CARLA-based digital twin via ROS for hybrid mobile robot testing ⋆

Charlotte Stubenvoll [*,**] **Tauno Tepsa** [**] **Tommi Kokko** [**]
Petri Hannula [**] **Heli Väätäjä** [**]

[*] *charlotte.stubenvoll@lapinamk.fi*
[**] *Lapland University of Applied Sciences, Rovaniemi, Finland*

Abstract: We developed the data connection via ROS (Robot Operating System) between a mobile robot and its digital twin in a CARLA-based autonomous driving simulator, which simulates realistic arctic winter weather conditions for safer, faster and less expensive autonomous vehicles testing. In our test setup we tested the hybrid case, where both robot twins were moving in the simulation and the real-world test area at the same time. Verifying our digital twin in regards to delays and applicability proved the communication via ROS to be occurring in almost real-time and the digital testing ground to profit from additional inbuilt reference points.

Keywords: Simulation, Robotics, digital twin, CARLA, ROS, multimaster_fkie, winter conditions, arctic, mobile robot, ATV

1. INTRODUCTION

Autonomously driving vehicles and robots that drive in public environments need to be safe and reliable under all weather conditions, including arctic winter conditions. Digital twins provide an opportunity to test autonomous vehicles in a safer, faster, and less expensive environment than carrying out tests in real-life conditions.

To our knowledge, the CARLA simulator for autonomous driving research is seldom, if at all, used for vehicles types not commonly found on public streets and for mobile robots other than autonomous cars. Mobile work robots, for example for deliveries, cleaning the sidewalks or for snow work, have to navigate public urban environments with pedestrians, bicycles or occasional traffic and can profit from digital twins that are capable to simulate those.

A data connection via ROS (Robot Operating System) between a mobile robot and its digital twin was developed. This allows for almost real-time exchange of commands, information, and sensor data between the twins. The digital twins of the robot and the testing ground are constructed in the WinterSIM, a CARLA-based autonomous driving simulator, which adds realistic arctic winter weather conditions to the simulation (Tepsa et al. (2021), Dosovitskiy et al. (2017)).

The digital twin design was informed by the intended future use cases: Testing, optimizing, controlling, and monitoring autonomous driving and snow cleaning functions first with the digital twin, then in hybrid approaches. We explored the applicability of our digital twin for those use cases in experiments assessing delays, verifying our digital twin setup and trying a hybrid obstacle avoidance scenario.

The backdrop for this research is the idea to use one or multiple mobile robots for autonomous snow cleaning of the public outside areas of the campus. One of the robots build from scratch for this purpose is the small mobile robot called miniATV, shown in Fig. 1 with its sensors and actuators. It has proven to be able to clean off freshly fallen snow with a snow blower attachment on the front while manually controlled.

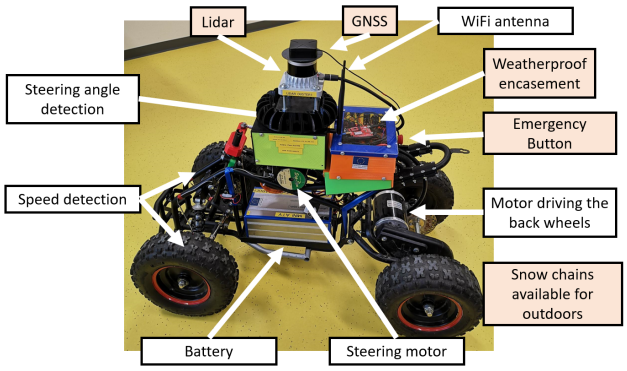

Fig. 1. The equipment, sensors and actuators of the miniATV

Automating the snow cleaning robot requires safe driving in the public outdoor spaces of the campus based on autonomous way-finding algorithms and robust obstacle avoidance algorithms, which is safer and more efficient to train within or in combination with a simulation. Simulations specifically for training autonomous snow cleaning robots have not been yet developed to our knowledge, so we used a simulation for autonomous driving for our purpose instead, which is described more in the next section on related work.

2. RELATED WORK

In the development of intelligent and automated traffic, it is particularly important to consider the impact of weather

⋆ Lapland Robotics and AI.R projects, Lapland University of Applied Sciences

conditions, especially in Finland and northern conditions. The goal of the WinterSIM project was to produce research and data on the performance of the most common vehicle sensors in winter conditions (FrostBit). In the WinterSIM project, a virtual reality (VR) simulation environment was implemented using a game engine, where weather conditions can be changed to demonstrate the impact of changing conditions on sensor data, particularly Lidar (Tepsa et al. (2021)). Generally, the produced simulation environments are utilized first, and then the driving data is transferred to the real world (Hu et al. (2023)).

The difference between the real world and its virtual counterpart is described by the concept of the "reality gap" (RG). This has been addressed in the publication by Hu et al. (2023). The study presents three different ways to reduce this gap: 1) transferring knowledge from simulation to reality, 2) learning in digital twins, and 3) learning by parallel intelligence technologies. (Hu et al. (2023)). The publication presents typical sim2real models for autonomous driving. In connection with the CARLA simulator, the KITTI vision benchmark suite is commonly used to bring features such as Lidar sensor data into the VR implementation. The KITTI benchmark suite was produced at the Karlsruhe Institute of Technology (KIT) and is freely available under a CC license. The problem of images produced by depth cameras in mixed-reality environments is particularly addressed by Argui et al. in their two publications (Argui et al. (2023) and Imane et al. (2023)). These publications present a method for combining depth camera images from the virtual world and the real world into a single view. The final conclusion of the publications is that this combination is potentially useful, but issues such as delays in combining camera data with the virtual world realistically could be problematic.

Bai et al. (2023) present an article on the Cyber mobility mirror architecture (CMM) to support Cooperative Driving Automation (CDA) research and development, and develop a CARLA-based co-simulation platform prototype. Additionally, it provides CARLA-based 3D object detection data and presents a case study demonstrating the necessity and functionality of lidar-based vehicle detection for CDA algorithm development.

The problem of accurately perceiving the environment is a very essential part of autonomous vehicles moving on public roads and in traffic systems. Cooperative perception (CP) and Vehicle-to-Everything (V2X) involve the sharing of data between autonomous vehicles and other road users, thereby extending traffic awareness beyond the sensory range of a single vehicle. This improves traffic flow and safety. Development towards this goal has been rapid in recent years, and a good overview can be found in the publication by Huang et al. (2024).

Deep and machine learning for decision-making and route planning in autonomous vehicles have benefited from developments in mapping and sensor technology. Examples of demanding decision-making include lane changing, where location data and lane markings, as well as tracking the movements of other vehicles, are utilized. Wang et al. (2023) present in their publication a deep learning method that improves the average success rate of lane changes compared to traditional planning control algorithms and reinforcement learning methods. Practical tests were conducted using the CARLA simulator. The reinforcement learning method and its decision-making architecture are addressed by Al-Sharman et al. (2023) in their publication.

The unreliability of simulators in Autonomous Driving Systems (ADS) can lead to inconsistent test results. This is addressed by Amini et al. (2024) in their publication. The publication seeks to answer two research questions: (1) How do flaky ADS simulations affect automated testing based on random algorithms? and (2) Can machine learning (ML) effectively identify flaky ADS tests while reducing the number of required test repetitions? The publication concludes that unreliability is common and poses a real problem, but the presented method can effectively identify flaky ADS tests.

To minimize the influence of the reality gap in testing of autonomous vehicles and to take their real-life physics better into account, X-in-the-Loop testing as presented by Moten et al. (2018) is now considered state of the art. Drechsler et al. (2022) propose a similar approach called Dynamic Vehicle-in-the-Loop for testing automated driving functions. They feed a automated vehicle on a test track with simulated sensor information from the same track in the CARLA simulator that includes obstacles and pedestrians crossing the street. Our future robot testing use case is inspired by those approaches and lead is to explore the suitability of our digital twin setup for such robot testing approaches in an experiment described in the next section.

3. METHODOLOGY

3.1 Software Architecture

The software architecture connects the digital and physical twins via ROS and in consequence ensuring updates from both sides fast enough for safety and to allow for deployment in use cases where both twins are used in parallel. Figure 2 shows the components of the software architecture and their interaction.

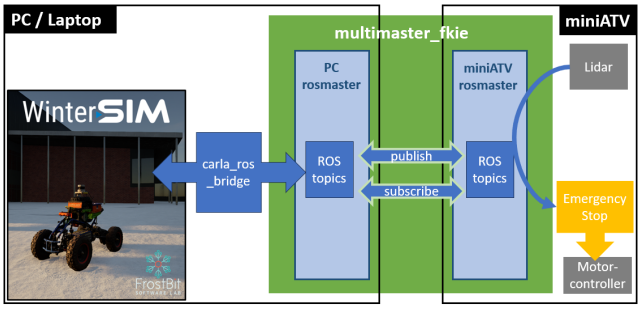

Fig. 2. Software architecture of the connection between the WinterSIM (in the PC or laptop) and the physical miniATV. The figure is explained from left to right.

Due to constraints in computation power, the simulation does not run on the robot itself, but on a separate machine, depicted in Fig. 2 as two black frames. Also any path-finding algorithm, object detection or test script is operating on the PC. This separation allows for cheaper, more compact and lightweight hardware architecture of

the robot, makes the robot easily interchangeable and the setup scalable to multiple robots.

Central to the software architecture is the *multimaster_fkie* ROS package developed by the Fraunhofer FKIE (Tiderko et al. (2016)), here depicted in green. It allows to run separate ROS masters on each machine while being able to publish and subscribe to the topics of the other machine. The more common approach to run one ROS master for both machines was discouraged by the robot's rosserial not transmitting and receiving messages on the ROS simulation time the simulation requires.

In the PC, the carla_ros_bridge (CARLA Simulator) allows the two-way communication of the CARLA-based WinterSIM simulation with ROS over ROS topics. The WinterSIM contains a model of the robot and a map of the campus area which is used as digital and physical testing ground. On top of the weather simulated in carla, it offers a simulation of realistic arctic winter conditions and its influence on the sensor output (Tepsa et al. (2021)).

The mobile robot "miniATV" runs its own ROS master to which the sensor data of the robots sensors is published. Any incoming command is first filtered through the robot's lidar based emergency stop which checks if any obstacle is closer than a threshold. If an obstacle is detected, the original command is replaced with a stop command, if no obstacle is detected too close, the steering command is waved through. This improves safety in testing of the physical robot. Over rosserial, the command is sent to a micro controller controlling the motors for driving and steering.

Since the software architecture consists of a variety of different components, it was necessary to automate the startup of those in the right order.

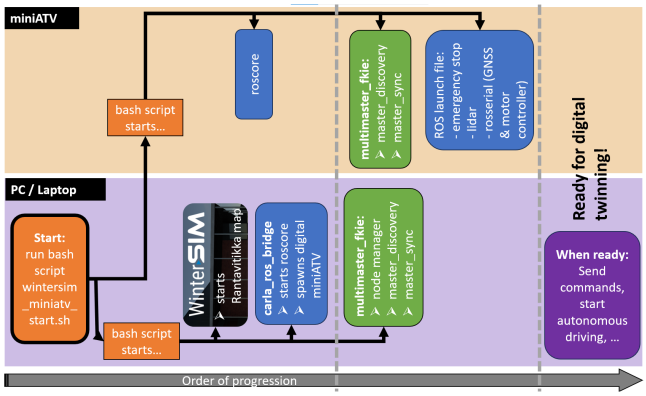

Fig. 3. The order of starting the software components for the digital twin

The components often rely on other software to be already running and cause errors if that is not the case. Especially for testing the setup it is important to have reliability and consistency in the startup process to minimize the human error. The startup script in the PC triggers the mobile robot's own startup sequence and starts first the WinterSIM and the carla_ros_bridge which also starts a roscore. Only after a ROS master is running in both the robot and the PC, the multimaster_fkie can be started on each, discover the other rosmaster in the same network and synchronize the topics between both machines. In the robot, the sensors and functionally necessary scripts are started then. After those are running and the data exchange via the multimaster_fkie is established between the physical robot and the simulation, any testing scripts or path-finding algorithms can be started.

3.2 Experiments

Delays: The usability of the setup as a digital twin for testing the digital and physical twins in parallel testing scenarios like Dynamic Vehicle-in-the Loop testing (Drechsler et al. (2022)) depends highly on the communication happening in almost real-time.

In a test, both the real and the digital mobile robot receive the same steering commands. The ROS topics are recorded and then the time of arrival of the command in the robots' steering command topic and the reaction time of the robots are tracked. The rosbag time is the point in time, when the ROS node for the recording became aware of a ROS message and recorded it. It can vary from the individual timestamp that is solidified when the ROS message is sent out. Since the rosmaster in the PC is running on ROS simulation time and the robot on non-simulation time due to the rosserial connection to the motor controlling micro controller, using the rosbag time instead simplifies the timestamp evaluation. To identify if the network connection quality has a noticeable influence on delays in the system, the connection was changed from a mobile phone hotspot to a WiFi router after half the test runs. Measurements on the network speed were taken before each half with iperf3.

A testing script was developed to give the same commands to both twins at the same points in time in each test run, see Fig. 4. First, a zero speed command is sent for 30 seconds to both the physical and the digital twin. The long initial standstill was meant to improve the GNSS positioning on the initial position, but since the GNSS suffered from electric interference from the Lidar, the GNSS data could not be evaluated. After the initial standstill, a maximum speed command was given to both twins for 10 seconds, followed again by a 30 seconds stop command. This results in over 10 seconds where the speed is greater than zero, since accelerating and decelerating are both included. The measurement of delays utilizes those points in time where the speed measurement in the ROS messages of the robot's status topic changes from or back to zero.

Fourteen test runs have been conducted, each orchestrated by the same commands from the test script. The network connection was changed after half the test runs as described above. Each test run was recorded as a rosbag recording with all ROS topics and messages. The evaluation relies on the rosbag recording timestamps to detect delays and their magnitude, since the robot and the machine running the simulation require different clocks.

Alignment A pre-existing map of the testing area on campus was used in the WinterSIM. It was created for sensor research in the area of the campus where the physical robot testing ground is located. Around the robot testing area, the buildings and details are modeled precisely because the sensor research also took place in

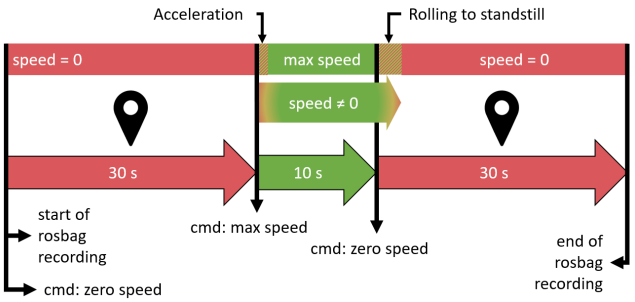

Fig. 4. Timing and course of each test run

that area. Further from this area used in testing, the map is sparse with details that were not conductive to the original purpose. The map was created based on a scan by a Leica BLK360 imaging laser scanner in winter and the surface height, which was offset due to snow covering, was later corrected with height data from the National Land Survey of Finland (NLS). Still, inconsistencies between the surface height of the real testing ground and the map were noted. The spatial relationship between buildings and details such as lampposts however is modeled realistically for the original sensor research purpose of the map. This test verifies the surface height of the digital map utilizing pre-existing GNSS reference points as ground truth in the physical space. Four exemplary points were chosen where the mobile robots would be driving later during tests. Since those reference points are not modeled in the map, triangulation to landmarks consisting of the precisely modeled details and buildings present in both the physical space and the digital map of the space was undertaken. Due to that, the number of reference points is chosen that small because the overlap between the map area detailed enough for triangulation, the area in which the robots would be roaming and the available reference points did not allow for more. The reference points are not modeled in the map of the robot testing area. To display them in the WinterSIM map, the coordinates of the reference points are transformed from the ETRS-TM35FIN/N2000 reference coordinate system to geodetic coordinates with the Paikkatietoikkuna portal maintained by the National Land Survey of Finland (NLS). The geodetic coordinates are then translated into CARLA's own coordinate system and made visible in the map of the CARLA-based WinterSIM with spawing a mark at this location. The marks consist of three axis in x, y, and z direction intersecting at the marked point for easy visibility of the reference point and precise placement of the mark at the landmarks' locations.

Any discrepancy in surface level height is measured by placing another mark on the projection of the reference point onto the surface and calculating the distance. To judge the alignment by triangulation, if the reference point coordinate is located at a similar spot in the digital map then in the real robot testing ground, the distance to the chosen landmarks in the digital map is measured with a mark similarly. Three landmarks were chosen per reference point out of prominent details and structures existing both in the digital and real map nearby the reference points. In the real testing ground, the distance between reference points and their landmarks was measured with a measuring tape. Unevenness of the surface, overgrown edges and even the grass on the lawn however added imprecision that suggests that the millimeter accuracies that scans can achieve will not be reflected in the comparison between the tape measurements and the distances in the map.

Object Avoidance The digital twin was created for robot testing and one of the intended use cases is a Robot-in-the-Loop test after the Dynamic Vehicle-in-the-Loop approach Drechsler et al. (2022) proposed. The digital twin we created would be fit for application in such a way if the real robot's reactions can be made dependent on the sensor input from the digital twin. If the sensors of the digital twin show an obstacle being too close, both the real robot and its digital twin should take action not to hit the object only present in the simulation. A prerequisite is a functioning communication between the twins in almost real-time, which was tested in the first experiment on delays above. This experiment examines the applicability of our setup for hybrid Robot-in-the-Loop testing scenarios.

Figure 5 shows the data flow for this experiment. In order not to interfere with the mobile robot's own internal lidar-based emergency stop in case of unexpected physical obstacles in the test area of the physical twin, the physical twin is not using the digital sensor data directly instead of its own sensor data. Instead, a similar lidar based emergency stop is created which receives the only the digital lidar data from the simulation and forwards the steering commands only if no digital obstacle is too close and otherwise sends a stop command. This forwarded steering command is then input to both the digital and physical robots. The physical robot then first checks in its emergency stop if no real physical obstacle is too close and only then puts this incoming command to action. The experiment on delays examines if this extra step can cause a delay.

Ten test runs have been conducted for this experiment orchestrated by a test script. Both the digital and the real robot are placed at the same location in the digital and real testing ground at the start of each test run. A digital obstacle is spawned at the same location for each test run only in the simulation in the digital robot's way. No obstacle is in the physical robot's way. The initial command that is then filtered through the robots' emergency stops according to Fig. 5 is then send out, the command is for maximum speed for ten seconds. During the execution of the command the simulated lidar data in the digital twin will then show the obstacle that soon is closer as the chosen threshold of 1.2 m. The digital twins emergency stop should then send a stop command instead of forwarding the maximum speed command to itself and the physical robot. Both twins stop propelling themselves forward and roll to a standstill since neither twin has brakes.

4. RESULTS

4.1 Experiment on Delays

Table 1 shows the measured network connection bandwidths between the command giving machine where the simulation and the test scripts were running and the real

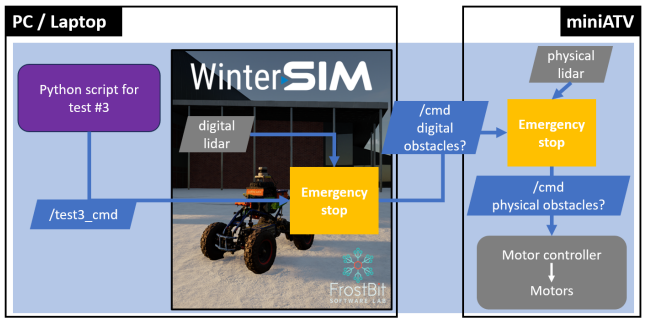

Fig. 5. How the cmd runs through the setup

Table 1. Bandwidth acquired with iperf3 for both the phone hotspot and the WiFi box

Network	From →To	Bandwidth sender [Mbits/sec]	Retry	Bandwidth receiver [Mbits/sec]
Phone hotspot	PC →miniATV	21.0	7	18.2
Phone hotspot	miniATV →PC	18.9	0	18.4
WiFi box	PC →miniATV	66.6	74	63.8
WiFi box	miniATV →PC	31.5	0	29.9

mobile robot after completing the automated startup sequence and before each half of the test runs. As expected, the WiFi has a better bandwidth compared to the phone hotspot, but an unexpected high number of retries of sending data packages. The quality of the network connection does not seem to have any noticeable influence on the delays though.

Figure 6 depicts the average delays in the command flow, both for the maximum speed command and the zero speed command combined. Only in the last step the depiction of the delays was divided since starting to drive after the first maximum speed commands and rolling to a standstill after the zero speed command are mechanically different and produced different delays.

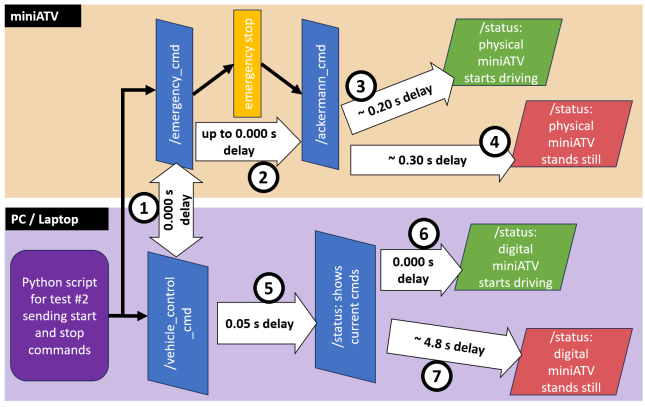

Fig. 6. Average over the delays in the command flow

The differences between the timestamps for each step in the flow of commands has been computed for each of the two commands (maximum speed and zero speed) for each of the fourteen test runs and then averaged. The only exception is delay number two (see Fig. 6). Only when sending the maximum speed command, never the zero speed command, the delay was up to 3.4 s. But the next delay in the flow, delay number three then showed a delay of over -3 s. This suggests that the delay was not in the command flow itself, but in the recording of the topic. The command on which the computation of delay two is based has been recorded in the rosbag sooner than the previous command. The delay calculation is based not on the timestamp when the ROS message originated since the simulation and the robot require vastly different clocks, but on the timestamp of when the message became available for the recording ROS node.

As a side note, no worsening of the delays has been found over the course of the fourteen test runs, which indicates that exposure of the equipment to temperatures around 0 °Celsius did not influence the delays.

4.2 Experiment on Alingment

Table 2. Difference in distances between the reference points and the landmarks between the digital and the physical twin

Reference Point	Landmark	Absolute difference [m]
1	Landmark1	0.349
	Landmark2	1.852
	Landmark3	0.151
2	Landmark1	3.195
	Landmark2	5.241
	Landmark3	1.010
3	Landmark1	0.475
	Landmark2	0.689
	Landmark3	2.132
4	Landmark1	1.533
	Landmark2	0.823
	Landmark3	0.031

The surface height of 3 out of 4 reference points is aligning within 10 cm. On the other reference point, the difference between the map surface and the coordinates of the reference point translated into a map position is 78 cm. This occurs in an especially uneven area of the testing ground.

For the x and y axis, the distances between the reference points and the landmarks varies greatly between the real testing ground and the digital map of the same area and does not reflect the millimeter or centimeter level accuracies that could be possible for a map based on a scan with this technique and equipment. Only half of the differences in distances to the 12 landmarks lie under one meter with 0.031 m as the lowest difference and only one under 10 cm. The highest difference is 5.241 m, suggesting that either the detail used as landmark was placed wrongly or a wrong detail has been taken for the landmark in the evaluation.

Experiment on Obstacle Avoidance Out of the ten test runs, the digital robot stopped propelling itself forward in all of them once the obstacle came closer than the threshold. Unfortunately, since neither the real nor the digital robot possess breaks, the digital robot did not roll to a standstill in time not to collide with the obstacle. This behavior warrants an adjustment in parameters of the digital twin model of the robot. CARLA offers a variety of parameters to adjust a simulated vehicles behavior, but they are tailored to vehicles typically found in public

traffic, not to self-build smaller mobile robots. Therefore, the adjustment is not something that could be done on the fly once the problem became apparent in the test, but needs multiple rounds of try and error.

The physical robot in the real testing ground only stopped in eight out of ten test runs. In one of those, the real robot stopped only after approximately ten seconds of driving with maximum speed, in the other the real robot did not even move from the starting position. Since the digital robot drove and reacted to the obstacle in those two test runs. One explanation for the first case would be that the physical robot lost connection and regained it after those approximately ten seconds. The test script would have ended after ten seconds as well, but it is probably a coincidence that the robot regained connection around the same time. If the real robot had not regained the connection to receive a stop command, it's own emergency stop would have prevented it from crashing into any obstacles. In the second case, when the physical robot would not start, the most likely explanation is that the startup sequence was not yet fully finished and that the test script was started by the experimenter before the respective components were ready in the physical robot. In the eight out of ten test runs in which the physical robot stopped due to the digital obstacle perceived by its digital twin, it almost immediately came to a standstill without crashing into the digital object.

5. SUMMARY AND DISCUSSIONS

The Experiment on delays finds that the communication via ROS between the digital twins is indeed in almost real-time with delays under 0.4 seconds both for command transmission and maximal speed changes, with the exception of the rolling to a standstill behaviour of the digital robot that has to be adjusted to be realistic. Therefore the setup is suitable for application in the intended robot training and testing scenarios. This experiment also uncovered a steep discrepancy between the rolling to a standstill behaviour between the digital and physical robot twins, which was discovered in the experiment on obstacles as well.

The Alignment experiment shows that the surface height of the map in the simulation is within 10 cm for three of the four reference points. The triangulation to verify the position of the reference points however did not provide clear indications if the alignment in the x and y axis is sufficient, half of the distances to landmarks differ more than one meter from the measurements in the real testing ground. Uneven ground, vegetation and differences in the level of detail likely added to inaccuracies in the measurement. Adding GNSS reference points to such maps in the simulation in future, for example by using targets measured with a GNSS total station on reference points, would not just improve the accuracies of the simulation but also make the verification process easier.

The Experiment on avoiding obstacles that exist only in the simulation suggests adjustment of the rolling parameters of the digital robot twin, so that the digital robot also rolls to a standstill almost immediately like the physical robot. Otherwise this difference in behavior could cause a worsening of the reality gap between training algorithms in the digital twin compared to the physical twin.

The physical robots behaviour during the test suggests adding a check to the startup sequence if all required components are running and responding before giving a clear for starting further scripts. Also monitoring if the connection to the command giving machine is still intact (and stopping if the connection is lost) would prevent the need to rely on the real robot's emergency stop and thus increasing safety in testing if the connection is lost.

With the described adjustments to the digital twin setup and model parameters, the digital twin is ready to be deployed in the intended robot training and testing scenarios. We will also introduce other robot into the digital twin for testing, such as an autonomous car or robots for heavier snow work. The testing ground will be equipped with 5G for a faster and more stable connection. Following the doubts about the sufficient accuracy of the map of the campus area that our tests showed, the map is currently remade from a drone scan with 1 - 2 cm accuracy and GNSS data. Additionally, a second test ground has been scanned with a Leica BLK 360 laser scanner and GNSS reference points to create a map for robots to train in an industrial environment with an overall accuracy of 5 mm. Since ROS 2 offers new features that are interesting for autonomous vehicles, we are currently moving our platforms to away from ROS 1.

ACKNOWLEDGEMENTS

This paper was made possible by the Lapland Robotics and AI.R projects of the Lapland University of Applied Sciences. Many thanks also to the creators of the WinterSIM (by the FrostBit Software Lab of the Lapland University of Applied Sciences) and especially Tuomas Herranen.

REFERENCES

Al-Sharman, M., Dempster, R., Daoud, M.A., Nasr, M., Rayside, D., and Melek, W. (2023). Self-learned autonomous driving at unsignalized intersections: A hierarchical reinforced learning approach for feasible decision-making. *IEEE Transactions on Intelligent Transportation Systems*, 24(11), 12345–12356. doi:10.1109/TITS.2023.3285440.

Amini, M.H., Naseri, S., and Nejati, S. (2024). Evaluating the impact of flaky simulators on testing autonomous driving systems. *Empirical Software Engineering*, 29(2), 47. doi:10.1007/s10664-023-10433-5.

Argui, I., Guériau, M., and Ainouz, S. (2023). A mixed-reality framework based on depth camera for safety testing of autonomous navigation systems. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2050–2055. doi:10.1109/ITSC57777.2023.10421982.

Bai, Z., Wu, G., Qi, X., Liu, Y., Oguchi, K., and Barth, M.J. (2023). Cyber mobility mirror for enabling cooperative driving automation in mixed traffic: A co-simulation platform. *IEEE Intelligent Transportation Systems Magazine*, 15(2), 251–265. doi:10.1109/mits.

2022.3203662. doi:10.1109/ MITS.2022.3203662.

CARLA Simulator, *ROS Bridge Documentation*. URL https://carla.readthedocs.io/projects/ros-bridge/en/latest/. Online. Accessed on 12.12.2023.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16.

Drechsler, M., Sharma, V., Reway, F., Sch¨

Huber, W. (2022). Dynamic vehicle-in-the-loop: A novel method for testing automated driving functions. *SAE International Journal of Connected and Automated Vehicles*, 5, 1–14. doi:10.4271/12-05-04-0029.

FrostBit, *WinterSIM*. URL https://www.frostbit.fi/en/portfolio/wintersim-2/. Accessed on 10.06.2024.

Hu, X., Li, S., Huang, T., Tang, B., Huai, R., and Chen, L. (2023). How simulation helps autonomous driving:a survey of sim2real, digital twins, and parallel intelligence.

Huang, T., Liu, J., Zhou, X., Nguyen, D.C., Azghadi, M.R., Xia, Y., Han, Q.L., and Sun, S. (2024). V2x cooperative perception for autonomous driving: Recent advances and challenges.

Imane, A., Guériau, M., and Ainouz, S. (2023). Building a vision-based mixed-reality framework for autonomous driving navigation. doi:10.1109/CoDIT58514.2023.10284251.

Moten, S., Celiberti, F., Grottoli, M., van der Heide, A., and Lemmens, Y. (2018). X-in-the-loop advanced driving simulation platform for the design,development, testing and validation of adas. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 1–6. doi:10.1109/IVS.2018.8500409.

Tepsa, T., Korhonen, M., Paananen, R., Narkilahti, A., and Herranen, T. (2021). Towards autonomous vehicle winter simulations utilising carla and real-world sensor data. In *27th ITS World Congress, Hamburg, Germany, 11-15 October 2021. Paper ID 895.*, 2330 – 2338.

Tiderko, A., Hoeller, F., and Röhling, T. (2016). *The ROS Multimaster Extension for Simplified Deployment of Multi-Robot Systems*, volume 625, 629–650. doi:10.1007/978-3-319-26054-9_24.

Wang, J., Chu, L., Zhang, Y., Mao, Y., and Guo, C. (2023). Intelligent vehicle decision-making and trajectory planning method based on deep reinforcement learning in the frenet space. *Sensors*, 23, 9819. doi:10.3390/s23249819.