Development of a Virtual Simulation Environment and a Digital Twin of an Autonomous Driving Truck for a Distribution Center

Ion Barosan(✉), Arash Arjmandi Basmenj, Sudhanshu G. R. Chouhan, and David Manrique

Eindhoven University of Technology, Eindhoven, The Netherlands
{i.barosan,a.arjmandi.basmenj,s.g.r.chouhan}@tue.nl,
d.a.manrique.negrin@student.tue.nl

Abstract. This paper presents the development of a Virtual Simulation Environment (VSE) and a Digital Twin (DT) of an autonomously driving truck for a distribution center. While autonomous driving on public roads still faces various technical and legal challenges, within a distribution center, which is a confined area, some of these restrictions do not apply. Therefore, distribution centers can be the first environment where the autonomous driving of trucks is possible. A distribution center is a closed environment with no, or minimal generic traffic, where the trucks have relatively low speeds, short stopping distance and layout precisely known. Dedicated sensors locate the trucks. This paper addresses the mentioned aspects of driving in the distribution centers describing the necessary steps taken for the design, implementation, and testing of a VSE for a distribution center, and a DT of an autonomously driving truck. The development of the VSE is based on the integration of a SysML modeling tool – IBM Rhapsody, MATLAB Simulink, and Unity Game Engine using a Model-Based System Engineering approach. The paper also presents the test and the validation of a driving scenario used in a distribution center, using the TruckLab setup of the Eindhoven University of Technology, The Netherlands. The VSE and the DT showed considerable potential as testing and validation tools for automotive engineers, making it possible to define driving test scenarios for different types of tractor and trailer combinations.

Keywords: Digital Twin · Autonomous driving truck · Model-Based System Engineering · Virtual Simulation Environment · Distribution center · SysML · MIL (Model-In-Loop) · DTIL (Digital-Twin-In-Loop)

1 Introduction

The automotive industry is facing a fundamental change by moving its focus from a mechanical to a software-intensive approach [1]. This fundamental change

H. Muccini et al. (Eds.): ECSA 2020, CCIS 1269, pp. 542–557, 2020.
https://doi.org/10.1007/978-3-030-59155-7_39

affects many aspects related to the way customers envisage vehicles and mobility. The constant introduction of innovations and functionality in modern automobile rely mostly on software engineering model-based competence [2]. As part of the automotive industry's innovation, autonomous driving trucks offers a significant area of technological advancement. Companies like Amazon, Daimler and many others [3–5] have started testing their autonomous truck technologies. However, the mentioned companies are majorly focused on driving on a highway, and not in the areas of the warehouses or the distribution centers.

Distribution centers are the basis of a supply network, and often one of the essential parts of a production or manufacturing operation. In distribution centers goods arrive in bulk, are stored until needed, retrieved, and then assembled into shipments [5]. Moreover, the distribution centers are equipped with the latest technology for order processing, warehouse management, transportation management.

The efficient processing of a distribution center greatly impacts the final price of the product delivered to the end-user. In general, a distribution center has three main areas: the receiving, the storage, and the shipping area. It may also have additional specialized areas. In the shipping area, each store can have dedicated dock doors. The receiving area can also be specialized based on the handling characteristics of freight being received, on whether the product is going into storage or directly to a store, or by the type of vehicle delivering the product [6].

While autonomous driving on public roads still faces various technical and legal challenges, within a distribution center, which is a confined area, some of these restrictions do not apply. It may very well be the first environment where autonomous driving trucks can be possible. In container terminals Automated Guided Vehicles are already driving for many years. As future trucks can be controlled by wire, they will be well equipped for autonomous driving.

A distribution center is a closed environment with no/minimal generic traffic, such as pedestrians or cyclists. In the distribution center, the trucks have relatively low speeds, short stopping distance and layout precisely known. To manage all these aspects dedicated sensors may be used to locate the vehicles within the distribution center, for example cameras and LIDAR. However, ensuring safety in the distribution centers is a critical aspect of both software and hardware deployed in the vehicle.

This paper presents the development of a Virtual Simulation Environment, and a Digital Twin of an autonomously driving truck used to simulate different use-case driving test scenarios in distribution centers. Moreover, the VSE and the DT can be used to simulate many types of tractor and trailer combinations, making the VSE a very flexible environment for testing their maneuverability at low speeds in small areas.

This paper is organized as follows: Section 2 gives more background and motives this work further. Section 3 reviews the related work. Section 4 covers the design of the TruckLab's virtual environment and the Digital Twin of the autonomous driving truck. Section 4.3 presents the development of the VSE. Also, this section presents the integration of different tools necessary for Model-In-Loop and Digital Twin-In-Loop. Section 4.4 covers the results of a driving test scenario in the distribution center. Finally, Sect. 5 presents the conclusions, discussions and future work.

2 Context and Motivation

The significance of Logistics to the Economy of The Netherlands is around 10% of its GDP (Gross Domestic Product per capita), making Logistics a key sector of the Netherlands economy [7]. The Autonomous Vehicles Readiness index of 2019 ranks the Netherlands as the global leader in terms of preparedness to deploy autonomous vehicles [8]. This circumstance suggests that the scope for autonomous vehicles to be deployed practically is ample.

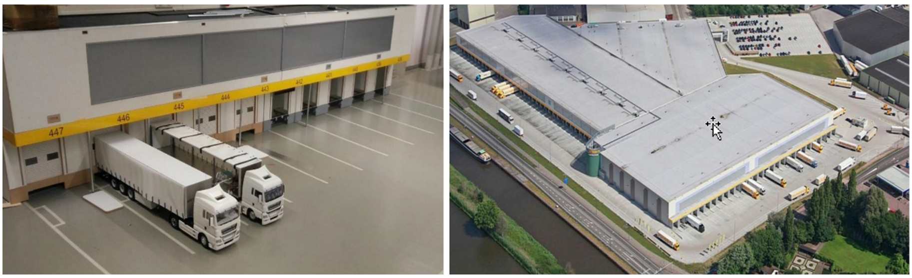

Fig. 1. Left, The TruckLab at Eindhoven University of Technology. Right, the Jumbo distribution center in Veghel, The Netherlands.

The Netherlands has the highest number of distribution centers in Europe [7]. INTRALOG project - Intelligent Truck Application in Logistics has been started for identifying the potential of automated driving within the Logistics domain [9]. INTRALOG is a consortium of companies and universities, with a focus on docking semitrailers in distribution centers.

There is a shortage of truck drivers in the Netherlands and other parts of Europe [10], which stimulates the opportunity for deploying autonomous trucks in the Logistics sector, without the concern of displacing human jobs.

In the distribution centers, the movement of trucks is done by human drivers, who could be time-dependent due to their non-availability past work hours. Thus, autonomous trucks can improve the efficiency of a distribution center and make their operation less dependent on human drivers. The most challenging driver's professional skill is the docking of a truck. An automated docking could be a significant step towards easing the manual effort required to perform the docking. With the advancement in modern sensor technology, autonomous trucks are becoming more viable for operation within short distances as that of a distribution center [11].

There have been a few attempts to develop systems that assist the driver to maneuver inside a distribution center. Eaton has developed a dock assist system as presented in [13]. With the supervision of a human driver, assistance is provided only for the case of docking. It is reported that the dock assist system made far more corrections and was slower compared to a human driver. Also, the most important limitation is that the driver needs to park the semitrailer parallel to the docking station initially.

Another major aspect of the distribution centers is safety. Besides collision between vehicles, collision avoidance is needed with objects located in the distribution centers area. With increasing attention to cooperative driving, scenarios like vehicles working together at a crossing become important. To guaranty the safety inside the distribution center, it is necessary to create the optimum path planning for the vehicles. This challenge increases the complexity of the automation in the distribution center and of the truck. When the path is predefined, it is known what kind of maneuvers the truck will execute in a specific situation. Being able to control the truck's maneuvers, we can avoid unsafe situation. We can plan the driving of the truck to the dock stations, based on the optimum path, the dock stations positions and the initial truck's parking position. In this way the safety in the distribution centers will increase.

The trucks driving between distribution centers have an increased overall length, 25 m or more, and multiple articulations. These so-called, high-capacity vehicles lead to an increase in transport efficiency both regarding costs and reduction of CO_2 emissions. The truck can be actuated fully electrically, 100% by wire, which will facilitate autonomous driving. The throttle, brake, clutch, and gearshift are already operated electrically, and the first trucks that have electric steering actuation are appearing on the market.

As a consequence, future scenarios for distribution centers can be defined, taking into account that in a distribution center:

- Safety is a crucial component for every employee.
- Articulated vehicles load and unload at a fast rate, making the distribution center a high-risk area.
- Multiple vehicles driving around each with a destination rely on the truck drives smooth maneuvers ability in a small area.
- More commonly used to increase the loads on trucks, larger vehicle combinations, which leads to an increase in complexity around distribution centers. For example, a double or triple tractor-trailer can quickly get stuck when making wrong maneuvers.
- Localization systems are needed to determine the position of the trucks and of the objects.
- Using the localization system, articulated vehicles can be monitored and position data could be made available.
- Using the vehicle's dynamics a control system for a truck can be created; a path-following controller can use the sensor information to maneuver the vehicle.

To test all the aforementioned aspects of autonomously driving trucks in a distribution center, a lot of financial and human resources are needed. It is rather difficult to build a real setup for testing and taking into account all the specified aspects. However, building a virtual testing and simulation environment for a distribution center is an affordable solution.

To design control systems that can support articulated vehicles in distribution centers, a virtual testing environment is needed. Currently, at Eindhoven University of Technology (TU/e) scaled models of articulated vehicles were developed. This is a necessary foundation for creating truck controllers before using them on large scale vehicles. These scale models are integrated into a setup called TruckLab, shown in Fig. 1 (left). The TruckLab represents a scaled model with scale 1:13.3 of the Jumbo distribution center in Veghel, The Netherlands, Fig. 1 (right).

To efficiently use the testing facilities of the TruckLab, an entire Virtual Simulation Environment was developed, which implements the supervisory control of the distribution center, the controllers of the autonomous driving trucks and the communication between the trucks and the distribution center.

To simulate the functionality of the distribution center, a Digital Twin (DT) of an autonomous driving truck was developed. Our DT is based on a five-dimensional framework approach [12], which consist of:

- ***Physical Entity* (PE)** - the mock-up truck, which contains various subsystems, sensory devices, and actuators. The sensors collect real-time states of the mock-up truck.
- ***Virtual Entity* (VE)** - the digital truck, a faithful mirror image of the physical entity containing the geometric parameters of the physical entity, such as shapes, sizes, and assembly relations. Also, the physical properties (e.g., speed, wear, and force) reflecting the physical phenomena of the entity, are part of the virtual entity. Moreover, we implemented in the VE the truck's dynamics, the numerical multi-body capabilities, and collision detection mechanism. The VE was integrated into the Unity Game Engine.
- ***The Connection* (CN)** - connects the PE and the VE. Also, the CN connects both entities to the provided services. All the connection are bidirectional.
- ***The Digital Twin Data* (DD)** - is denoted as the data from the PE, mainly including the operation states and working conditions. Also, the DD refers to the data from the VE and consists of model parameters and model operation data.
- ***The Services* (SRs)** - the services for both the PE and the VE. The SRs make the PE work as expected through real-time regulation, and sustains high fidelity of the VE with the PE through model parameters calibration. In our paper we consider only the monitoring service for the PE. For the VE, the SRs consist of construction service, calibration service, and test service for the SysML and Simulink models.

The development of the Digital Twin implied the integration of four major automotive domains: vehicle dynamics, real-time software engineering, power trains and human machine interface. To tackle the complexity of the Digital Twin, we used a Model-Based Systems Engineering (MBSE) approach based on SysML. As a MBSE methodology the SYSMOD [14] was applied. For the elicitation and understanding of the requirements, the Thinking in Time TRIZ tool [15] was used. For the analysis of the Digital Twin solutions space we used the TRIZ Contradiction Toolkit. We model the requirements, the use cases, and the architecture and the behaviour of the Digital Twin in a SysML modeling environment using the IBM Rhapsody and MATLAB Simulink. Also, base on the Unity Game Engine, a Virtual Environment of the TruckLab was created - The Virtual Truck Lab, which represents the scaled Jumbo distribution center. In addition, the Digital Twin and the Virtual Truck Lab were integrated into VSE.

3 Related Work

Several projects have been implemented in the TruckLab related to cooperative driving, collision avoidance, autonomous controlling, and localization of the trucks [17,18]. The main objective of the projects was to autonomously control the vehicle using a prerecorded input of a joystick controller [20]. The results of the tests were not optimal. The vehicle was not able to follow the course while driving backward [19]. This was because of poor road surface conditions and the limited accuracy of the localization system used. However, the path-following controllers implemented were capable of controlling an articulated vehicle both forward and backward. These path-following controllers were integrated into our VSE for testing purposes.

Hertogh had created a virtual environment and supervisory control for the TruckLab [20]. He integrated the following components in his virtual environment: a path-following controller, a VRML 3D representation of the TruckLab, and the scaled mock-up of a tractor semi-trailer combination. Hertogh made the first attempt to create a Digital Twin for a truck semi-trailer combination. However, the virtual environment and supervisory control were not flexible enough to facilitate the configuration of vehicle physics: a rigid body, collision detection, the wheels controllers, the vehicle's suspension and the axles. Moreover, the 3D virtual environment is not easily adjustable. In VRML every object needs to be defined with an orientation, position and visual components, making the creation of a large 3D scene difficult and time-consuming.

In our implementation, we use the Unity Game Engine, which is flexible and optimized for large and complex 3D scenes. Moreover, a SysML modeler is integrated into the VSE, which makes possible the development of any test driving scenario in the TruckLab or the virtual TruckLab using the Digital Twin.

The idea of creating a virtual environment for testing is not new. Coupling a 3D virtual environment with simulation tools have been used in different domains [21,22]. There is a large variety of vehicle driving simulators using virtual environments. These vehicle driving simulators are built and used to obtain insights that will be relevant to future real-world applications [23–25]. However, they are custom made and do not provide a comprehensive development platform for modeling and simulating new applications.

In recent years, autonomous driving has become an important research area. Many new tools have been developed to support autonomous driving research. The predominantly used open-source simulation engines to simulate training data in autonomous driving research are CARLA [26], TORCS [27] and AirSim [28]. CARLA is an open-source simulator developed from the ground up to support the development, training, and validation of autonomous driving systems. CARLA provides open digital assets for urban layouts, buildings, vehicles. AirSim, is released by Microsoft to support the development of autonomous vehicles like drones, cars and more. AirSim main goal is to narrow the gap between simulation and reality. TORCS, The Open Racing Car Simulator is a highly portable multi-platform car racing simulation. TORCS can be used as an ordinary car racing game, as an AI racing game, and as a research platform.

From the three mentioned simulation engines, CARLA was a good candidate for our virtual simulation environment. However, the already available integration between the Unity Game Engine and MATLAB Simulink was decisive for our design decision to use Unity Game Engine instead of CARLA.

To conclude, we created a general flexible and robust VSE for comprehensive testing of driving scenarios, by integration of three primary environments:

- IBM Rhapsody - a SysML modeling tool.
- MATLAB Simulink - a graphical programming environment for modeling, simulating and analyzing multi-domain dynamical systems.
- Unity Game Engine - a virtual reality and games development platform.

4 System Design

4.1 The Virtual Truck Lab and the Digital Twin

The TruckLab setup at the Eindhoven University of Technology is a demonstrative setup that allows testing of autonomous articulated vehicles. The TruckLab consists of two tractor-semitrailers combinations, a distribution center with a docking station, a video camera-based localization system, a computational unit and a communication system.

To address the design and test complexity of the Virtual Truck Lab, we used a SysML based Model Driven System Engineering approach. All the aspects related to requirements, structure and behavior of the Virtual Truck Lab had been modeled using the SysML diagrams. Figure 2 (right), present the overall architecture of the Virtual Truck Lab, where the *DigitalTwinTruck* component implements the Digital Twin of the truck, *EnvironmentDC* implements the 3D environment of the distribution center, *Obstacles* implements the collision detection management for the truck, and the *LocalizationSystem* component implements the localization system of the TruckLab.

To create a 3D virtual environment for the TruckLab set up a suitable 3D virtual framework development was need. The framework must support and bring

Fig. 2. Left, the 3D representation of the Truck Lab, including all the 3D assets: the buildings, the docking stations, the parking area, the obstacles inside the distribution center and the digital twin truck. Right, the high level components of the Virtual Truck Lab's architecture containing: the virtual Truck lab, the digital twin truck with its truck trailer combination, the truck manager, the collision manager.

together several core areas. We needed to add 2D and 3D graphical objects and assets in our 3D environment; assembled those assets into scenes and environments; adding lighting, audio, special effects, physics and animation, interactivity. The framework should be flexible enough to allow the development of plugins for MATLAB/Simulink and SysML modelers, by example IBM Rhapsody or Enterprise Architect. Also, because it was nearly impossible to simulate the truck's dynamics and collision detection of the truck with objects of any shapes in the Simulink, it was required to use a more general simulation environment. Taking these aspects into consideration, Unity Game Engine was selected for this purpose. Also, Unity Game Engine is using NVIDIA PhysX engine to simulate the multi-body problem in real-time, in the presence of collision, friction and joint constraints. Besides its efficient numerical multi-body simulation capabilities, Unity Game Engine can render the scene with state-of-art technologies and provide a realistic graphical output of the scene that can be used for demonstration purposes. This graphical output can also be used to feed autonomous driving algorithms in the possible future use-cases.

Using the Unity Game Engine framework a 3D replica of the TruckLab was built, as presented in Fig. 2 (left). In the 3D scene the docking station, a dummy pedestrian and the 3D representation of the Digital Twin of the truck are rendered.

The vehicles used in the TruckLab setup are tractor-semitrailers. The autonomous movement of the tractor-semitrailer in a distribution center requires making an autonomous movement from the parking station to one of the docking stations and back. The initial movement requires forward mode driving, whereas docking requires reverse mode driving as the semi-trailer must be presented to the docking station to unload the goods into the distribution center.

The Digital Twin must behave in the same way as its real counterpart vehicle, the mock-up truck. The Digital Twin must allow inputs and outputs for different parameters, which are needed to control the vehicle autonomously in the distribution center. For example, two critical parameters, the steering angle and the speed are generated by a path-following controller which uses a reference path, so that the vehicle maneuvers to the reference path and follows it. To replicate the real truck, we needed to design the Digital Twin's structure and integrate it into the Unity Game Engine. Every aspect of the actual vehicle should be modeled into the Digital Twin counterpart such that the Digital Twin can be used in different testing scenarios in the distribution centers.

Figure 3 (left), shows the top level view of the architecture of the Digital Twin's Truck component using a SysML Block Definition diagram. There are a few Truck's components which are worthwhile to mention. The *VehiclePhysicsTruck* contains the key aspects necessary for the numerical multi-body simulation capabilities of the truck. The *VehiclePhysicsTruck* is based on NWH's Physics Package [16], suitable for a wide range of vehicles including a wheel controller 3D for wheel physics used by the *WheelController* component.

The *TrailerAttachementPoint* controls the attachment of the semi-trailer to the tractor, which allows the coupling and decoupling of the semi-trailer from the tractor. The *Camera* component presents different views from the 3D scenes to the user: one from the cabin, one from the front of the truck. The *Body* component describes the main components of the truck's body. Every component of the Digital Twin has its data attributes and behavior, which are implemented in Unity Game Engine using C# scripts. Also, every component has a GUI which facilitates the interaction between the user and the component. Similar to the *DigitalTruck* component, the *DigitalTrailer* architecture is shown in Fig. 3 (right). The Digital Trailer has a body, wheels, wheels controllers, lights, an attachment point, and physics properties. The body has a suspension, doors, and a trailer interior and exterior components. Also, the *VehiclePhysicsTrailer* is implemented based on NWH's Physics Package [16].

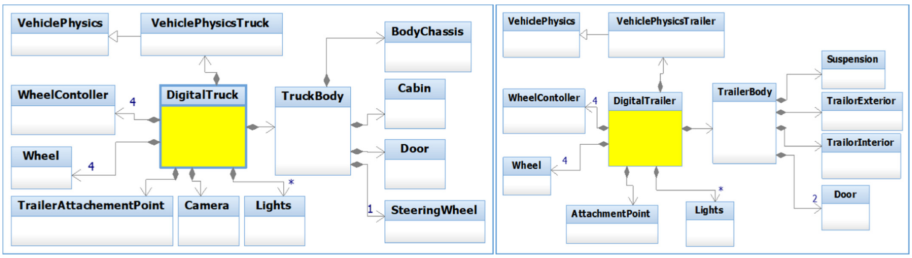

Fig. 3. Left, the high level Digital Truck's Architecture containing the wheels controllers, the trailer attachment point, the vehicle physics, the cameras, the lights and the truck body. Right, the abstract architecture of the Digital Trailer's Architecture, modeling the same abstract components as for the Digital Truck's Architecture.

4.2 Setting Up a Vehicle

After the Truck and the Trailer component have been defined, we need to set up a vehicle. In our case, the following components have to be set up for the *VehiclePhysicsTruck* and the *VehiclePhysicsTrailer*: a rigid body, collision detection, wheels controllers, vehicle's suspension and the axles. For example, the axle's functionality is implemented in a C# script that contains variables defining how much torque the axle will receive, and all the geometry related data for each of the axle and its wheels, like the steer coefficient, Ackermann percent, toe angle, caster angle, camber at top and the Anti Roll Bar Force. The amount of power an axle will receive is stored in a variable as a ratio for both the front and the rear axles. Similarly, we have variables storing the braking coefficient along with the hand brake coefficient that defines what fraction of the total brake and hand brake torque the axle will receive.

For setting up a truck, the Vehicle Physics provides the following parameters: **Sound, Steering, Effects, Engine, Transmission, Axles, Brakes, Tracks, Driving Assists, Traction control system, Damage, Trailer Handler, Ground Detection**. NWH's Physics Package [16] presents a complete description of these parameters and theirs scripts.

4.3 The Virtual Simulation Environment - VSE

After the Virtual Truck Lab and the Digital Twin of the autonomous truck were developed, a Virtual Simulation Environment was built, as presented in Fig. 4 (left). VSE integrates these two components for testing different driving use-cases in the distribution center. The intended simulations shall be able to simulate truck dynamics and collision detection of the truck with its environment, which in this case is the docking station and the distribution center area.

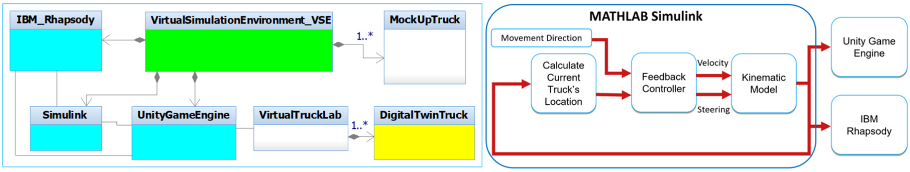

Fig. 4. Left, the architecture of the Virtual Simulation Environment. Right, the Simulink implementation of collision detection. The colors indicate: the truck's digital components (yellow), the tools we integrate (blue) and the Visual Simulation Environments (green). (Color figure online)

The VSE is made up of a SysML modeling tool - IBM Rhapsody, a simulation tool - Simulink, the Virtual Truck Lab, and the actual mock-up truck. Using the Simulation Environment different scenario-based can be implemented for Digital-Twin-in-Loop testing. The structure and behavior of the VSE is modeled in IBM

Rhapsody, including the structure and behavior of the distribution center, and the structure and behavior of the Digital Twin of the autonomously driving truck.

The path planning and path-following, the kinematic model of the tractor-semitrailer combination, and the collision detection mechanism are implemented in Simulink, as presented in Fig. 4 (right).

The communication between IBM Rhapsody and Simulink is implemented by a Functional Mock-up Interface (FMI). Rhapsody-Simulink integration does not provide interaction with Unity. Therefore, TCP/IP is used to establish a reliable connection between Rhapsody-Simulink and Unity, which is necessary for the DTIL testing.

A TCP/IP interface is provided to control the Digital Twin and the mock-up trucks. The steering and acceleration control signals are sent through this interface. The mock-up trucks use Raspberry Pi hardware controllers and different sensors and actuators. The truck's linear velocity and position of three ArUco markers, placed on top of the trucks, are sent back as feedback. It is possible to use any external tools to communicate and control the muck-up trucks in the simulated docking station. Collision information of the truck with objects in the scene are also available through a similar interface. As mentioned in Sect. 4, using the Virtual Truck Lab environment, it is possible to customize vehicle dynamics parameters such as engine characteristics, suspension system, steering, transmission. For the customization of the parameters the Vehicle Physics by NWH is used.

4.4 Testing of VSE

To test and validate the VSE, including the Virtual Truck Lab and the Digital Twin, a few use-cases scenarios were implemented. We have implemented different scenarios in which some boundary conditions were tested, including the accuracy of the truck path controller, the collision detection between crossing trucks, etc. For example, when all the docks are occupied, based on a predefined priority list, the trucks can leave to the allocated parking places.

There are many aspects that we have considered during the testing of the DT, for example:

- What is happening when the trucks cross each other path?
- Should the truck's path be computed dynamically of predefined?
- What is happening when the communication between the DC and the trucks, or V2V failed?

In all mentioned situations the DT behaved accordingly. However, during the tests, we encountered some challenges related to the path-controller. Once a path is generated using the specific algorithm, it is essential to ensure that the truck follows that designed path. To facilitate the tracking of the truck and reduce any deviations from the path, specific control strategies need to be defined. Specific control strategies need to be defined to facilitate the tracking of the truck and reduce any deviations from the path. Before the controller design, it is essential to understand the truck model. During the forward motion of the truck, its position is monitored by monitoring the location of the front axle of the tractor. In case, it is required, a correction input, a steering angle, and velocity should be given to reduce deviation from the path. This correction input leads to a stable system as the point of measurement and the point of control coincide and so the trailer perfectly follows the tractor. However, during reverse driving, since the truck's position is monitored by monitoring the rear axle of the trailer, the required correction input should be given as a steering angle, otherwise, the system is unstable. Therefore, for designing control strategies, the two motions that are forward and reverse, are treated separately. This is enabled by defining the direction of motion at each instant of time, which is done in the path panning phase.

In this paper, we presented the following use cases scenario, as example:

- After arrival at the distribution center, the driver parks the truck in a designated location.
- The controller of the distribution center calls the truck to the dock at a specific location.
- A route is determined, the truck drives to the correct dock and communicates with other trucks for optimization and collision prevention, and reverses to allow access to the rear doors.
- The cargo is unloaded from the truck.
- After unloading, the truck is directed to another dock to load goods.
- The truck drives autonomously to the parking spot again.The driver pick up the truck and drives towards the supermarket.

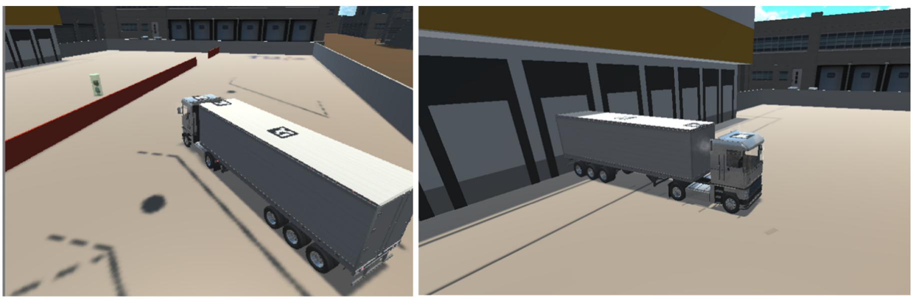

Fig. 5. Left, the initial parking position of the Truck inside the distribution center. Right, the truck parked at the designated docking station.

To test the mentioned scenario an actual simulation of tractor semi-trailer system was done. The simulation controlled the tractors velocity using a path-following controller. The throttle signal was going to be the control signal produced by the speed controller. During the simulation, the tractor semi-trailer combination started to move from its initial position until it arrived at the designated docking station.

Figure 5 (left), shows a screen-shot of the truck parked at the designated location inside the distribution center. After parking, the driver was able to monitor and to interact with the truck using a GUI. Figure 6 (left), presents an example of a GUI implemented in IBM Rhapsody, which displays information about the state of the truck. Also, using the GUI the DC operator can: allocate a docking station or a parking location for a specific track, monitor the state of the docking station, assign a parking place for the truck, halt the entire system in case of a emergency, monitor the truck driving autonomously, change the camera view of a truck and monitor the environment. Moreover, the driver can switch on/off the autonomous mode of the truck.

Fig. 6. Left, a GUI used: to monitor the truck's state, to select a docking station and monitor its state, to select a parking destination for the truck. Right, cabin camera view showing the environment from the driver's perspective.

While running, it was possible the choose between 3 cameras available in the scene. The first and default camera provided a 2D (orthographic projection) top view of the truck parking station. The second camera offered a third perspective of the tractor semi-trailer combination. The third camera provided view of the driver, which included a cockpit view of the tractor with a dashboard and driving wheel, and also the tractor's side mirror. Switching between camera's does not affect the scene simulation and only provides different views of the truck for better viewing the current simulation state. While the scene was running, it was also possible to reset the states of the scene. Figure 6 (right) shows the view from the cabin using the second camera.

The user could select between manual control of the tractor semi-trailer combination or using a network interface to control it by an external program, by example, Simulink or IBM Rhapsody. Figure 7 shows a collision test of the truck with an external object, in this case, a pedestrian. When the truck touched the bounding box of an object, it stopped. Also, the color of the cylinder around the pedestrian changes from green to red in case of a collision, as seen in the Fig. 7 (right).

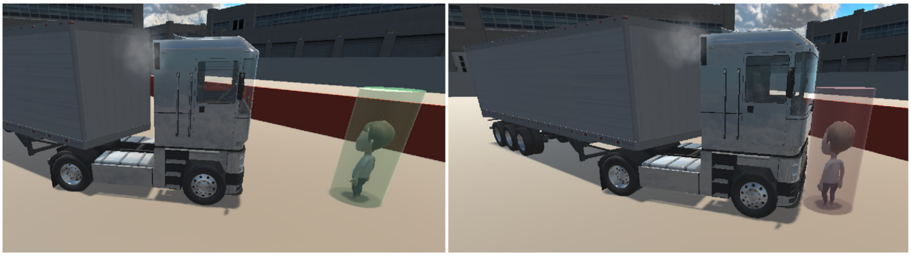

Fig. 7. The collision detection mechanism in action: left, the green cylinder indicates no collision; right, the red cylinder indicates an eminent collision with a pedestrian. (Color figure online)

After a route is determined, the truck drove to the correct dock and communicated with other trucks for optimization and collision prevention, and reversed to allow access to the rear doors. Figure 5 (right), shows the truck parked at the designated docking station. During docking time the truck was unloaded or loaded with specific goods and then drove to the designated parking location.

5 Conclusions and Future Work

Research in autonomous driving in distribution centers is hindered by infrastructure costs and logical difficulties of training and testing of the system in the physical world. This paper presented an overview of the development of a VSE for an autonomously driving truck in a distribution center. Also, the paper described the steps taken for the development of a DT for an autonomous driving truck.

The DT was designed using a Model-Based System Engineering approach based on SysML. The VSE, based on the Unity Game Engine, Simulink, and IBM Rhapsody offers considerable possibilities for testing. The pairing of the Digital Twin model with the physical world was realized at the TruckLab of the Eindhoven University of Technology. Different kinematics models for the truck and trailers were used. Also, different path-controllers were implemented and tested. The tests showed good accuracy of the truck's localization, dynamics and collision detection with objects located in the distribution center. Using the DT, many scenarios related to docking, driving, and parking in the Distributed Center's areas were tested and simulated, making the VSE a vital testing tool for engineers.

The Virtual Simulation Environment allows the integration of many trucks making possible the development and testing of applications using platooning and multi-trucks driving scenarios. Also, it is possible to define different truck and trailer combinations using the setup mechanism for vehicles, making the Virtual Simulation Environment very flexible for testing different use-case scenarios. However, a future improvement of the VSE will be to create a live-link between the SysML modeling tool and the Unity Game Engine. The live-link will allow a direct conversion of the SysML architecture and behavior of trucks and the distribution center as a 3D object into Unity Game Engine. In this way, the entire structure and the behavior of the truck will be visualized together with its 3D representation; in the same environment, in this case, the Unity Game Engine. Also, the path-controller, which ensures that the trucks follows the designed path during the reverse driving, need to be improved. Therefore, the forward and reverse motions have to be treated separately for designing control strategies. This is enabled by defining the direction of motion at each instant of time, which is done in the path panning phase. Also, the integration of CARLA simulation tool in our VSE is another area of future research.

References

1. Dajsuren, Y., van den Brand, M.G.J.: Automotive Systems and Software Engineering State of the Art and Future Trends. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-12157-0
2. Ebert, C., Favaro, J.: Automotive Software **34**, 33–39 (2017)
3. Frangoule, A.: Self-driving trucks are being tested on public roads in Virginia, September 2019. https://www.cnbc.com/2019/09/10/self-driving-trucks-are-being-tested-on-public-roads-in-virginia.html
4. Baum Hedlund Law, SAmazon Tests Self-Driving Trucks, Invests in Tech Startup, August 2019. https://medium.com/@baumhedlund/amazon-tests-self-driving-trucks-invests-in-tech-startup-a39ba986162b
5. O'Dell, J.: Daimler starts highway test of autonomous freightliner truck, September 2019. https://www.trucks.com/2019/09/09/daimler-starts-highway-test-autonomous-freightliner-truck
6. van Duin, J.R., van Kolck, A., Anand, N., Taniguchi, E., et al.: Towards an agent-based modelling approach for the evaluation of dynamic usage of urban distribution centres. Procedia-Soc. Behav. Sci. **39**, 333–348 (2012)
7. Netherlands office for Science and Special Report Technology, Smart logistics in the Netherlands, Association for Computing Machinery, pp. 5–6 (2018)
8. KPMG: Autonomous Vehicles Readiness Index 2019, pp. 5–6 (2019)
9. Gerrits, B.: Multi-agent system design for automated docking of semi-trailers by means of autonomous vehicles, University of Twente, pp. 5–6 (2016)
10. Trans. Info, November 2019. https://trans.info/en/lack-of-drivers-in-the-netherlands-every-second-company-has-trouble-finding-workers-79697
11. Rosen, R., Von Wichert, G., Lo, G., Bettenhausen, K.D.: About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine **48**(3), 567–572 (2015)
12. Tao, F., Zhang, M., Liu, Y., Nee, A.Y.C.: Digital twin driven prognostics and health management for complex equipment. CIRP Ann. Manuf. Technol. **67**(1), 169–172 (2018)
13. Berg, T.: Driver keeps hands off as eaton system docks a trailer, May 2017. https://www.truckinginfo.com/159971/driver-keeps-hands-off-as-eaton-system-docks-a-trailer
14. Weilkiens, T.: SYSMOD - The Systems Modeling Toolbox, 2nd edn. MBSE4U (2016)
15. Gadd, K., Goddard, C.: TRIZ for Engineers: Enabling Inventive Problem Solving. Wiley, Hoboken (2011)

16. NWH Vehicle Physics Manual. http://nwhcoding.com
17. Rajagopalan, S.: MATLAB based Control of a Scaled Tractor Semi-trailer, D&C 2017031, Eindhoven University of Technology (2017)
18. Hertogh, M.A.M.: Automatic docking of articulated vehicles, D&C 201703, Eindhoven University of Technology (2017)
19. Lousberg, T.A.H.: Building a MATLAB/Simulink-based truck simulator, D&C 2016012, Eindhoven University of Technology (2016)
20. Hertogh, M.A.M.: Development of a virtual environment and supervisory control for TruckLab, D&C 2018.031, Eindhoven University of Technology (2018)
21. Waas, T., Kucera, M., Földi, A.: Simulation environment for real-time applications 3D visualisation/simulation environment for testing real-time electronic control units. In: 2013 Proceedings of the 11th Workshop on Intelligent Solutions in Embedded Systems (WISES) (2013)
22. Tsai, P.-S., Wu, T.-F., Hu, N.-T., Tang, J.-H., Chen, J.-Y.: Virtual reality to implement driving simulation for combining CAN BUS and automotive sensors. In: IEEE International Conference on Information, Communication and Engineering IEEE-ICICE (2017)
23. Lindemann, P., Rigoll, G.: A diminished reality simulation for driver-car interaction with transparent cockpits. In: IEEE Virtual Reality (VR) (2017)
24. Kucera, E., Haffner, O., Leskovský, R.: Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine. In: Proceedings of the 29th International Conference Cybernetics & Informatics (K&I) (2018)
25. Yang, C.-W., Lee, T.-H., Huang, C.-L., Hsu, K.-S.: Unity 3D production and environmental perception vehicle simulation platform. In: Proceedings of the IEEE International Conference on Advanced Materials for Science and Engineering IEEE-ICAMSE (2016)
26. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: an open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning, pp. 1–16 (2017)
27. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In: Hutter, M., Siegwart, R. (eds.) Field and Service Robotics. SPAR, vol. 5, pp. 621–635. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67361-5_40
28. Wymann, B., Dimitrakakisy, C., Sumnery, A., Guionneauz, C.: Torcs: the open racing car simulator (2015)