

# Digital Twins for Autonomous Driving: A Comprehensive Implementation and Demonstration

Kui Wang\*, Tao Yu\*, Zongdian Li\*, Kei Sakaguchi\*, Omar Hashash<sup>†</sup>, and Walid Saad<sup>†</sup>

\*Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, Japan.

<sup>†</sup>Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA, USA.

Emails: \*{kuiw, yutao, lizd, sakaguchi}@mobile.ee.titech.ac.jp, <sup>†</sup>{omarnh, walids}@vt.edu

**Abstract**—The concept of a digital twin (DT) plays a pivotal role in the ongoing digital transformation and has achieved significant strides for various wireless applications in recent years. In particular, the field of autonomous vehicles is a domain that is ripe for exploiting the concept of DT. Nevertheless, there are many challenges that include holistic consideration and integration of hardware, software, communication methods, and collaboration of edge/cloud computing. In this paper, an end-to-end (E2E) real-world smart mobility DT is designed and implemented for the purpose of autonomous driving. The proposed system utilizes roadside units (RSUs) and edge computing to capture real-world traffic information, which is then processed in the cloud to create a DT model. This DT model is then exploited to enable route planning services for the autonomous vehicle to avoid heavy traffic. Real-world experimental results show that the system reliability can reach 99.53% while achieving a latency that is 3.36% below the 3GPP recommended value of 100 ms for autonomous driving. These results clearly validate the effectiveness of the system according to practical 3GPP standards for sensor and state map sharing (SSMS) and information sharing.

**Index Terms**—Digital twin, autonomous driving, roadside unit, route planning, implementation.

## I. INTRODUCTION

The realm of the Internet of Things (IoT) is rapidly expanding beyond the conventional frameworks, branching out to encapsulate a broader spectrum of digital connectivity [1]. This emerging paradigm facilitates intelligent data sharing among entities, thereby advancing our ability to monitor, control, and optimize the physical world. As a prominent manifestation of IoT, digital twin (DT) technology has begun to establish its foothold across different industries [2]. The potential of DTs lies in their capabilities to construct comprehensive digital representations, thereby enabling bidirectional interaction between the physical space and cyber space [3]. Henceforth, DTs empower real-time decision-making and enhance efficiency, productivity, and adaptability in a multitude of applications [4].

With the rapid advancement in autonomous driving technologies, the convergence of DTs and autonomous vehicles presents an exciting new frontier. The implementation of DTs within the autonomous driving ecosystem creates vast possibilities for improved safety, efficiency, and robustness of the system [5]. As such, leveraging real-time synchronization between the digital and physical worlds, autonomous vehicles can access global information to “see more and see further”, enhancing their awareness and understanding of the traffic environment [6]. Furthermore, a DT allows for effective testing and validation of

autonomous driving algorithms in a controllable and scalable virtual environment, accelerating the development and deployment of autonomous driving [7].

Hence, there has been a growing interest in introducing the concept of DTs in smart mobility [8]–[16]. The mainstream roles of vehicular DTs are discussed and summarized in [8]. The authors in [9] proposed a DT-enabled scheduling architecture to help multiple vehicle users fulfill their personalized requirements for path planning. Nevertheless, this prior work [9] was only conducted and evaluated based on simulation results, without realizing path planning for vehicles in a real-world setting. Recent works in [10]–[12] investigated the generation of highly accurate DT models by using various onboard and infrastructure sensors with edge computing to collect and process real-time information from the physical space. However, the decision feedback from cyber space to entities in physical space is not properly captured. In [13]–[15], the implementation of DT systems was done in order to provide situational awareness and cooperative driving services for *human* drivers, which fails to consider the *autonomy* of vehicles. To the best of our knowledge, there is a lack of the holistic integration of *DTs with autonomous vehicles*, as well as an absence of any practical end-to-end (E2E) system designs and implementation of such a system. Hence, in our previous work [16], we proposed a system architecture for the smart mobility DT and conducted a preliminary proof-of-concept (PoC) test. Our PoC provides a route planning service that allows a vehicle to avoid overcrowded traffic. However, this fundamental work in [16] only considered offline collection of sensor data and overlooked the route planning decisions and actions that were to be executed at the vehicle level.

In contrast to these prior works, the main contribution of this paper is a real-time smart mobility DT framework for autonomous driving, as well as a comprehensive real-world implementation of this system. The proposed system requires a realistic virtual representation of real-life traffic in autonomous driving scenarios as well as the establishment of an effective and stable request-feedback loop for the autonomous vehicle to obtain cloud-based decisions. Thus, from a system design perspective, the challenge lies in holistic consideration and integration of hardware, software, communication methods, and collaboration of edge/cloud computing. In addition, it is important to implement and test the whole system in real-time traffic conditions, which allows us to evaluate and validate the

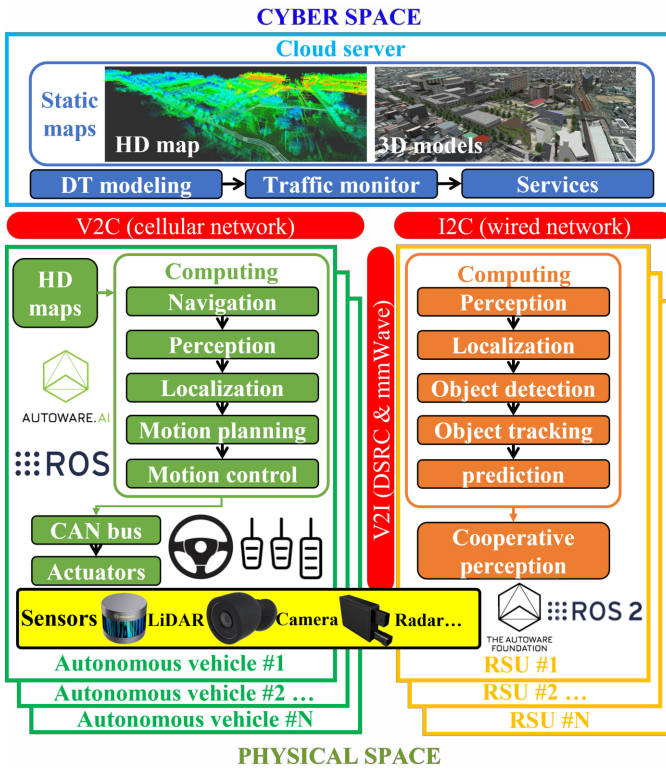


Fig. 1. System architecture of smart mobility DT.

performance of the designed DT. Hence, the system testing is carried out in different scenarios to provide a thorough analysis on the impact of various factors. In summary, our key contributions include:

- We design a smart mobility DT that reflects objects of interest in the context of road driving scenarios, e.g., roads, infrastructures, vehicles, and pedestrians.
- We implement the DT platform for autonomous driving by employing the roadside units (RSUs) and the autonomous vehicle equipped with sensors, communication modules, and edge computing capabilities.
- We design a cloud-based route planning service for the autonomous vehicle based on real-time traffic information.
- Our results show the realization of the DT modeling and demonstrate the effectiveness of the route planning service, whereby the planned routes are successfully transmitted to the vehicle for real-time execution.
- We validate the system performance in terms of reliability and latency based on 3GPP standards for sensor and state map sharing (SSMS) and information sharing.

The rest of the paper is organized as follows. Section II provides a concise overview of the system design. Details about hardware deployment, software installation, and route planning are shown in Section III. Section IV discusses the experimental evaluation. Section V concludes the paper.

## II. DESIGN FOR SMART MOBILITY DT

We first develop and propose our DT system architecture that is tailored for implementation in a specific but general-

izable environment. The proposed smart mobility DT system is essentially based on our work in [16], [17]. This proposed system enables real-time DT modeling and real-time feedback services to the autonomous vehicle.

Fig. 1 gives the system architecture, consisting of autonomous vehicles and RSUs in the physical world and a DT system maintained within the cloud server. We exploit the computing capabilities of the edge and cloud by distributing various functions and services on these two planes. The edge, referring to RSUs and autonomous vehicles, provides better real-time operation in order to address delay-sensitive tasks such as environmental perception and vehicle maneuvering. Meanwhile, the cloud offers robust data processing capabilities with higher latency, making it suitable for handling large-scale and computation-intensive tasks that need global information, such as DT modeling and deriving cloud-based services. Thus, our system design incorporates edge and cloud computing to enhance efficiency and safety for autonomous driving.

In the physical world, diverse sensors on the vehicle and RSUs capture real-time data from their surroundings. Then the vehicle can achieve a high level of autonomy, incorporating functional modules such as navigation, perception, localization, motion planning, and control. Similarly, RSUs use sensors for object detection, tracking, and prediction, upload the perception data to the cloud, and share it with nearby vehicles to enhance their safety during the driving process.

The smart mobility DT within the cloud plane is dynamically updated using real-time perception data from the physical system, thereby accurately mirroring the current state of the real-world traffic. After receiving the real-time perception data from RSU edges, the cloud server synchronizes incoming edge channels and locates detected objects based on their relative coordinates to different edge sensors. By integrating with static HD maps and 3D models, we can realize the generation of a virtual representation from the physical space to cyber space. By utilizing the smart mobility DT, we can facilitate the provision of a variety of derivative services for autonomous vehicles that include route planning, risk prediction, and congestion alerting. Since these services are predicated on the comprehensive global information from the DT, they can assist vehicles in evading high-risk and congested areas outside onboard sensing horizons and improving their overall commuting efficiency.

It is important to note that the communication within this system is based on a heterogeneous vehicle-to-everything (V2X) network [18], which includes vehicle-to-infrastructure (V2I), vehicle-to-cloud (V2C), and infrastructure-to-cloud (I2C) communications. Considering the large communication distance between vehicles and the cloud, the cellular network is applied for V2C communication. Depending on the application scenario, RSUs may send data of different levels to the cloud server and nearby vehicles, such as raw data or processed data, which have distinct requirements on communication coverage and bandwidth. Hence, we use wired networks for I2C communication and employ both dedicated short-range communications (DSRC) and millimeter wave (mmWave) technologies for V2I.

TABLE I  
COMMUNICATION REQUIREMENTS AND PERFORMANCES

Function	Requirements	Message contents	Methods	Performances
V2C	Speed: $\geq 80$ Mbps Coverage: $\geq 1$ km	Upload: vehicle position and motion states. Download: cloud-based decisions	WiMAX	$\geq 120$ Mbps $\geq 50$ km
V2I	Speed: $\geq 1$ Mbps Coverage: $\geq 50$ m	Cooperative perception (processed data, e.g., detection and tracking result)	Wi-Fi router	$\geq 10$ Mbps $\geq 200$ m
	Speed: $\geq 1$ Gbps Coverage: $\geq 20$ m	Cooperative perception (raw sensor data, e.g. LiDAR point cloud and camera images)	WiGig	$\geq 1$ Gbps $\geq 120$ m
I2C	Speed: $\geq 1$ Gbps	Environmental perception (both raw data and processed data)	Ethernet	$\geq 1$ Gbps

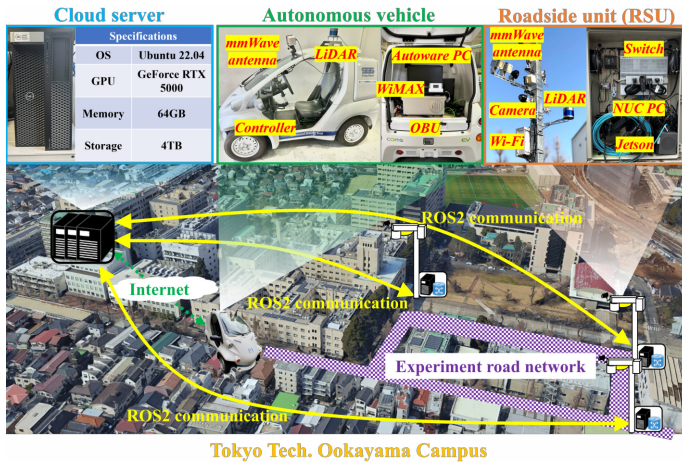


Fig. 2. Hardware components and Tokyo Tech. smart mobility field.

### III. IMPLEMENTATION FOR SMART MOBILITY DT

In this section, we will discuss the implementation of our system in the Tokyo Tech smart mobility field, as well as the route planning workflow involved.

#### A. Hardware Deployment

Our testing field in Tokyo Tech Ookayama campus, as well as our hardware devices, are shown in Fig. 2. In the field setting, we have three RSUs and one autonomous vehicle. The three RSUs are located at the corners of a square road section within the campus. Each RSU is equipped with an 80-layer LiDAR. These sensors collect raw data from the physical world, capturing information about dynamic objects in the traffic environment. Additionally, each RSU is also outfitted with edge computing devices, i.e., NVIDIA Jetson AGX Orin, where we launch functional modules for functions like object detection, tracking, and prediction, among others. The communication between RSUs and the cloud server relies on the campus Ethernet network. To cater to the requirements of different application scenarios, we upload varying levels of perception data, such as raw point cloud data, detection and tracking results to cloud.

The autonomous vehicle is equipped with a 32-layer LiDAR sensor positioned on the rooftop to sense its surroundings. A dedicated Autoware PC is utilized for processing the LiDAR data and performing environmental perception, localization, motion planning, and motion control. The control signals are then transmitted to the vehicle’s onboard unit (OBU) to enable

autonomous driving. Additionally, the vehicle is equipped with communication modules, including worldwide interoperability for microwave access (WiMAX) for V2C communication, a Wi-Fi router as a replacement for dedicated short-range communications (DSRC), and a WiGig antenna for millimeter wave (mmWave) communication. The selection of communication equipment is based on specific requirements and functions. Table I shows the required communication speeds and coverage in our experiment field, as well as the performance of selected communication methods. These devices effectively facilitate information sharing among edge and cloud planes.

The cloud server works as the central hub for data aggregation and storage, global information processing, and providing feedback services to the autonomous vehicle. Therefore, the performance requirements for the cloud server are very high, which encompass scalability, processing power, and reliability to efficiently and securely handle large volumes of real-time data. Hence, we opted for a computer equipped with a GeForce RTX 5000 GPU, ensuring ample memory and storage capacity to meet the demanding computational requirements.

#### B. Software Installation

The software in our platform is centered around two key open-source software, i.e., Autoware [19] and Robot Operating System (ROS).

In the RSU edges and cloud server, we deploy Autoware Universe and version 2 of the ROS (ROS2). Autoware Universe serves as the primary software framework for our system. It provides comprehensive functionality for the development of autonomous driving systems, making it well-suited for DT modeling. Autoware facilitates the detection and tracking of traffic participants within the RSU sensor range. The detection module applies the CenterPoint framework [20], which can detect, identify, and visualize 3D objects from the LiDAR point clouds in real-time. Then, a tracking algorithm, named Multi-object Tracker [21], is responsible for assigning the detected objects with IDs and estimating their velocities. Moreover, Autoware Universe is built on ROS2, which enables easy deployment and seamless communication among distributed computers. In our case, both the cloud server and the RSU edge computing devices are connected to the campus Ethernet network and assigned IP addresses within a local area network (LAN). Hence, we can subscribe to all the ROS2 topics on the cloud server. The utilization of ROS2 greatly facilitates the modeling of the DT and further enhances the interoperability

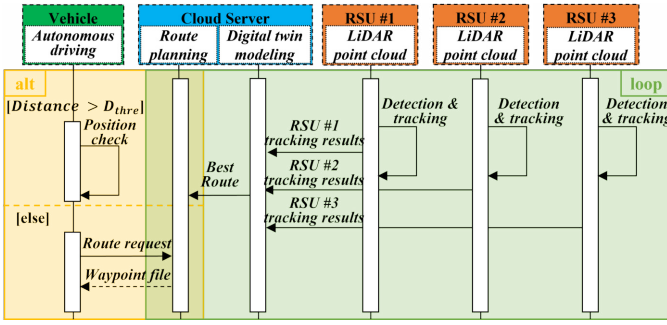


Fig. 3. Sequence diagram of route planning service.

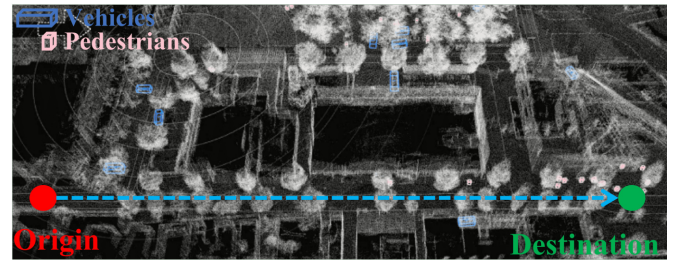
and scalability of the DT system. After acquiring object tracking data from three RSUs, the cloud server utilizes objects' positions to align them with the corresponding road segments, which facilitates traffic monitoring and the computation of congestion level on each road section.

The Autoware PC onboard the autonomous vehicle is installed with an Ubuntu 16.04 operating system, enabling the implementation of autonomous driving functionalities using Autoware AI based on ROS. Autoware AI is responsible for driving the vehicle's LiDAR sensor and utilizes the normal distributions transform (NDT) algorithm for LiDAR scan matching with the 3D point cloud map, which enables real-time localization with centimeter-level accuracy. The route from origin to destination is in the form of a prerecorded waypoint file, consisted of a set of route points with position, velocity, and orientation information. The automated vehicle can sequentially track these waypoints to follow any given route. To perform local motion planning and motion control, a velocity planner adjusts the velocity plan based on the waypoints to decelerate or accelerate in response to nearby objects and road characteristics, including stop lines and traffic lights. Then, a pure pursuit algorithm is employed to generate coordinated sets of velocities and steering angles that guide the ego vehicle to follow the target waypoints.

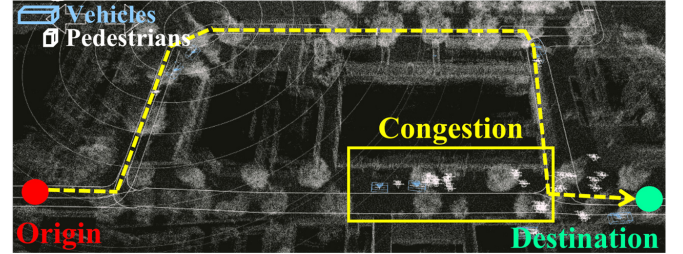
### C. Route Planning Scenario and Workflow

Based on the hardware and software implementation, we design the workflow of the route planning service, as shown in Fig. 3. There are two main sequences in the workflow, i.e., DT modeling and route requisition. The DT modeling sequence is a loop fragment for real-time edge sensing, detection, tracking, and data acquisition on the cloud plane. Three RSUs run the LiDAR driver, object detection, and tracking. The detection and tracking results are transmitted from the RSUs to the cloud. Then the cloud computing will help the ego vehicle choose the optimal path to circumvent traffic congestion and to improve its efficiency, using the shortest-paths algorithm like Dijkstra's algorithm and A\* algorithm. Since the specific route planning is not the primary focus of this paper, it is not elaborated upon in detail here.

The route requisition sequence is an alternative combination fragment, obeying the "if then else" logic. During the vehicle's movement from the starting point to the destination, it utilizes LiDAR-based localization and the road vector map to calculate



(a)



(b)

Fig. 4. Real-time DT modeling: (a) Scenario #A: no congestion in the road network, (b) Scenario #B: congestion occurs on the straight route

the distance between itself and upcoming intersections. When the vehicle approaches an intersection, within a predetermined threshold  $D_{thre}$ , it initiates a path request to the cloud server via the hypertext transfer protocol (HTTP). The HTTP server running on the cloud receives the request and sends the uniform resource locator (URL) of the optimal path to the vehicle. Then the autonomous vehicle downloads the route and tracks it for vehicle navigation.

## IV. EXPERIMENTAL DEMONSTRATION

In this section, we will show the outdoor test results and the evaluation from the aspects of functionality and reliability.

### A. Field Experiment Results

The real-time DT modeling results on the cloud plane are illustrated in Fig. 4. As discussed in Section III.A, we upload diverse types of perception data in different application scenarios. In this case, we only transmit object-tracking data to alleviate communication loads. The tracking result has a significantly lower data rate, ranging from 50 to 300 Kb/s, depending on the number of detected objects, compared to the raw point cloud data (approximately 46 Mb/s). Within the LiDAR detection range of the RSU, various entities are identified and visualized using different colored bounding boxes, indicating their categories, positions, and approximate shapes. In the figure, pink boxes represent pedestrians, while blue boxes represent vehicles. The tracking frequency can achieve a rate of 30 Hz. The cloud server performs fusion and processing of these recognized objects to choose the best global route for the vehicle. We also mark the origin and destination of the autonomous vehicle in this figure, i.e., the vehicle drives from the bottom-left road segment to the bottom-right road segment. In scenario #A, where no congestion is observed on any road segment, the cloud server selects the straight route

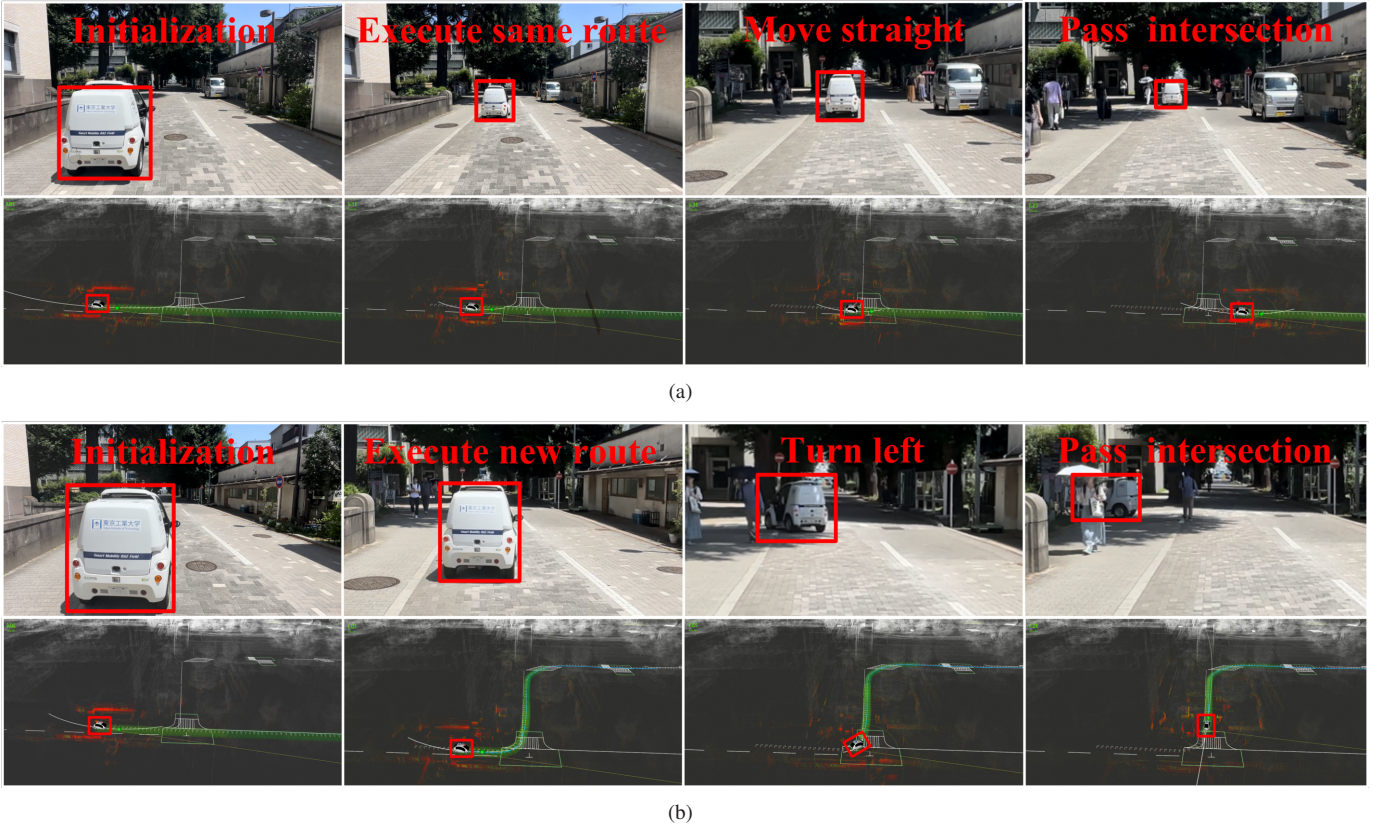


Fig. 5. Autonomous driving operation results: (a) Result #A: driving on default route, (b) Result #B: driving on alternative route

indicated by the blue dashed line. In scenario #B, congestion occurs within the straight-line segment due to the presence of pedestrians and vehicles. In such cases, the route planner selects the path represented by the yellow dashed line, which has a longer distance but effectively helps the autonomous vehicle avoid heavily congested road segments, enabling an efficient and smooth driving process.

The results of the autonomous driving operation, from real-world images and Autoware visualization tool Rviz, are shown in Fig. 5. During the initialization phase, the vehicle follows the default straight route in the road network, so in both results, the vehicle initially tracks the straight trajectory. When the ego vehicle arrives at the predetermined threshold distance  $D_{\text{thre}}$  from the intersection area, it will send a route planning request to the cloud. Result #A corresponds to scenario #A, where the vehicle downloads the same default path from the cloud and successfully traverses the intersections in a straight manner. Result #B corresponds to scenario #B, where congestion is observed on the straight-line segment. In this case, the vehicle downloads the alternative path from the cloud. The vehicle makes a left turn and proceeds along the alternative route.

In our experiment setting, the threshold distance from the ego vehicle to the intersection is determined with the consideration of safety and comfort issues. The ego vehicle should have enough time to process the planned route before reaching the intersection. Given the presence of pedestrians crossing at intersections, it is necessary for the vehicle to comfortably decelerate

and stop before reaching the intersection. In our system, the ego vehicle's free-flow speed  $v_f$  is set to 15 km/h. Braking can be classified into emergency and comfortable types. In the presence of unexpected objects, the majority of emergency braking deceleration is more than  $4.5 \text{ m/s}^2$  [22]. According to Institute of Transportation Engineers (ITE) recommendations, a comfortable deceleration rate  $a_{\text{comfy}}$  should be less than  $10 \text{ ft/s}^2$ , equivalent to  $3.048 \text{ m/s}^2$  [23]. Thus the threshold distance is determined as  $D_{\text{thre}} = 0.039 \times v_f^2 / a_{\text{comfy}} = 2.9 \text{ m}$ .

### B. Evaluation

The proposed platform is evaluated from *reliability and latency*. In DT modeling progress, congestion caused by heavy traffic flow is not a typical concern in the campus environment. Instead, congestion may arise during specific periods due to high pedestrian volumes or gatherings. In our demonstration, the frequency of changes in the path selection does not exceed 0.92 changes per minute. Hence, real-time route planning decisions exhibit low sensitivity to time variations. On the other hand, reliability, i.e., the packet delivery rate (PDR) in communication from RSUs to the cloud server, is critical as severe packet loss of tracking results could lead to incorrect planning decisions. If only the tracking results are uploaded from the RSUs, the PDR approaches 100%. When simultaneously uploading other levels of data, such as raw LiDAR point clouds, the PDR is around 99.53%, which meets the reliability requirement for SSMS, i.e., higher than 95% [24].

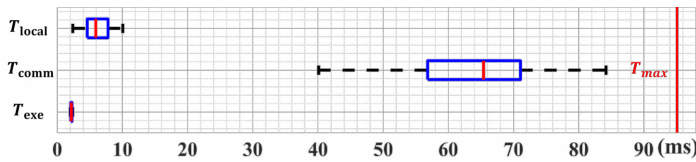


Fig. 6. Latency in route planning service.

Fig. 6 shows the latency in the route request process, including delays caused by localization and position check  $T_{local}$ , path execution  $T_{exe}$  on Autoware PC, and communication  $T_{comm}$  with cloud server. The maximum total latency  $T_{max}$  is 96.61 ms, which is 3.39% below the threshold of the max E2E latency requirements for information sharing (less than 100 ms) proposed by 3GPP [24]. Assuming the autonomous vehicle maintains the constant free-flow speed during this period, it is still approximately 2.4 m away from the intersection. This distance provides sufficient room for the vehicle to smoothly maneuver through the intersection without deceleration or, in the presence of pedestrians, to appropriately decelerate ( $3.62 \text{ m/s}^2$  on average, smaller than the emergency braking deceleration  $4.5 \text{ m/s}^2$  [22]) and stop in front of the intersection.

## V. CONCLUSION

In this paper, we have designed a smart mobility DT for autonomous driving. Our system utilizes RSUs to capture real-world traffic information, which is processed in the cloud to create a real-time DT model, enabling route planning services for the autonomous vehicle. We have implemented and demonstrated the proposed system in the Tokyo Tech smart mobility field. Test results show that the PDR of DT modeling can reach 99.53% and the latency of route planning service is smaller than 96.61 ms, which validates the effectiveness of the system in terms of reliability and latency. The latency performance currently leaves room for improvement, indicating that future work should focus on optimizing the communication system.

## ACKNOWLEDGMENT

This work was supported by the MEXT Doctoral Program for World-leading Innovative & Smart Education for Super Smart Society (WISE-SSS), JST SPRING (Grant Number JPMJSP2106), NICT-JUNO (Promotion of Commissioned Research for Advanced Communications and Broadcasting Research and Development (#22404)), and by the U.S. National Science Foundation under JUNO3 Grant CNS-2210254.

## REFERENCES

- [1] S. Nižetić, P. Šolić, D. L.-d.-I. Gonzalez-De, and L. Patrono, "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future," *Journal of Cleaner Production*, vol. 274, p. 122877, 2020.
- [2] Z. Wang, R. Gupta, K. Han, H. Wang, A. Ganlath, N. Ammar, and P. Tiwari, Prashant, "Mobility digital twin: Concept, architecture, case study, and future challenges," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17 452–17 467, 2022.
- [3] O. Hashash, C. Chaccour, W. Saad, T. Yu, K. Sakaguchi, and M. Debbah, "The seven worlds and experiences of the wireless metaverse: Challenges and opportunities," *arXiv preprint arXiv:2304.10282*, 2023.

- [4] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital twin for 5G and beyond," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.
- [5] S. Almeaided, S. Al-Rubaye, A. Tsourdos, and N. P. Avdelidis, "Digital twin analysis to promote safety and security in autonomous vehicles," *IEEE Communications Standards Magazine*, vol. 5, no. 1, pp. 40–46, 2021.
- [6] O. Hashash, C. Chaccour, and W. Saad, "Edge continual learning for dynamic digital twins over wireless networks," in *Proc. of IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, Oulu, Finland, Jul. 2022, pp. 1–5.
- [7] O. Veledar, V. Damjanovic-Behrendt, and G. Macher, "Digital twins for dependability improvement of autonomous driving," in *Proc. of European Conference on software process improvement*, Cham, Switzerland: Springer, 2019, pp. 415–426.
- [8] C. Schwarz and Z. Wang, "The role of digital twins in connected and automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 41–51, 2022.
- [9] Y. Hui, Q. Wang, N. Cheng, R. Chen, X. Xiao, and T. H. Luan, "Time or reward: Digital-twin enabled personalized vehicle path planning," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, Feb. 2021, pp. 1–6.
- [10] L. Kloeker, A. Kloeker, F. Thomsen, A. Erraji, and L. Eckstein, "How to build a highly accurate digital twin-intelligent infrastructure in the corridor for new mobility-accord," in *Proc. of 30th Aachen Colloquium Sustainable Mobility 2021*, vol. 5, Oct. 2021, pp. 651–680.
- [11] V. Tihanyi, A. and Rövid, V. Remeli, Z. Vincze, M. Csonthó, Z. Pethő, M. Szalai, B. Varga, A. Khalil, and Z. Szalay, "Towards cooperative perception services for ITS: Digital twin in the automotive edge cloud," *Energies*, vol. 14, no. 18, p. 5930, 2021.
- [12] R. Sell, E. Malayjerdi, M. Malayjerdi, and B. C. Baykara, "Safety toolkit for automated vehicle shuttle-practical implementation of digital twin," in *Proc. of International Conference on Connected Vehicle and Expo (ICCVe)*, Lakeland, FL, USA, Mar. 2022, pp. 1–6.
- [13] H. Xu, A. Berres, S. B. Yognath, H. Sorensen, P. J. Nugent, J. Severino, S.A. Tennille, A. Moore, W. Jones, and J. Sanyal, "Smart mobility in the cloud: Enabling real-time situational awareness and cyber-physical control through a digital twin for traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3145–3156, 2023.
- [14] X. Liao, Z. Wang, X. Zhao, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "Cooperative ramp merging design and field implementation: A digital twin approach based on vehicle-to-cloud communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4490–4500, 2021.
- [15] J. Dong, Q. Xu, J. Wang, C. Yang, M. Cai, C. Chen, Y. Liu, J. Wang, and K. Li, "Mixed cloud control testbed: Validating vehicle-road-cloud integration via mixed digital twin," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2723–2736, 2023.
- [16] K. Wang, T. Yu, Z. Li, and K. Sakaguchi, "Cloud and edge computing empowered mobility digital twin for automated driving: Design and proof-of-concept," in *Proc. of IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, Florence, Italy, Jun. 2023.
- [17] T. Yu, Z. Li, K. Sakaguchi, O. Hashash, W. Saad, and M. Debbah, "Internet of federated digital twins (iofdt): Connecting twins beyond borders for society 5.0," *arXiv preprint arXiv:2312.06432*, 2023.
- [18] Z. Li, K. Wang, T. Yu, and K. Sakaguchi, "Het-SDVN: SDN-based radio resource management of heterogeneous V2X for cooperative perception," *IEEE Access*, vol. 11, pp. 76 255–76 268, 2023.
- [19] Autoware universe documentation. [Online]. Available: <https://autowarefoundation.github.io/autoware-universe/main/>.
- [20] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.
- [21] Autoware multi-object tracker. [Online]. Available: [https://autowarefoundation.github.io/autoware-universe/main/perception/multi\\_object\\_tracker/](https://autowarefoundation.github.io/autoware-universe/main/perception/multi_object_tracker/).
- [22] P. Greibe, "Braking distance, friction and behaviour," *Trafitec, Scion-DTU*, 2007.
- [23] Institute of Transportation Engineers. Technical Council Committee 4A-16, "Determining vehicle change intervals: A proposed recommended practice," *Institute of Transportation Engineers*, 1985.
- [24] 3GPP, "Study on enhancement of 3GPP support for 5G V2X services (Release 16)," *Tech. Rep. 22.886 V16.2.0*, 2018.