subsystem interactions, including power electronics," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 825–836, 2005.

[3] H. Le-Huy and G. Sybille, "Matlab/simulink and pspice as modelling tools for power systems and power electronics," in *2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134)*, vol. 2, 2000, pp. 766–767 vol. 2.

[4] T. Zhao and Q. Wang, "Application of matlab/simulink and pspice simulation in teaching power electronics and electric drive system," in *2005 International Conference on Electrical Machines and Systems*, vol. 3, 2005, pp. 2037–2041 Vol. 3.

[5] M. Darwish, B. R. Alamri, and C. Marouchos, "Orcad vs matlab simulink in teaching power electronics," in *2015 50th International Universities Power Engineering Conference (UPEC)*, 2015, pp. 1–5.

[6] S. Khader, A. Hadad, and A. A. Abu-aisheh, "The application of psim matlab/ simulink in power electronics courses," in *2011 IEEE Global Engineering Education Conference (EDUCON)*, 2011, pp. 118–121.

[7] B. Hellenthal, "Power electronics - key to the next level of automotive electrification," in *2012 24th International Symposium on Power Semiconductor Devices and ICs*, 2012, pp. 13–16.

[8] V. Rodriguez, "Automotive component emc testing: Cispr 25, iso 11452–2 and equivalent standards," *IEEE Electromagnetic Compatibility Magazine*, vol. 1, no. 1, pp. 83–90, 2012.

[9] C. Fanning, "An overview of automotive emc standards and emerging requirements," in *2018 IEEE Symposium on Electromagnetic Compatibility, Signal Integrity and Power Integrity (EMC, SI PI)*, 2018, pp. 1–43.

[10] D. Bululukova and M. Kramer, "Application of existing wireless power transfer standards in automotive applications," in *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, 2014, pp. 863–864.

[11] M. Helsper and N. Rüger, "Requirements of hybrid and electric buses - a huge challenge for power electronics," in *2014 16th European Conference on Power Electronics and Applications*, 2014, pp. 1–11.

[12] J. Zhang, Z. Qiu, E. Zhang, and P. Ning, "Comparison and analysis of power cycling and thermal cycling lifetime of igbt module," in *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, 2018, pp. 876–880.

[13] T. Riipinen, V. Vaisanen, M. Kuisma, L. Seppa, P. Mustonen, and P. Silventoinen, "Requirements for power electronics in solid oxide fuel cell system," in *2008 13th International Power Electronics and Motion Control Conference*, 2008, pp. 1233–1238.

[14] T. Halder, "Hard and soft switching geometries for operations of the mosfet used for the smps," in *2021 Devices for Integrated Circuit (DevIC)*, 2021, pp. 1–6.

[15] S. Tiwari, J. K. Langelid, O.-M. Midtgård, and T. M. Undeland, "Hard and soft switching losses of a sic mosfet module under realistic topology and loading conditions," in *2017 19th European Conference on Power Electronics and Applications (EPE'17 ECCE Europe)*, 2017, pp. P.1–P.10.

[16] F. H. Gandoman, A. Ahmadi, P. Van den Bossche, J. Van Mierlo, N. Omar, A. E. Nezhad, H. Mavalizadeh, and C. Mayet, "Status and future perspectives of reliability assessment for electric vehicles," *Reliability Engineering & System Safety*, vol. 183, pp. 1–16, 2019.

[17] N. Ghaviha, J. Campillo, M. Bohlin, and E. Dahlquist, "Review of application of energy storage devices in railway transportation," *Energy Procedia*, vol. 105, pp. 4561–4568, 2017.

[18] F. H. Gandoman, T. Kalogiannis, M. Berecibar, J. V. Mierlo, and P. V. d. Bossche, "Quantitative analysis techniques for evaluating the reliability of li-ion battery: challenges and solutions," in *2020 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2020, pp. 1–5.

[19] T. Finken, M. Felden, and K. Hameyer, "Comparison and design of different electrical machine types regarding their applicability in hybrid electrical vehicles," in *2008 18th International Conference on Electrical Machines*, 2008, pp. 1–5.

[20] C. M. Tan, U. Narula, and D. Kapoor, "Reliability paradox for worldwide automotive electronics," in *2017 Annual Reliability and Maintainability Symposium (RAMS)*, 2017, pp. 1–7.

[21] V. v. Tils and S. Zsembinszki, "Design requirements for automotive reliability," in *European Solid-State Device Research Conference*, 2006.

Facilitating Safe Perception of Objects For Autonomous Trucks Within a Distribution Center

L.A.J.M. Thelen

Abstract—**Autonomous trucks promise to revolutionize the transportation chain. An improvement of transport efficiency, reduction of emissions, solution for personnel shortage and reduction of work-related fatalities. To implement automated driving, it is of high importance to meet functional safety requirements. The traditional functional safety approach of following ISO 26262 and/or ISO/PAS 21448 is claimed to be insufficient for autonomous driving according to new competing standards like UL4600. In this paper, we apply the traditional standards to an example system and assess the functional safety concept. Moreover, the research explores potential shortcomings of the traditional standards by evaluating the safety claim using UL4600.**

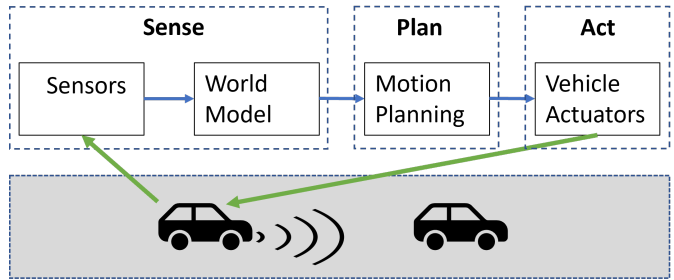


Fig. 1. Simplified autonomous vehicle system architecture.

I. Introduction

THE European freight market is forecast to be worth €431.360 million with a projected real compound annual growth rate of 3.0% from 2021 to 2026 [1]. On a global scale, the parcel shipping volume is expected to reach approximately 262 billion parcels in 2026 [2]. These developments lead to enormous challenges in the worldwide transportation sector, e.g., continuing to support the increasing demand for transportation services, managing personnel shortages and reducing environmental impact [3]. Adoption of autonomous trucks within the logistics sector is considered as a major contribution in improving transport efficiency, reducing pollutant emissions, lowering total costs of ownership and enhancing safety [4]. The potential safety improvements can reduce the total work-related injuries significantly. According to the U.S. Department of Labor, 7310 nonfatal injuries and 201 deaths are caused by non-roadway accidents involving motorized land vehicles in 2020 [5]. Moreover, 337 work-related deaths in 2021 are the result of collisions between vehicles and pedestrians [6]. Since in 94% of all crashes, the critical reason is assigned to the human driver [7], autonomous vehicles can play a key role in reducing accidents and injuries.

A simplified autonomous vehicle system architecture is shown in Figure 1. The system consists of three high-level tasks: Sense, Plan and Act. Sensor inputs are used for creating a world model, which includes subtasks like sensor fusion or object detection. The 'Plan' action is dedicated for motion and/or direction planning and the last 'Act' step performs the actuation of vehicle in terms of steering angle, throttle position etc. [8].

Although much research is done on each of these system tasks, i.e., route optimization [9], improving logistics management [10], or energy consumption [11] of autonomous driving trucks, less research is done on ensuring safety of these complex systems by evaluation of the current automotive standards.

Moreover, some researchers claim that the current accepted Automotive standards ISO 26262 and ISO/PAS 21448:2019 are not sufficiently covering safe operation of autonomous vehicles [12]. Instead of using the traditional prescriptive approach, they suggest the UL4600 standard, to add another layer of extension to the traditional standards. Where traditional standards tell engineers how to do safety, UL4600 tells engineers what a safety case should address.

This paper focuses on the functional safety of the perception system in a distribution center. The research explores the potential shortages of the automotive standards ISO 26262 and ISO/PAS 21448:2019 for object detection by applying them to an example distribution center. Eventually, the main research question of: "Does compliance with the ISO 26262 and ISO/PAS 21448 standards provide enough safety for safe object detection in a distribution center?". This paper first presents an overview of the state-of-the-art in terms of perception system technology and relevant functional safety standards. Then, a safety analysis and assessment based on the considered standards is applied to the object detection system of TruckLab (Figure 2), a down-scaled version of the Jumbo distribution center in Veghel, the Netherlands. In section IV the results obtained from the safety analysis and assessment are presented. section V discusses the outcome and points out potential shortcomings of the current standards to evaluate and assess safety. A brief summary of the report is presented in section VI.

II. Overview of automotive perception systems and safety standards

This section aims to provide a brief overview on state-of-the-art perception systems and functional safety standards, to obtain a basic understanding that is needed before starting functional safety engineering in subsection III-B.

Fig. 2. TruckLab environment with two autonomous trucks parked at a loading dock.

A. Perception systems in autonomous driving

Perception in a self-driving architecture is the stage dealing with information from sensors and turning it into meaningful data for different high-level tasks like navigation and motion planning [13]. Autonomous vehicles use multiple sensors, computational hardware and software like presented in [14] & [15].

State-of-the-art perception systems are usually based on combinations of the following sensor types [16], [17]:

1) LIDAR: A sensing technique that is based on transmitting and receiving light pulses to obtain distance information to other objects. By transmitting a light pulse, nearby objects reflect the pulse back to the LIDAR. From the difference in time between the transmitted and received pulse, an estimation about the object's distance to the LIDAR can be made. Dependent on the LIDAR type, a 1D, 2D or 3D environment mapping can be created by means of point cloud data in combination with post processing software.
2) RADAR: A distance-based sensing technique that makes use of the transmission and detection of electromagnetic waves. Current state-of-the-art automotive RADARs apply the principle of phase or frequency modulation in a continuous wave fashion in the millimeter wave domain. The modulation techniques are applied against interference caused by other transmitters. Moreover, radar enables to estimate the relative speed of the identified objects by applying the Doppler effect.
3) Cameras: A sensing technique based on detecting light signals emitted from objects in the nearby environment. The light signals are converted to electrical signals by means of a 2D array of photosites, a circuit consisting of photodetectors and other electronics. A camera in combination with object detection software, is able to detect stationary and moving objects within its field of sight.

B. Traditional functional safety standards in automotive

Functional safety is the absence of unreasonable risk due to hazards caused by malfunctioning behaviour of electrical and electronic systems [18]. Currently, two accepted standards in automotive are used: ISO 26262 and ISO/PAS 21448. The major difference between these standards is that ISO 26262 provides guidelines to prevent control and/or mitigate safety hazards due to a system failure, whereas ISO/PAS 21448, or SOTIF (Safety Of The Intended Functionality) focuses on preventing, controlling and/or mitigating safety hazards that can arise in the absence of a system failure.

ISO 26262 consists of ten parts and defines the vehicle safety as the absence of unreasonable risks that arise from malfunctions of the Electric and Electronic system in part 1. Unreasonable risks are interpreted as risks judged by society to be morally unacceptable in a certain context according to valid societal moral concepts. ISO 26262 provides requirements and recommendations to avoid and control random hardware failures and systematic failures in its second part. Whereas the third part specifies a Hazard Analysis and Risk Assessment. As example, the situation in which a camera-based perception system's lens breaks due to mechanical stress on an autonomous vehicle [19], is covered in the ISO 26262 standard. Whereas a situation in which the same perception system suddenly is blinded by an approaching vehicle, is not. The fundamental difference between these cases is that the perception system in the first situation experiences a system failure, but does not in the second situation.

With current the automotive trend towards more Advanced Driver Assistance Systems (ADAS) and autonomous driving features, it became clear that the detection and mitigation of faults addressed in ISO 26262 were no longer sufficient to cover all the engineering challenges related to ADAS and autonomous driving [20].

SOTIF is defined as the absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or by reasonably forseeable misuse by persons [21]. Hazardous behaviour following from intended functionality is the main focus of this standard. According to the standard, the key to ensuring safety is to obtain a proper understanding of the function by the user, its behavior and its limitations (including the human/machine interface).

III. Methods

The methods section first introduces the example distribution center TruckLab and then presents how the functional safety engineering is applied.

A. TruckLab

TruckLab is a research laboratory located at the Eindhoven University of Technology. The lab represents a scaled-down version of the Jumbo distribution center in Veghel, The Netherlands and is used as a test and research facility for autonomous driving of trucks. The system architecture, of the trucks shown in Figure 2, is presented in Figure 3. The functional safety analysis and assessment in this paper focuses on the perception system for object detection.

Object detection, localization and classification in TruckLab is based on a LIDAR sensor and a detection pipeline including preprocessing, a neural network and a decoder. For environmental perception, the concept of Occupancy Grid Map (OGM) in 2D is used. An OGM is able to represent the detected objects on a 2D map at a given time instant. Due to the relative low computational power in the truck, the detection pipeline is performed on an external control PC. Figure 4 illustrates the architecture including detection pipeline (blue blocks) for object detection in TruckLab. The preprocessing step concerns conversion to the appropriate Tensor dimensions (416×416×3) for the neural network. The neural network is based on the YOLOv3-416 architecture [22]. The detection pipeline ends with a decoder that processes the output elements of Figure 4 in order to draw bounding boxes around the detected objects and discard overlapping detections by means of non maximum suppression.

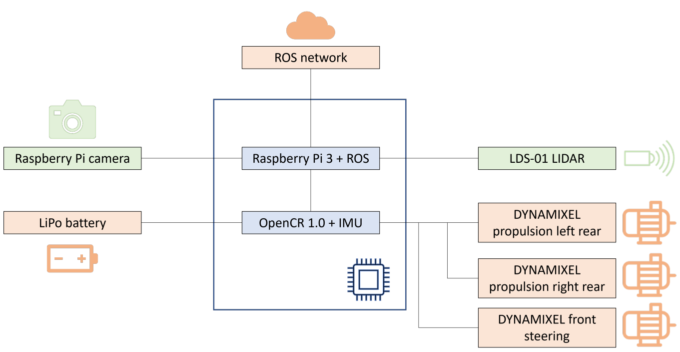


Fig. 3. TruckLab vehicle architecture with central processor and board unit in blue, perception elements in green and other auxiliaries in orange.

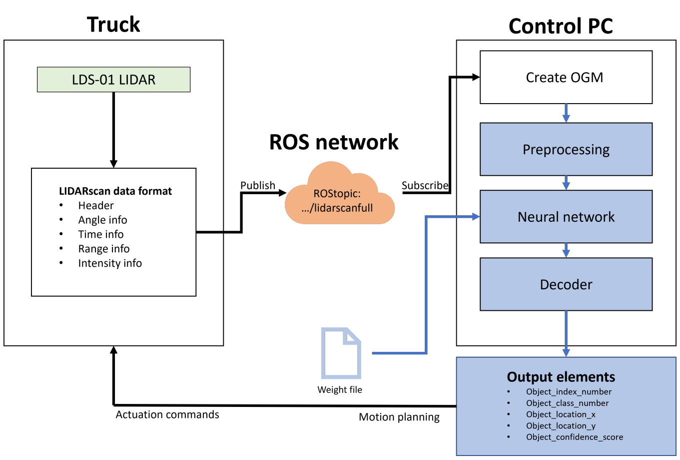


Fig. 4. TruckLab object detection architecture implementation including detection pipeline.

B. Hazard analysis and risk assessment

Hazard analysis and risk assessment is a method to identify and categorize hazardous events of items and to assess them based on Automotive Safety Integrity Levels (ASILs). Where an item is a system or an array of systems that implement a function at the vehicle level, e.g., steering, braking and accelerating.

The output of HARA is a set of safety goals with corresponding ASIL rating to provide a starting point for the

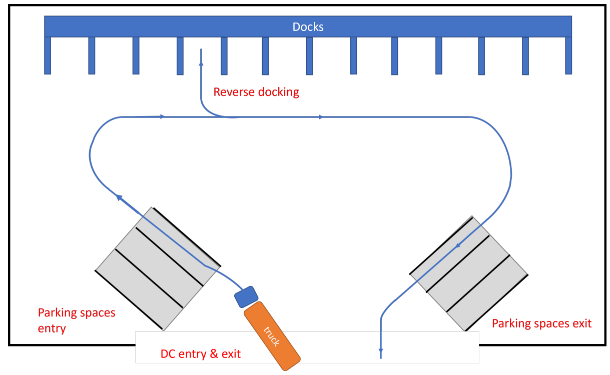


Fig. 5. TruckLab driving scenario: Enter the DC, drive to parking, park at entry, drive to dock, reverse dock, drive to parking exit, park at exit and leave DC.

Functional Safety Concept (FSC). ISO 26262-3 provides requirements and recommendations for performing HARA in an Automotive context. This research implements the following 7 steps for HARA:

1) **Item definition:** Define and describe the item, its dependencies on, and interaction with, the environment and other items.
2) **Operational situations and operating modes definition:** Define all relevant operational situations and operating modes.
3) **Hazard identification:** Systematically identify hazards of the relevant item, where a hazard can arise from an item failure or reasonable foreseeable misuse by a driver. For systematic hazard identification, ISO 26262 mentions brainstorming, checklists, Failure Mode and Effect Analysis (FMEA) and more techniques. The hazardous events shall be defined at item level for relevant combinations of operational situations and hazards. Moreover, they must include consequences, e.g., Failure of the car's lighting system can cause loss of forward and/or backward illumination.
4) **Hazardous event classification:** Classify all identified hazardous events based on Severity, i.e. the severity of any potential injury, Exposure, i.e., how frequently an operational situation arises and Controllability, i.e., whether a situation can be managed to avert injury.
5) **ASIL determination:** Determine the ASIL rating for each hazardous event. The possible levels are quality management (QM), ASIL A, B, C and D, with D the most safety critical.
6) **Safety goal determination:** Determine a safety goal for each hazardous event. In other words, this step concerns formulation of functional objectives or top-level safety requirements for the item.
7) **HARA verification and validation:** Verify and validate the HARA by means of independent assessors. This step will not be executed in this paper.

C. Functional safety assessment

The functional safety assessment in this research consists of comparing the results of the functional safety concept according to ISO 26262 and SOTIF, to the current implementation of object detection in TruckLab. Based on a safety claim for the current implementation of object detection, an estimation about the safety of the TruckLab object detection can be made. Moreover, using HARA, possible fault situations that are not covered by ISO 26262 and SOTIF can be discovered to assess the completeness of the automotive standards for functional safety.

IV. Results

This section presents the results of the functional safety engineering using the approach as described in section III

A. Hazard Analysis and Risk Assessment

To specify the central object of the functional safety concept, an item definition is presented as follows:

Item name: Perception system (for object detection).
Item description: The perception system is responsible for detecting objects within the environment of a distribution center. The perception system includes a LIDAR sensor, Raspberry Pi 3, ROS communication network, Control PC and a detection pipeline.
Item boundaries: The item is only applied within a distribution center. The item consists of all operations between environment sensing with the LIDAR to object detection.
Item interactions: The perception system interacts with a motion planning system as well as the actuation system of the truck (Figure 1). As external interactions, the perception system will interact with its environment: the distribution center.
Item assumptions:

1) The item is only operational within the area of a distribution center (DC).
2) The item has a dashboard for displaying relevant information to the driver.
3) The distribution center is a controlled environment where only trucks, forklifts, employees, animals, infrastructure elements and cargo can be considered as objects.
4) the item experiences the scenario of Figure 5 3 times per day.

Since the environment is defined to be a distribution center only, the operational situations of the item are considered to be: arriving at DC, leaving the DC, forward driving and reverse driving. Other situations like waiting at a parking space or docking, are not considered to be relevant as they cannot cause a hazardous event by themselves.
The item has multiple operating modes:

1) Item switched on: The item is in operation and performs the procedures to detect objects.
2) Object detected: The item has detected an object and communicates this through the truck dashboard.
3) Item failed: The item is in a failure state.

The definition of the operational situations and operating modes is important for identifying potential hazards. Each hazard in combination with an operational situation is presented as a hazardous event in Table I. The possible consequences are defined by propagating the hazardous events through the vehicle architecture of Figure 1. A wrong detection of an object is not harmful, but an actuation based on a wrong detection can be.
Since a major part of the hazards can happen in all defined operational situations and operating modes, the hazardous event is defined for all situations and modes simultaneously. Besides this, the item can experience a hazardous event that is situation-dependent.

Up to now, HARA has provided an overview of the potential hazards and risks regarding TruckLab's perception system for object detection. The next step concerned the assessment of the hazardous events based on [18]. The results of the assessment and safety goal definition are shown in Table II. The criteria for the safety ratings can be found in the HARA according to [18].

To assess the safety concept of the perception system of TruckLab, a safety claim is constructed. The safety claim diagram is shown in Figure 6. The claim: ”The object detection system in TruckLab prevents collisions to happen” is supported by arguments in green. In purple, actual evidence in the form of test results or documentation is provided ([23] [24] [25]).

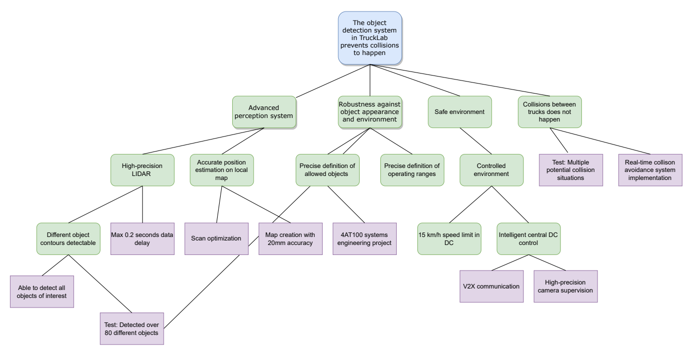


Fig. 6. Safety claim (blue), arguments (green) and evidence (purple) of TruckLab object detection system according to ISO 26262 and ISO/PAS 21448.

V. Discussion

The HARA provides a thorough understanding of the potential risks and hazards of an object detection system within a DC. Although object detection itself cannot directly harm something or somebody, the propagation of the perception towards actuation of an autonomous vehicle can. This propagation led to a rather complex functional safety engineering challenge. According to Table I, the possible consequence: ”collision with object” is a result from almost all hazardous events. In terms of severity, this consequence can be considered as a worst-case situation. This consequence must therefore be avoided at all times under all operating conditions and operating modes.
Keeping in mind the severity of potential consequences, the hazardous event assessment shows a mixture of low and high safety-critical scores. On one hand the low QM scores.

TABLE I
IDENTIFICATION OF POSSIBLE HAZARDOUS EVENTS FOR RELEVANT OPERATIONAL SITUATIONS AND OPERATING MODES

Operational situation	**Operating mode**	**Possible hazardous events**	**Possible consequences**
All	All	Object out of item's relevant field of view	Collision with object (front, rear or sideways).
All	All	Reflections from environment through light, fog, snow or heavy rainfall	Short- or long-term sensor blindness or severe visual impairments.
All	All	No/incorrect object detection due to object appearance (shape, colour, texture)	Collision with object (front, rear or sideways) or scrape damage.
All	All	(Electromagnetic) interference on item	Sudden/unexpected movement of truck.
All	All	Exceed limitations of item (computation speed, temperature, connection quality & bandwidth)	Collision with object (front, rear or sideways).
All	All	Incorrect location mapping of object onto OGM	Collision with object (front, rear or sideways) or scrape damage.
All	All	Communication failure (ROS & in-truck)	Incorrect/incomplete environment information at some instance.
All	All	No detection that item is switched off	False assumption leading to collision with object of scrape damage.
All	Object detected	Loosing detected object out of sight	Accelerate against object or infinite waiting. loop
Arriving at DC	Item switched off	Too late activation of item	Collision with object, or scrape damage.

TABLE II
HAZARDOUS EVENT ASSESSMENT AND SAFETY GOAL DEFINITION USING AUTOMOTIVE INTEGRITY SAFETY LEVELS

Hazardous event	**Exposure**	**Severity**	**Controllability**	**ASIL**	**Safety goals**
Object out of item's relevant field of view	E4	S2	C3	C	1) Avoid collision/scrape with object.
Reflections from environment through light, fog, snow or heavy rainfall	E3	S2	C3	B	1) Ensure robust perception (for assumable environmental conditions at DC). 2) Ensure that short-term issues cannot cause an accident.
No/incorrect object detection due to object appearance (shape, colour, texture)	E3	S1	C2	QM	1) Avoid collision/scrape with object. 2) Ensure robust perception for objects. 3) Specify allowed vehicle appearances.
(Electromagnetic) interference on item	E1	S2	C2	QM	1) Minimize target area for EMC. 2) Suppress interference impact.
Exceed limitations of item (computation, temperature, connection)	E2	S2	C2	QM	1) Define safe operating ranges for system. 2) Avoid exceedance situation. 3) Have enough performance and stability.
Incorrect location mapping of object onto OGM	E3	S2	C2	A	1) Avoid collision/scrape with object. 2) Minimize wrong OGM impact. 3) Maximize mapping performance.
Communication failure (ROS & in truck)	E2	S2	C2	QM	1) Detect communication failure. 2) Maximize connection performance. 3) Minimize wireless communication use.
No detection that item is switched off	E2	S2	C3	A	1) Ensure system state is known. 2) Ensure system on when in DC.
Loosing detected object out of sight	E3	S2	C3	B	1) Avoid collision/scrape with object. 2) Maximize field of view around truck. 3) Estimate object's position.
Too late activation of item	E2	S2	C3	A	1) Ensure system is on when at DC. 2) Notify truck driver about system state.

However, the scores are often only valid under certain assumptions. As example, a wrong or incorrect object detection is a low safety-critical event when sufficient (distance) margins are considered, speed is low and a high perception update frequency. Moreover, an event like communication failure, is considered to be low safety-critical since a short connection loss is very unlikely to cause a collision and connection quality is easy detectable.

On the other hand, high safety-critical scores like ASIL C & B do show severe risks. In most cases, the high ASIL rating follows from a hazardous event that is a potential cause for a collision with an object.

The safety claim in Figure 6 shows argumentation to support the claim. The use of an advanced perception system in combination with OGM provides a detailed vision of the environment at a given instance. However, during development of the perception system for object detection of TruckLab, the involvement of functional safety according to ISO 26262 and ISO/PAS 21448 is omitted. Although the performance of the object detection system seems promising, the safety level is insufficient due to absence of safe-state transition, redundancy (in perception, communication and actuation) and minimal safety-related documentation in development. Moreover, the perception is solely based on a single sensor, a 2D vision is used, which ignores all objects lower than the sensor's height and no emergency brake system is present. Therefore, the TruckLab's object detection does not achieve sufficient functional safety according to ISO 26262 as well as ISO/PAS 21448.

If the TruckLab did achieve sufficient safety according to ISO 26262 and/or ISO/PAS 21448, would that actually be 'sufficient' according to new automotive standards like UL4600?

One of UL4600's main focus is to learn from experience, something that is not covered enough in ISO 26262 and ISO/PAS 21448. The key to higher safety levels for this situation, is to know what is safe to do in all possible scenarios. Something that is more likely to be solved by a goal-based safety approach instead of a prescriptive one.

According to UL4600, the safety claim of Figure 6 is weak under various scenarios. For instance, what if an encountered object is not belonging to those 80 of the test? Or what if false assumptions in the development process lead to unconsidered scenarios?

The UL4600 standard can definitely be considered as a valuable extension to enhance safety for autonomous driving. It can even be stated that traditional functional safety methods are not sufficient for concepts like TruckLab due to their fundamental approach type and shortcomings.

VI. Conclusion

Autonomous driving entails enormous challenges. Technical difficulties, moral issues and safety concerns. The current way-to-go for functional safety is by following the ISO 26262 and/or ISO/PAS 21448 standard. The question asked in this research is whether the traditional approach is still valid for advanced automation concepts like TruckLab.

This paper explored the relevance of ISO 26262 and ISO/PAS 21448 in terms of ensuring functional safety of the object detection system of the TruckLab environment, a laboratory located at the Eindhoven University of Technology. A Hazard Analysis and Risk Assessment provided a thorough understanding of the potential hazards. The hazardous events were classified from a severe risk potential (ASIL C) rating to a low risk potential (QM) rating. The major safety goal concerned the avoidance of collisions and to control the environment to support object detection performance. Although the object detection shows promising results, the system is unsafe according to ISO 26262 and ISO/PAS 21448. There was a clear lack of functional safety engineering in the development process. Additionally, UL4600 implies shortcomings in the traditional automotive standards and is a valuable extension for the traditional automotive standards.

Acknowledgment

The author would like to thank Caspar Hanselaar, MSc. for his supervision and contributions to this research.

References

[1] Ti Research, "European road freight transport market forecasts: 2021 and 2026," 2022.

[2] Statista, "European courier, express and parcels market - statistics & facts," 2023. [Online]. Available: https://www.statista.com/topics/6888/courier-express-and-parcels-cep-market-in-europe/#topicOverview

[3] C. Katrakazas, C. Antoniou, N. S. Vazquez, I. Trochidis, and S. Arampatzis, "Big data and emerging transportation challenges: Findings from the noesis project," *MT-ITS 2019 - 6th International Conference on Models and Technologies for Intelligent Transportation Systems*, Jun 2019.

[4] E. Kim, Y. Kim, and J. Park, "The necessity of introducing autonomous trucks in logistics 4.0," *Sustainability (Switzerland)*, vol. 14, Apr 2022.

[5] National Safety Council (NSC), "Work safety: Non-roadway incidents - injury facts." [Online]. Available: https://injuryfacts.nsc.org/work/safety-topics/nonroadway-incident-involving-motorized-land-vehicle/

[6] National Safety Council (NSC), "Work safety: Pedestrian vehicle incidents - injury facts." [Online]. Available: https://injuryfacts.nsc.org/work/safety-topics/pedestrian-vehicle-incident/

[7] National Highway Traffic Safety Administration and US Department of Transportation, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," Mar 2018.

[8] C. A. Hanselaar, E. Silvas, A. Terechko, and W. P. Heemels, "Detection and mitigation of functional insufficiencies in autonomous vehicles: The safety shell," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2022-October, pp. 2021–2027, Oct 2022.

[9] M. I. Nasri, T. Bektaş, and G. Laporte, "Route and speed optimization for autonomous trucks," *Computers Operations Research*, vol. 100, pp. 89–101, Dec 2018.

[10] C. M. Chen and Y. Lu, "Shipment sizing for autonomous trucks of road freight," *International Journal of Logistics Management*, vol. 32, pp. 413–433, Nov 2020.

[11] D. Liu, B. Eksioglu, M. J. Schmid, N. Huynh, and G. Comert, "Optimizing energy savings for a fleet of commercial autonomous trucks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 7570–7586, Jul 2022.

[12] P. Koopman, U. Ferrell, F. Fratrik, and M. Wagner, "A safety standard approach for fully autonomous vehicles," *Computer Safety, Reliability, and Security*, vol. 11699 LNCS, pp. 326–332, Aug 2019.

[13] G. Velasco-Hernandez, D. J. Yeong, J. Barry, and J. Walsh, "Autonomous driving architectures, perception and data fusion: A review," *Proceedings - 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing, ICCP 2020*, pp. 315–321, Sep 2020.

[14] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, pp. 60–68, Dec 2015.

[15] S. Malik, M. A. Khan, H. El-Sayed, J. Khan, and O. Ullah, "How do autonomous vehicles decide?" *Sensors*, vol. 23, Jan 2023.

[16] H. A. Ignatious, H. E. Sayed, and M. Khan, "An overview of sensors in autonomous vehicles," *Procedia Computer Science*, vol. 198, pp. 736–741, Nov 2021.

[17] Y. Li, J. Moreau, and J. Ibanez-Guzman, "Emergent visual sensors for autonomous vehicles," 5 2022. [Online]. Available: http://arxiv.org/abs/2205.09383

[18] International Organisation for Standardization, "Road vehicles - functional safety (iso standard no. 26262:2011)," 2011.

[19] A. Ceccarelli and F. Secci, "Rgb cameras failures and their effects in autonomous driving applications," *IEEE Transactions on Dependable and Secure Computing*, Mar 2022.

[20] P. Koopman and D. Prince, "Ul4600 technical overview 10 oct 2019," Oct 2019. [Online]. Available: https://www.ecr.ai/ul4600

[21] International Organisation for Standardization, "Road vehicles — safety of the intended functionality (iso standard no. 21448:2019)," 2019.

[22] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," Apr 2018. [Online]. Available: https://arxiv.org/abs/1804.02767v1

[23] P. K. Sharma, "Utilization of on-board sensors of the new robotic platform based scaled truck," Nov 2020.

[24] A. P. Kumar, "Software design of a perception system applied to the trucklab environment," Oct 2022.

[25] A. Maheshwari, "Development of digital twin framework for trucklab environment," Oct 2021.

References

[1] URL: https://www.netsuite.com/portal/resource/articles/inventory-management/distribution-centers.shtml.

[2] URL: https://trans.info/en/the-past-present-and-future-of-technology-in-the-warehouse-166914.

[3] URL: https://www.sae.org/blog/sae-j3016-update.

[4] URL: https://www.sae.org/blog/sae-j3016-update.

[5] URL: https://group.mercedes-benz.com/innovation/case/autonomous/drive-pilot-2.html.

[6] URL: https://en.wikipedia.org/wiki/Self-driving_truck.

[7] URL: https://www.dfds.com/en/about/insights/newsletters/self-driving-trucks.

[8] URL: https://loconav.com/blog/self-driving-trucks/.

[9] URL: https://www.mdpi.com/2076-3417/10/10/3535.

[10] URL: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470684320.app1.

[11] URL: https://semiwiki.com/eda/synopsys/7480-functional-safety-is-a-driving-topic-for-iso-26262/.

[12] URL: https://www.embitel.com/blog/embedded-blog/understanding-how-iso-26262-asil-is-determined.-for-automotive-applications.

[13] URL: https://neweagle.net/blog/how-iso-26262-2018-update-affects-you/.

[14] HAL. How can triz contribute to solve the problems of green logistics? 2017. URL: https://hal.science/hal-01498633/document, doi:2017.

[15] HSE. Example risk assessment for a warehouse. doi:3.10.19.

[16] David Probert Alejandro Torres Padilla Imoh Ilevbare, Rob Phaal1. Integration of triz and roadmapping for innovation, strategy, and problem solving. 2011. URL: https://www.ifm.eng.cam.ac.uk/uploads/Research/CTM/Roadmapping/triz_dux_trt_phase1_report.pdf, doi:2011.

[17] Carl Jacobson Jan Johansson Rolf Vinter Jonny Nilsson, Josef Bergenhem. Functional safety for cooperative systems. 2013. URL: https://www.researchgate.net/publication/258289530_Functional_Safety_for_Cooperative_Systems, doi:2.10.4271/2013-01-0197.

[18] PWC. Supply chain and risk management. 2011. URL: https://www.pwc.nl/nl/assets/documents/pwc-supply-chain-and-risk-management.pdf, doi:2011.

[19] Ronald S Ross. Engineering trustworthy secure systems. 2022. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1r1.pdf, doi:10.6028/NIST.SP.800-160V1R1.

[20] SHIFA SULAIMAN and AP SUDHEER. Implementation of pure pursuit algorithm based controller for a mobile platform.

[21] Jeroen P Van Den Berg. A literature survey on planning and control of warehousing systems. *IIE transactions*, 31(8):751–762, 1999.

[22] Accellera FS WG and Proposed WG. Functional safety working group. 2013. URL: https://www.accellera.org/images/downloads/standards/functional-safety/Functional_Safety_White_Paper_051020.pdf, doi:2.10.4271/2013-01-0197.