Published in: IEEE Access - Vehicular Technology Society
DOI: 10.1109/ACCESS.2026.3702246

Date of publication 10 June 2025, date of current version 23 June 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3702246

Digital Triplet for Developing Future Reliable 6G Cooperative Intelligent Transport Systems: Case Studies from Teleoperation to Platooning

HENDRIK SCHIPPERS[1], NIKLAS A. WAGNER[1], LENNART REIHER[2], HARUN TEPER[3], LUKAS OSTENDORF[2], JIAN-JIA CHEN[4,5], LUTZ ECKSTEIN[2], and CHRISTIAN WIETFELD[1]

[1]Chair of Communication Networks (ComNets), TU Dortmund University, Otto-Hahn-Str. 6, 44227 Dortmund, NRW, Germany
[2]Institute for Automotive Engineering (ika), RWTH Aachen University, Steinbachstr. 7, 52074 Aachen, NRW, Germany
[3]Design Automation of Embedded Systems (DAES), TU Dortmund University, Otto-Hahn-Str. 16, 44227 Dortmund, NRW, Germany
[4]Lamarr Institute for Machine Learning and Artificial Intelligence, TU Dortmund University, Joseph-von-Fraunhofer-Str. 25, 44227 Dortmund, NRW, Germany
[5]Chair of Cyber-Physical Systems, RWTH Aachen University, Im Süsterfeld 9, 52072 Aachen, NRW, Germany

Corresponding author: Hendrik Schippers (e-mail: hendrik.schippers@tu-dortmund.de).

This work has received funding by the German Federal Ministry of Research, Technology and Space (BMFTR) in the course of the *research hub 6GEM* and the *transfer hub 6GEM+*, and the *GEM-X* project under grant numbers 16KISK038, 16KIS2412, 16KISS005, 16KISK036K, and 16KIS2409K.

ABSTRACT Future Cooperative Intelligent Transport Systems (C-ITSs) will impose stringent requirements on 6G mobile networks in terms of latency, data rate, and reliability. These requirements must be met in rapidly changing wireless channel conditions. Hence, their fulfillment requires rigorous and reproducible testing environments that enable a seamless transition from simulation to real-world deployment. However, existing testbeds and simulation environments either lack physical realism or fail to scale cost-effectively, hindering comprehensive C-ITS evaluation. To overcome these limitations, we implement the concept of a *Digital Triplet*, comprising three complementary layers: a full-scale real-world test field, a co-simulation environment for integrating additional simulated entities, and a scaled-down replica of the test field. This architecture enables reliable, reproducible, and scalable testing of future 6G-enabled C-ITS applications across different levels of abstraction. We demonstrate the applicability of the *Digital Triplet* through multiple case studies. For remote driving, we deploy and optimize a low-latency video streaming system and simulate a mmWave communication channel incorporating Reconfigurable Intelligent Surfaces (RISs). An up to 74% reduced median Glass-to-Glass (G2G) latency is achieved in a mobile network compared to related work using a local network. Moreover, emulating real-world vehicle localization in the scaled test field enables the implementation of cross-domain Nonlinear Model Predictive Control (NMPC)-based vehicular platooning. Finally, we evaluate the NMPC platooning controller in both decentralized (on-vehicle) and centralized (edge cloud) configurations. The results highlight the strong potential of the *Digital Triplet* to accelerate the development and validation of C-ITS applications within future 6G networks.

Video abstract: tiny.cc/digitaltriplet

INDEX TERMS 6G, Cooperative Intelligent Transport Systems (C-ITS), digital triplet, digital twin, mmWave, mobile networks, platooning, Robot Operation System 2 (ROS 2), teleoperation.

I. INTRODUCTION

The rapid evolution of Cooperative Intelligent Transport Systems (C-ITSs) toward cooperative and autonomous functionality promises significant gains in safety, efficiency, and comfort. Future 6G mobile networks are expected to deliver low-latency, high-throughput wireless connectivity to support complex coordination among vehicles, infrastructure, and digital twins. However, scalable and reliable validation of these systems under realistic, safe, and reproducible conditions remains challenging. Existing testbeds and simulation environments either lack physical realism or fail to scale cost-effectively, hindering comprehensive evaluation of C-ITSs. Consequently, a unified framework combining realism, scalability, and reproducibility is still missing.

The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kaige Qu.

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, including reprinting/republishing this material for advertising or promotional purposes, collecting new collected works for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

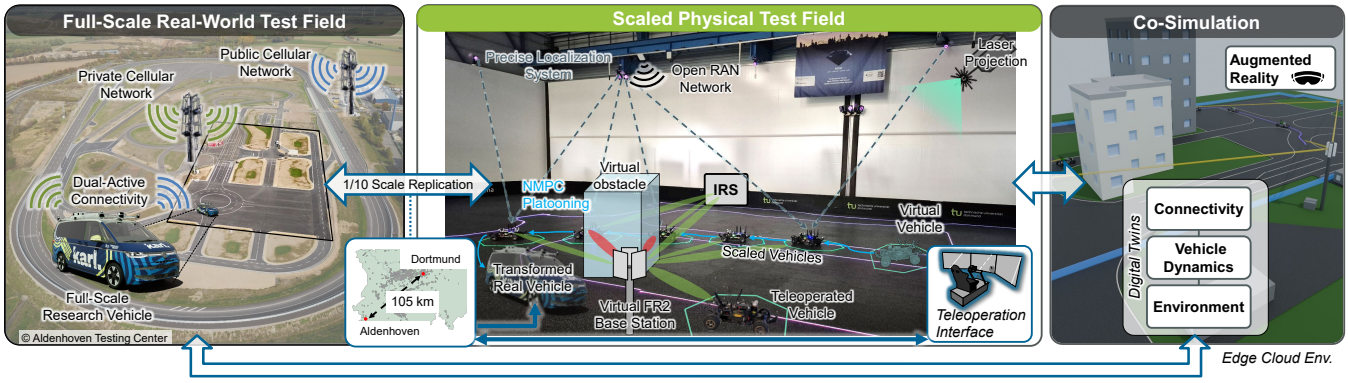


FIGURE 1. Concept of the proposed vehicular *Digital Triplet* integrating real-world, scaled, and simulated environments to improve the development of future 6G C-ITS applications.

While real-world experiments in full-scale test fields exhibit the most realistic effects, they are rather expensive, less sustainable, and may not be easily reproducible. Further, safety certification and regulatory requirements can be problematic. Simulations offer repeatability but often fall short of accurately capturing real-world phenomena. Scaled vehicular platforms can bridge the sim-to-real gap by improving accuracy, scalability, and reproducibility, while incurring lower costs than full-scale vehicles. However, they cannot fully replicate full-scale vehicle dynamics and channel characteristics. By aligning real-world, scaled, and simulated environments into the proposed *Digital Triplet*, the complementary advantages of all three domains can be combined in a single evaluation system. The objective is system-level consistency across domains, improving the reliability and efficiency of testing during C-ITS development. A real-time co-simulation layer represents and interconnects real, scaled, and simulated vehicles so they can interact, as shown in Fig. 1. Unlike traditional digital twins that mirror a single physical entity, the *Digital Triplet* integrates all domains into a synchronized testing ecosystem, enabling cross-domain validation for future 6G C-ITS solutions.

In this work, we build on the earlier vision of unifying real, scaled, and simulated vehicles [1] and its partial realization [2]. In contrast to prior work limited to simulated and scaled platforms, a full-scale vehicle is integrated. Furthermore, we interconnect a larger fleet of scaled vehicles via a private 5G Open Radio Access Network (O-RAN) testbed, comparing it to commercial public and private mobile networks. This improved connectivity for the scaled vehicles allows for synchronized interaction across all three domains.

To demonstrate the capabilities of the proposed *Digital Triplet*, we evaluate two representative 6G C-ITS use cases: Nonlinear Model Predictive Control (NMPC)-based platooning and remote driving. Platooning enables efficient autonomous traffic flows by coordinating multiple vehicles via real-time cooperative control, typically following a single leader that might be fully autonomous. However, challenging situations may still need human intervention. In this case, teleoperation serves as a critical fallback mechanism to support C-ITS. Real-time teleoperation demands low-latency, high-reliability mobile connectivity with immediate availability, making it a demanding benchmark for future 6G networks.

The key contributions are summarized as follows:

- **Real-time *Digital Triplet*** implementation for realistic, scalable testing of emerging reliable 6G use cases.
- Comprehensive closed-loop **teleoperation evaluation and optimization** with end-to-end latency decomposition and proactive bitrate adaptation, improving responsiveness and robustness.
- ***RoboRacer***: **Extension** of the Bill of Materials (BOM) and software stack to enable 5G connectivity, improved launch, and driving dynamics.
- Robust, **domain-agnostic localization**, enabling cooperative vehicular platooning.
- Cross-domain **networked NMPC platooning** in distributed and centralized real-time deployment, providing a challenging benchmark for C-ITS deployments.
- **Open-sourcing** major parts of the implementation.

The remainder of the paper is structured as follows: After discussing the related work relevant to this study in Sec. II, the approach to implementing the *Digital Triplet* architecture is explained in Sec. III. Afterward, an overview of the methodological aspects of the selected use cases is given in Sec. IV. Finally, detailed results of these are provided in Sec. V.

II. RELATED WORK

Research on C-ITS spans multiple interdisciplinary areas. Here, we will focus on discussing related work on platooning, remote driving, co-simulation and scaled vehicular testbeds.

A. CO-SIMULATION AND SCALED TESTBEDS

Numerous scaled test platforms have been developed at various scales with different focuses [3]. Several works, including [4–6], experiment with scaled vehicular platforms. While [4] employs scaled semi-trailer trucks in a real testbed, their approach does not extend to full-scale trucks or precise digital twins. In [5] and earlier studies, teleoperation is assessed in both full-scale and scaled vehicles and in simulations; however, these domains do not interact in real time. The authors of [6] utilize a scaled testbed for various autonomous

TABLE I. Required and measured remote driving latencies of related work: This work reduces latencies below 100 ms and evaluates subsystem latencies (cf. Fig. 2).

	Work	v_{max} [km/h]	Latency Components [ms] τ_1	τ_2	τ_3	τ_4	τ_5	E2E [ms]
Specified Requirements	[7, 8]	50	——100——			— 20 —		120
	[9]	250	✗	5	✗	5	✗	$10+\tau_{1,3,5}$
	[8, 10]	8	——120——			80	✗	$200+\tau_5$
	[11, 12]	15	55*	40	55*	✗	✗	$150+\tau_{4,5}$
	[13]	80	——————200——————					200
Research Systems	**Measurements and Deployed Systems**							
	[5]	✗	——160——			15	✗	$175+\tau_5$
	[14]	✗	26	48	46	14	60	190
	[15]	✗	——— 90 ———			23*	✗	$113+\tau_5$
	[16]	✗	——202——			23*	✗	$205+\tau_5$
	[17, 18]	✗	✗	38-64	✗	23-43	✗	$84+\tau_{1,5}$
	[8, 19]	60	✗	25-50*	✗	25-50*	✗	$75+\tau_{1,3,5}$
	Simulations							
	[20, 21]	✗	✗	150*	66	150*	✗	$366+\tau_{1,5}$
	[22]	✗	——— 640 ———			— 560 —		1,200
	[23]	✗	30*	60*	30*	60*	✗	180+
Commer. Grade	[24]	✗	—— 200 ——			✗	✗	$200+\tau_{4,5}$
	[8, 25]	50	50*	45	50*	15	5	165
	[26]	✗	——35-65——			✗	✗	$35\text{-}65+\tau_{4,5}$
This Work	Scaled	n.a.	23	25	16	9	n.a.	72
	Full-Scale	30	20	34	16	17	✗	$87+\tau_5$

*Assumed even distribution into categories — n.a.: not applicable

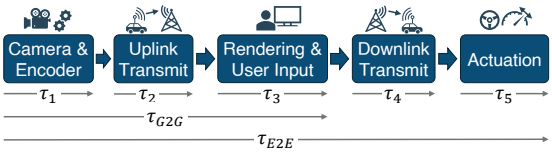


FIGURE 2. Latency components of remote driving.

functions but highlight the limitation posed by the sim-to-real gap between the scaled setup and full-scale vehicles.

Similarly, [3] identifies several challenges in scaling from small to full-scale deployments. These include a reliance on precise global positioning, incompatible with outdoor environments, unrealistically ideal indoor conditions regarding lighting, environmental factors, and weather, as well as difficulties in accurately replicating real-vehicle dynamics and sensor configurations in smaller testbeds. Existing testbeds either focus on scaled prototypes, real-vehicle setups, or simulation environments, but lack synchronized integration between physical systems and co-simulation environments.

B. ROBUST REMOTE DRIVING OF VEHICLES

Remote driving is a challenging application for mobile networks due to the high uplink data rate and strict End-to-End (E2E) latency requirements. To enable safe remote driving, both Key Performance Indicators (KPIs) need to be complied with reliably and without any gaps across a large area. Although achieving full system reliability may be infeasible [27], clear performance thresholds must be defined. Within related work, exact minimum required service levels are debated and vary largely with the exact scenario.

In general, delay and reduced video bitrates negatively impact the ability to control a vehicle [23, 28]. The impact, however, is nonlinear, and the performance of human drivers starts to deteriorate more drastically over a particular threshold. Drive tests under different latencies in [21] show a sharp degradation in driving performance once the total E2E latency exceeds roughly 360 ms. Experimental evaluations in [16] also conclude that remote driving at low speeds is feasible with E2E latencies of around 200 ms.

The 5G Automotive Association (5GAA) demands latency and data rate requirements depending on the exact teleoperation use case [7, 8], as shown in Table I. These requirements are contextualized in the demands posed by several publications of research projects on remote driving [10, 11]. All these demanded E2E latencies are in the range from 100 ms to 200 ms for velocities up to 50 km/h. As stated by the 5GAA, the remote driving capability is affected by the total latency only. Therefore, there is no need to specify individual uplink and downlink limits. This is also taken into account by German legislation [13], which stipulates a total latency of below 200 ms. Otherwise, velocity reductions must be applied immediately. In contrast, the 3rd Generation Partnership Project (3GPP) demands a network layer Round-Trip Time (RTT) below 10 ms for high-speed remote driving [9].

Overall, the latency analysis across related works proves difficult, partly due to different measurement methods and latency definitions. Thus, the overall remote driving delay of related work is compared by dividing it into five categories (cf. Fig. 2). These consist of the image capture by the camera and the subsequent video encoding delay τ_1, the uplink transmission delay τ_2, the display and user-input delay τ_3 at the operator, the downlink transmission delay τ_4, and the actuation delay τ_5 within the vehicle. Despite methodological differences, it is clear that some remote driving systems listed in Table I exceed the specified latency thresholds.

The authors of [5, 14] and [15] measure the performance of their remote driving deployments in public 4G networks. In [16, 18], 5G Non-Standalone (NSA) networks are utilized, with minor improvement in E2E delay. In [17], the uplink and downlink latency of the proposed system is evaluated in 5G Standalone (SA) networks. While other delay factors are not given, the average uplink delay τ_2 is significantly reduced, demonstrating lower 5G uplink latencies. Whereas many works focused on network and video streaming delay, [14] also analyzed the actuator delay τ_5, showing potential to optimize the E2E latency. Many works summarize τ_1 to τ_3 as the Glass-to-Glass (G2G) delay. While [5] and [14] focus on the application perspective, this paper will also focus on wireless communication and video processing. Consequently, this work yields comparatively low E2E latency values for both scaled and full-scale vehicles, as illustrated in Table I.

The authors of [19] use a proprietary protocol based on 802.11p to employ a steering controller to directly control a vehicle and achieve even lower latencies, but are restricted to a local deployment. Other works performed simulations of

teleoperation deployments to study the effects of latency in a controlled way [2, 20–22].

In addition to low latency processing steps, it is crucial to consistently meet latency requirements all the time, e.g., at challenging connectivity. The necessary steps can be divided into network-based adaptations, video coding optimizations and Artificial Intelligence (AI)-based prediction and compensation methods. In [29, 30], an analysis of suitable 5G network architectures for remote driving has yielded the need for video bitrate reductions, especially in deployments at scale. Numerous video encoding approaches [31] are available to address these constraints. As a result, various strategies exist to enhance quality and minimize bitrate while maintaining low latency in teleoperation applications. For instance, region-of-interest-based encoding is employed in [32] and [33] to reduce the video bitrate while preserving essential details, applicable on a per-stream basis [34]. Additionally, non-traditional encoding methods can reduce bitrate and latency by transmitting only essential information [35]. Predictive methods may be utilized to forecast future video data in case of erroneous transmissions [36] or high latencies [22].

While these strategies improve resilience under variable connectivity, they remain confined to individual subsystems. A holistic approach that models interactions between communication, computation, and control is often missing.

C. VEHICULAR PLATOONING FOR BENCHMARKING C-ITS

Vehicular platooning has been extensively studied as a key C-ITS use case to enhance road capacity, safety, and fuel efficiency [37]. By driving with short inter-vehicle gaps, platooning can significantly reduce aerodynamic drag. On the communications side, the requirements for reliable platooning regarding latency and positioning accuracy are especially high as specified by the 3GPP standardization [9]. Platooning therefore represents a particularly demanding cooperative driving application, making it a suitable use case for evaluating advanced development and testing frameworks.

From a control-theoretic perspective, platooning strategies can be categorized into reactive controllers such as Cooperative Adaptive Cruise Control (CACC), predictive optimization-based approaches such as Model Predictive Control (MPC), and data-driven methods including reinforcement learning. Reactive controllers regulate spacing and velocity errors based on instantaneous data of predecessor vehicles. While such approaches are computationally efficient, they lack predictive capabilities and cannot explicitly account for system constraints. In contrast, MPC methods optimize vehicle behavior over a prediction horizon, while considering vehicle dynamics and safety constraints. Data-driven approaches can learn complex behaviors from data but often lack explicit safety guarantees and require extensive training.

Early works, including [38], which was carried out in a real-world deployment, laid the foundation for platooning control strategies using Vehicle-to-Vehicle (V2V) communications. These relied on reactive approaches such as CACC and regulated the longitudinal spacing between vehicles. Subsequent research has focused on hardening control algorithms to maintain platoon stability under varying conditions, including diverse communication impairments and environmental factors [39–41]. While these studies focus on co-design of control and communication to ensure string stability, they primarily consider longitudinal control of platoons in simulation environments. MPC approaches have been proposed to enhance platoon stability and guarantee safety constraints, even in interaction with human drivers [42]. Further improvements in the reliability and latency of 5G and prospective 6G networks enable centralized control paradigms leveraging edge computing to coordinate large-scale vehicular platoons [43].

While most of the previously presented works focus solely on longitudinal control of platoons, our work also considers lateral control. Works on lateral and longitudinal control address 2D coordinated platooning maneuvers [38, 44–46]. A significant challenge in combined longitudinal and lateral control is preventing corner cutting, where vehicles following a curved path may deviate inward to minimize travel distance or lateral acceleration. This becomes particularly critical in platooning scenarios where multiple vehicles must maintain formation while navigating curves. Specifically, [45] and [44] present combined lateral and longitudinal control approaches that do not rely on leader-trajectory information, thereby simplifying control design. As stated by the authors of [44], the method is limited to constant-curvature paths, restricting its applicability to less complex driving scenarios. Progressing from classical control methods, [47] employs a reinforcement learning-based approach to optimize both lateral and longitudinal control. This work proposes a combined lateral and longitudinal control strategy, utilizing leader trajectory information to enhance platoon stability and prevent corner cutting. The employed NMPC controller provides the necessary planning capability to handle complex maneuvers, serving as a representative predictive control approach within this challenging application.

Many recent works rely on simulations [42–44, 46] or scaled testbeds [45, 47] without real-time integration with full-scale vehicles. Works solely relying on full-scale deployed systems, such as [38], however, require extensive efforts for testing. Here, the *Digital Triplet* enables thorough evaluation and fine-tuning of combined control strategies across different scales, reducing cost while providing real-time feedback from simulation and physical testbeds.

III. THE DIGITAL TRIPLET APPROACH

The proposed *Digital Triplet* framework establishes a unified, synchronized environment connecting real R, scaled S, and virtual V vehicular domains for developing and evaluating C-ITS. This approach combines the complementary advantages of each domain into one cohesive testing ecosystem.

A. ARCHITECTURE OF THE DIGITAL TRIPLET

The *Digital Triplet* architecture is organized into interconnected subsystems that form a combined end-to-end closed-loop testing environment. Each layer contains its own vehicu-

lar platforms, wireless networking modules, and control entities, all linked through a co-simulation interface. The central coordination layer synchronizes data and control commands among all domains $d \in \mathcal{D} = \{R, S, V\}$, ensuring that the same scenario can be executed simultaneously across physical and virtual entities. This synchronization allows vehicles and infrastructure components to interact under identical environmental and network conditions, which is critical for verifying the transferability of algorithms from simulation to real-world deployments. Each domain is a dynamic system $\Sigma_d = (X_d, U_d, Y_d, f_d, h_d)$ with state space X_d, input space U_d, output space Y_d, transition function f_d, and sensor model h_d. Describing the physical dynamics, f_d also includes domain-specific behavior that is challenging to model precisely. While the goal is to achieve similar dynamics across domains, i.e., $f_s \approx f_v \approx f_r$, full-scale dynamics will thus inevitably differ from simulation and scaled replicas. Further, influences from other domains z_d and disturbances w_d affect each domain, such that f_d can be written as in (1).

$$\dot{x}_d = f_d(x_d(t), u_d(t), w_d(t), z_d(t)) \tag{1}$$

The states x_d of each domain can be observed via a sensor model h_d returning the observed system state y_d depending on the sensor noise v_d described in (2). Since sensor characteristics and noise properties differ across domains, measures need to be taken to maintain inter-domain consistency, e.g., by emulating the noise term v_d.

$$y_d = h_d(x_d(t), v_d(t)) \tag{2}$$

For compatibility of domains, state mappings $\phi_{i\leftarrow j} : X_i \to X_j$, input mappings $\psi_{i\leftarrow j} : U_i \to U_j$ and output mappings $\gamma_{i\leftarrow j} : Y_i \to Y_j$ betwen arbitrary domains i and j are defined. Finally, the *Digital Triplet* $\mathcal{T}$ can be written as

$$\mathcal{T} = \Big(\{\Sigma_d, C_d, \theta_d\}_{d\in\mathcal{D}}, \{\phi_{j\leftarrow i}, \psi_{j\leftarrow i}, \gamma_{j\leftarrow i}\}_{i\neq j}\Big). \tag{3}$$

Here, θ_d denotes the time synchronization function for the inter-domain influence function C_d, providing

$$z_d = C_d(\ \{\phi_{j,d}(x_j(\theta_j(t)))\}_{j\neq d}\ ). \tag{4}$$

Due to delays in this distributed system, θ_d is required to align timestamps in different domains and entities. Although exact physical synchronization is not required, the logical time of the *Digital Triplet* must progress consistently with physical time, since real and scaled entities experience state transitions f_d in real time. If this is not guaranteed, similarity between domains deteriorates. Thus, sufficiently established time synchronization needs to be assured.

From a system-theoretic perspective, the *Digital Triplet* represents a network of coupled dynamical systems. The mappings $\phi_{j\leftarrow i}$, $\psi_{j\leftarrow i}$, and $\gamma_{j\leftarrow i}$ ensure compatibility of states, inputs, and observations across domains, while the coupling function C_d enables cross-domain interaction. This formulation allows algorithms developed in one domain to be evaluated under consistent closed-loop conditions across real, scaled, and simulated systems.

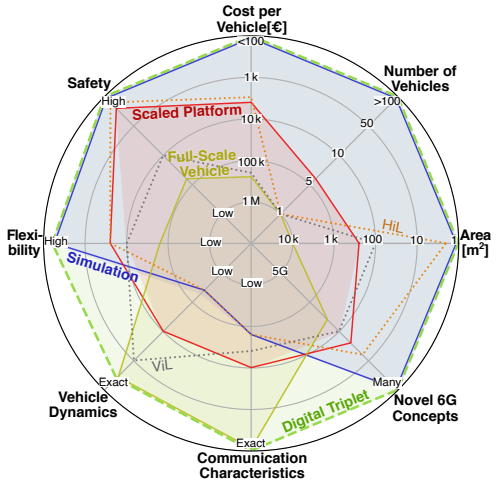


FIGURE 3. Comparison of Hardware-in-the-Loop (HiL), Vehicle-in-the-Loop (ViL), scaled, full-scale, and simulated approaches across multiple categories, illustrating the *Digital Triplet* combining the complementary strengths of real, scaled, and virtual domains.

While simulations offer advantages in scalability and safety, full-scale vehicles provide realistic communication and vehicle dynamics, as illustrated in Fig. 3. Scaled vehicles occupy an intermediate position, offering real-world dynamics while remaining more cost-efficient, safer, and reproducible than full-scale vehicles. Thus, they form a practical bridge between simulation-based development and full-scale validation. By integrating the complementary strengths of real, scaled, and simulated environments, the *Digital Triplet* achieves the largest combined capability area.

In contrast to Hardware-in-the-Loop (HiL) and Vehicle-in-the-Loop (ViL) approaches [48], the proposed *Digital Triplet* does not merely replace missing real-world components. Instead, entities from all domains operate simultaneously within a synchronized scenario and can mutually influence each other. Thus, HiL approaches cannot provide physical accuracy, reproducibility, safety, and scalability at the same time.

Meanwhile, digital twins are typically defined as paired representations of a physical system with bidirectional data exchange. The *Digital Triplet* extends this concept by synchronizing multiple domains simultaneously and enabling mutual interaction between them. While digital twins may exist as subsystem components within the *Digital Triplet*, the two concepts are structurally different. Consequently, a single digital twin cannot provide the same combination of realism, scalability, and physical fidelity as the *Digital Triplet*.

B. IMPLEMENTED DIGITAL TRIPLET INSTANCE

Practical implementations of the *Digital Triplet* cannot achieve perfect synchronization or dynamic equivalence between domains, as real-world deployments require practical compromises. However, the described functionalities can be adopted in a modular fashion. In the following, we describe our implementation of the *Digital Triplet* concept. As illustrated in Fig. 4, communication within the *Digital Triplet* is facilitated through Robot Operating System 2 (ROS 2) using

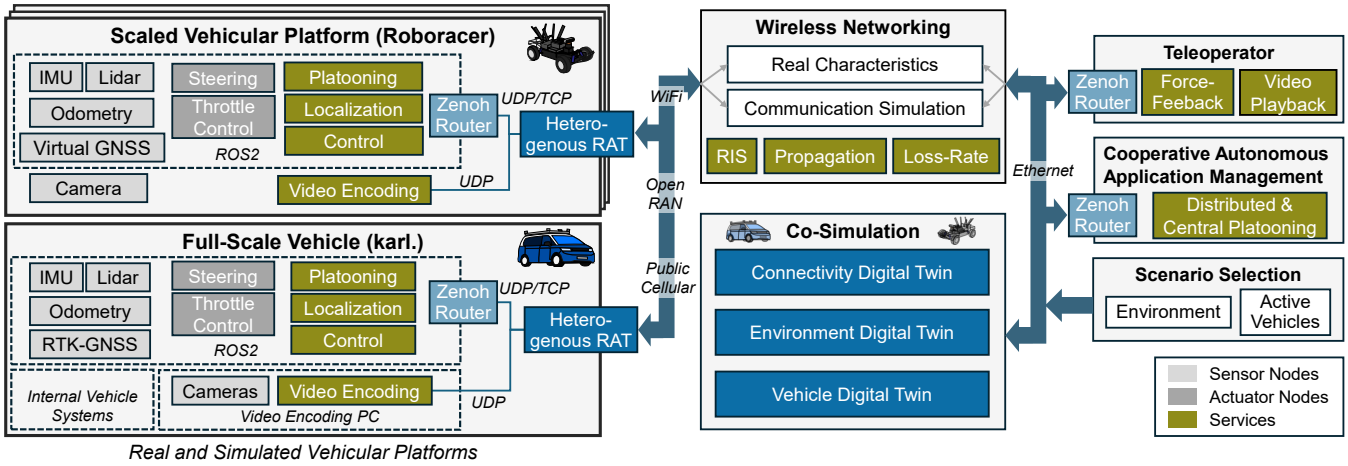


FIGURE 4. Implemented *Digital Triplet* system architecture for evaluating C-ITS use cases combining scaled and full-scale vehicles and co-simulation.

Zenoh as the ROS Middleware (RMW). In contrast to other RMWs, *Zenoh* is more flexible in configuring the routing of ROS 2 messages and allows for choosing the transport protocol. By selectively routing messages only to relevant nodes instead of using broadcasts, network overhead and latency are decreased. Regardless of the domain, each vehicle is equipped with onboard computing systems running ROS 2 nodes for perception, localization and control. These nodes interact over heterogeneous Radio Access Technologies (RATs), including WiFi and public or private mobile network links. This allows real and simulated entities to coexist in a shared environment with realistic network behavior.

The three layers of the *Digital Triplet* interconnect via multiple digital twin entities that model vehicles, communication channels, and the surrounding environment. The connectivity digital twin represents the underlying mobile network by emulating radio channel characteristics and extends to Reconfigurable Intelligent Surfaces (RISs) (cf. Sec. III-C4). The vehicle digital twin mirrors the dynamics and sensor data of real and scaled vehicles, while the environment digital twin integrates additional simulated and real objects into the scenario. This setup allows for the analysis of hypothetical real-world scenarios, enabling systematic what-if evaluations.

The architecture's modular design allows each layer to function independently or in synchronized operation, depending on the requirements of the specific test scenario. A cooperative application management entity coordinates interactions between automated control algorithms and human teleoperators, enabling both fully autonomous and human-in-the-loop evaluations. Depending on the configuration, the same control logic can be deployed to real vehicles, their scaled counterparts, or entirely simulated vehicles. This interchangeability allows evaluating specific aspects of services under equivalent operating conditions.

Overall, the *Digital Triplet* approach introduces a tightly coupled integration of communication, computation, and control domains. By combining real vehicles with scaled and virtual representations within a unified framework, it enables reproducible and scalable experimentation, accelerated development cycles, and comprehensive validation of C-ITSs.

While the proposed *Digital Triplet* architecture is conceptually not restricted to a fixed number of entities, its scalability is practically limited by several factors. Each additional entity increases the simulation's computational load, although this can be mitigated by allocating additional compute resources. In the physical domain, additional scaled vehicles require corresponding hardware and localization capacity in the testbed. In many configurations, since vehicle status messages are small, the communication network is not expected to be a limiting factor. If scaled massively, central communication entities like *Zenoh* routers, discovery mechanisms, and the communication network may become a bottleneck if not deployed in a distributed configuration. While scaling with additional scaled vehicles is possible, scaling through virtual vehicles can be extended much further. The *Digital Triplet*, therefore, enables selecting an appropriate trade-off between experimental realism and scalability.

In our concrete deployment, the subsystems of the *Digital Triplet* are interconnected using mobile networks. For the real vehicle, either a commercial private 5G NSA network or a public mobile network is used. In case of the scaled vehicles, an *O-RAN*-based private network is used, parameterized as listed in Table II. To minimize potential influence on results, measurements are controlled via an additional WiFi connection. The full-scale and scaled test fields are connected over the public internet through a dedicated *WireGuard* Virtual Private Network (VPN) tunnel using a public IP endpoint. Time synchronization, especially for synchronized measurements, is provided by distributing Global Navigation Satellite System (GNSS)-based timestamps in the subsystems. For the full-scale vehicle, a GNSS-based time synchronization via Precision Time Protocol (PTP) is deployed at the vehicle, and a PTP Grandmaster Ethernet switch at the operator. In the scaled test field, we use a dedicated Ethernet network for high-frequency time synchronization during the remote driving latency measurements using a local Network Time Protocol (NTP) server. In case of the other case studies, NTP via the 5G network is used.

TABLE II. Parameterization of the Used Wireless Communication Systems

Parameter	Scaled Test Field	Full-Scale Test Field	Simulated Test Field
Technology	5G Standalone	5G NSA	5G mmWave
Frequency	3.75 GHz (n78)	1.77 GHz (B3) / 3.45 GHz (n78)	28 GHz (n257)
Bandwidth	100 MHz	20 / 50 MHz	50 MHz
Subcarrier Spacing	30 kHz	15 / 30 kHz	60 kHz
Duplex Type	TDD	FDD / TDD	TDD

C. COMPONENTS & TEST FIELDS OF THE DIGITAL TRIPLET

As described, the *Digital Triplet* framework integrates three main vehicular domains: a full-scale automated vehicle in an outdoor test field, scaled vehicular platforms in an indoor test field, and a virtual test field with network and vehicle co-simulation. These parts will be detailed in this subsection.

1) Full-Scale Automated Vehicle in Outdoor Test Field

For the full-scale part of the *Digital Triplet*, the automated and connected research vehicle *karl.* [49] is used (cf. Fig. 5a). *karl.* is a *Volkswagen Multivan* retrofitted with a drive-by-wire system for automated longitudinal and lateral control, a sensor suite including 360° camera coverage, as well as high-performance computing and communication hardware.

Eight *StereoLabs ZED X* stereo cameras are connected via GMSL2 to two dedicated *NVIDIA Jetson AGX Orin* developer kits, which handle full image processing for the cameras. For mobile communications, an *Ericsson Cradlepoint R1900* automotive-grade dual-active 5G router is connected to a roof-mounted 4×4-MIMO antenna. The entire automated driving software stack is based on *ROS 2* and is containerized using *Docker*. More details on the vehicle's hardware and software architecture can be found in [49].

In order to maintain a controlled environment for testing and validation, *karl.* is operated at the Aldenhoven Testing Center (ATC),[1] Germany. At the ATC, a commercial-grade private 5G network (cf. Table II) is operated, which is used for all communication to and from the full-scale vehicle in this work, unless otherwise specified.

2) Small-Scale RoboRacer in Vehicular Indoor Test Field

The open source scaled vehicular platform *RoboRacer* (formerly F1TENTH) [50, 51] is extended within this work. In Fig. 5b, the deployed customized vehicle is shown. The platform is centered around an *NVIDIA Jetson* System-on-a-Chip (SoC), connected to different sensors and actuators, powered by a lithium battery, as shown in Fig. 6. This setup allows it to replicate selected functionalities of the full-scale vehicle. As shown in Table III, the scaled vehicle approximately follows the 1 :10 scaling factor with respect to key vehicle parameters.

This relationship can also be illustrated analytically using the kinematic bicycle model [52]. Under geometric scaling $\phi_{s\leftarrow r}$ with factor $\lambda = 0.1$ and proportional velocity scaling ($v_s = \lambda v_r$), the model equations remain form-invariant, yielding identical normalized curvature and yaw-rates for both vehicles. Consequently, they follow geometrically similar trajectories for the same steering inputs. However, dynamic similarity is not preserved. To show this, we employ the Froude number $Fr = v/\sqrt{gR}$, originating from fluid-dynamics, which characterizes the ratio between inertial and gravitational forces g, dependent on a length R. When applying it to our system (R is the curve radius), the scaling implies $Fr_s = \sqrt{\lambda} \cdot Fr_r$, yielding that the real vehicle experiences higher g-forces. Thus, the lateral acceleration must be limited in the scaled test field to prevent slip and dynamic saturation in full scale. While perfect dynamic similarity cannot be achieved, the platform captures the relevant system characteristics within the considered operating regime, enforced by constrained dynamics. However, the original *RoboRacer* platform exhibits several limitations that prevent it from adequately reproducing the desired behavior, motivating the proposed hardware and software improvements described below.

Critically, original *RoboRacer* scaled vehicles lack mobile network connectivity and were observed to be unable to replicate full-scale vehicle dynamics when accelerating from a standstill, both of which are crucial for reliable, low-latency operation. Furthermore, with the default RMW configuration, ROS 2 topics are broadcast over the limited wireless channel, and safety features on connection loss are lacking. Thus, several improvements have been implemented compared to the original BOM and code of the *RoboRacer* project [50].

Formerly, the *VESC* motor controller of the *RoboRacer* operated with a sensorless brushless DC motor, requiring the motor state to be inferred rather than directly measured.

(a) Full-scale research vehicle *karl.*

(b) Scaled vehicular platform *RoboRacer*

FIGURE 5. Full-scale automated and connected research vehicle *karl.* and scaled representation of one of six customized *RoboRacer* vehicles.

[1] https://www.aldenhoven-testing-center.de/en/tracks/urban-environment.html (accessed: Jan. 15, 2026)

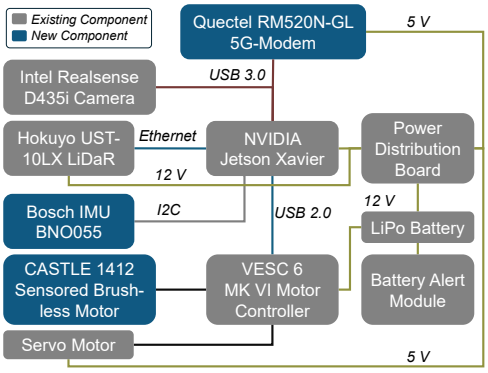


FIGURE 6. Architecture of the modified *RoboRacer* system.

TABLE III. Comparison between physical parameters and limitations of driving dynamics between the full-scale and the scaled vehicles in their test bed; control limits of the full-scale vehicle relate to automated operation and may be limited by its drive-by-wire system for safety reasons; steering angles relate to a single-track model with Ackermann steering [49]

Parameter	Full-Scale	Scaled 1/10	Scale	Opt. Scale
Wheelbase L_w	3.124 m	0.325 m	9.61	10
Track Width T_w	1.658 m	0.24 m	6.91	10
Max. Speed $v_{\max}$	130 km/h	150 km/h	1.27	$\leq$10
Max. Acceleration $a_{\max}$	2.0 m/s^2	9.51 m/s^2 [51]	0.21	$\leq$10
Max. Deceleration $a_{\min}$	4.5 m/s^2	9.51 m/s^2 [51]	0.47	$\leq$10
Max. Steer. Angle $\delta_{\max}$	$\pm$31°	$\pm$24° [51]	1.29	$\leq$1
Max. Steer. Rate $\dot{\delta}_{\max}$	—	183 °/s [51]	—	$\leq$1
Localization Accur. Δp	1 cm	<1 mm	$\geq$10	$\geq$10

During spin-up, the lack of immediate rotor position feedback can lead to slower start-up behavior and increased latency between commanded and actual velocity. Thus, a new motor with Hall-encoded absolute motor orientation feedback is integrated. This allows for precise closed-loop control and, thus, reduces driving command execution latency.

As ROS 2 *Jazzy* with *Zenoh* is used in the *Digital Triplet*, the *RoboRacer* code was ported to *Jazzy* to enable full interoperability between domains. An emergency brake mechanism was implemented within the motor controller driver for additional safety: Firstly, if control commands are suspended, an immediate emergency braking is initiated. Secondly, an enable signal is issued regularly to activate driving functions. The code is openly available.[2]

To decrease overall system complexity and reduce latency, a custom kernel for the *Jetson* on the scaled vehicles has been built. It features high-speed modem compatibility with the *qmi_wwan* driver based on instructions of the modem manufacturer [53], and adds kernel-level *WireGuard* support. This eliminates the need for an additional SoC for mobile network connectivity previously required [23], while enabling direct access to network KPIs from the attached modem. We provide instructions for modifying the kernel as described.[3]

The scaled vehicles are deployed in a local indoor test field,[4] measuring over 10×20 m. It is equipped with a laser projection system, an optical Motion Capture (MoCap) system with submillimeter accuracy and an O-RAN 5G SA commercial private network (cf. Table II).

[2]https://tiny.cc/roboracer-ros2-jazzy
[3]https://tiny.cc/jetson-wwan-kernel
[4]https://flw.mb.tu-dortmund.de/research/innolab (accessed: Jun. 6, 2026)

(a) Portable operator workplace: an ultra-widescreen monitor and a tablet to assess the scene, a steering wheel and pedals remotely control the vehicle.

(b) Remote driving Graphical User Interface (GUI) of operator workplace showing a stitched frontal (and rear/side) multi-camera view: speed, steering wheel angle, and system engagement status are displayed.

(c) Operator workplace at the scaled test field set up for remote driving with co-simulation to the side.

FIGURE 7. Modular operator workplaces for remote driving.

3) Operator Workplace for Remote Driving

A modular and reconfigurable operator workplace has been developed to enable safe and intuitive remote control of vehicles in all domains within the *Digital Triplet* framework. It aims to provide a low-latency, high-fidelity human-machine interface that is portable, flexible, and adaptable to different remote operation use cases and research studies.

The operator workplace shown in Fig. 7a is built around a rigid aluminum frame made from extrusion profiles, allowing rapid reconfiguration without redesign. In its current configuration, a single 49-inch 32:9 ultra-widescreen monitor serves as the primary visual interface, offering a wide-angle view of the vehicle's surroundings. The operator GUI combines a multi-camera view with telemetry data, as shown in Fig. 7b.

For proper scene awareness, an image-stitching algorithm fuses an arbitrary number of calibrated camera feeds into a panoramic view using planar or equirectangular projections, available as open source.[5] The video streams shown in Fig. 7 are built around *GStreamer* (cf. Sec. IV-A).

For the control path, a *Fanatec CSL DD* direct-drive wheelbase and *Fanatec CSL Elite* pedals provide high-resolution steering, throttle, and brake input with precise haptic feedback of steering forces. A dedicated ROS 2 driver is also available as open source.[6] Like the rest of the *Digital Triplet*, the containerized software stack of the operator workplace is based on ROS 2 *Jazzy* with RMW *Zenoh* for seamless integration.

A second operator workplace was built, co-located with the scaled test field. Shown in Fig. 7c, it features three displays for live video streams, scaled test field observability, and visualization of the co-simulation of the *Digital Triplet*. Despite differences in physical layout, both workplaces operate identically thanks to modular hardware abstraction and standardized ROS 2 interfaces. Both operator workplaces have proven successful in remotely driving the full-scale research vehicle *karl.* at the ATC real-world test field.

4) Virtual Test Field and Network Co-Simulation

The virtual test field provides a unified simulation environment that integrates full-scale and scaled vehicle models within a common virtual framework. This approach enables consistent co-simulation of all vehicle types at a single virtual scale, ensuring that interactions z_d can be analyzed under comparable conditions. Spatial consistency across domains is ensured through real-time coordinate transformations $\phi_{i\leftarrow j}$, implemented as ROS 2 nodes. The environment also supports the simulation of further virtual vehicles and dynamic obstacles, thereby extending the range of test scenarios. The 3D simulation environment is implemented in *Unity*, which provides advanced visualization, physics simulation, and real-time rendering. A digital model of the ATC has been integrated to replicate realistic test-track conditions.

In the scaled and virtual test field, the network connectivity is simulated to reflect real-world characteristics. To demonstrate the capability and extensibility of the proposed *Digital Triplet* system, the test field incorporates a virtual mmWave communication system (cf. Table II). At the shorter wavelengths λ in Frequency Range 2 (FR2), significantly more bandwidth is available, potentially allowing higher data rates. However, FR2 also poses challenges related to signal attenuation and shadowing. The characteristics of the FR2 channel are simulated using the Urban Micro (UMi) model [54] in combination with Line-of-Sight (LoS)-detection to replicate communication paths in real time. Recent outdoor 28 GHz measurements in dense urban environments report a Root Mean Squared Error (RMSE) of approximately 11.9 dB between the measured path loss and the 3GPP UMi model in Non-Line-of-Sight (NLoS) conditions, with links exhibiting deeper shadowing than predicted by the model [55]. To account for these empirically observed conditions, an additional margin of 11.9 dB is applied to NLoS links.

Large-scale RISs with a configurable size $a \times b$ (width times height) can be deployed within the simulated scenario to mitigate shadowing effects in NLoS areas. Since large-scale RIS systems are not yet commercially available, the simulation relies on assumptions and represents a best-case estimate of the achievable performance. In particular, instantaneous reflector alignment to the optimal configuration available to the RIS is assumed. The goal is not to predict exact real-world performance but to demonstrate how the *Digital Triplet* enables testing of future communication technologies before real hardware becomes available. Nevertheless, practical limitations such as discrete reflection directions and phase quantization are modeled. Regardless of the RIS hardware, the reflected signal strength is affected by the incoming (in) and outgoing (out) azimuth ϕ and elevation angle θ of the signal relative to the RIS, described in (5).

$$\gamma_{\text{angle}} = \cos(\phi_{\text{in}}) \cdot \cos(\theta_{\text{in}}) \ \cdot \ \cos(\phi_{\text{out}}) \cdot \cos(\theta_{\text{out}}) \tag{5}$$

Based on the radar equation, specific RIS models of [56] were extended to three-dimensional space [57] to replicate far-field reflection characteristics. In case of near-field reflections, when the distance r_{out} between RIS and receiver is below the Fraunhofer distance $d_{\text{F}} = 2 \cdot \max(a^2, b^2)/\lambda$ [58], a correction term ρ_{NF} is applied to account for the lower effective radar cross-section of the reflector [59], as shown in (7).

$$\rho_{\text{NF}} = \begin{cases} 1, & r_{out} \geq d_F \\ (r_{\text{out}}/d_F)^2, & r_{\text{out}} < d_F \end{cases} \tag{6}$$

$$RCS_{\text{eff}} = \frac{4\pi \cdot (a \cdot b)^2}{\lambda^2} \cdot \gamma_{\text{angle}} \cdot \rho_{\text{NF}} \cdot \rho_{\text{HW}} \tag{7}$$

Further, a hardware correction factor ρ_{HW}, defined in (8), is applied. It models reflection inefficiencies η_{R} due to inherent losses, beam misalignment η_{A} due to limited discrete alignment configurations, and phase quantization imperfections η_{Q}, which can be approximated as $\eta_{\text{Q}} = \text{sinc}^2(1/K)$ for $K \in \mathbb{N}^+$ quantizations for RISs with many elements [60, 61].

$$\rho_{\text{HW}} = \eta_{\text{R}} \cdot \eta_{\text{A}} \cdot \eta_{\text{Q}} \tag{8}$$

By combining the antenna gains G_{TX} and G_{RX}, the path loss $PL_{\text{TX-RIS}}$ from the base station to the RIS, and the path loss $PL_{\text{RIS-RX}}$ from the RIS to the vehicle, the total path loss PL_{total} can be calculated according to the radar equation (9).

$$PL_{\text{total}} = G_{\text{TX}} \cdot G_{\text{RX}} \cdot PL_{\text{TX-RIS}} \cdot RCS_{\text{eff}} \cdot PL_{\text{RIS-RX}} \cdot 4\pi/\lambda^2 \tag{9}$$

For the path loss between base station and reflector, a free-space path loss model is used, as a LoS connection is guaranteed by the reflector placement. The selected configuration of the evaluated scenario is summarized in Table IV.

The mapping of the signal attenuation to an achieved data rate can be calibrated based on measurements as in [62] and calculated as specified in [63]. *Unity* is connected with ROS 2 via the *ROS-TCP-Endpoint* interface [64]. Thus, simulated

[5] https://tiny.cc/image_reprojection
[6] https://tiny.cc/fanatec_ros_driver

TABLE IV. Network Co-Simulation Parameters

	Parameter	Value
BS	Height	15 m
	Antenna	8 × 8 Phased Array
	Peak Gain	25.6 dB
UE	Height	1.5 m
	Antenna	Omni-directional Rod
RIS	Height	6 m
	Antenna	Optimum Reflection Pattern
	Size	1 m × 1 m
	Hardware Loss (ρ_{HW})	η_R= 3.1 dB, η_A=0.7 dB [60] $\eta_Q = \mathrm{sinc}^2(1/6)$=0.4 dB [61]
Channel	BS-UE	UMi
	BS-RIS	Free-Space Path Loss
	RIS-UE	UMi
	NLoS Correction	11.9 dB

FR2 connectivity restrictions can be communicated and applied to full-scale and scaled vehicles via emulation, given the underlying real communication systems provide a sufficient data rate. The respective case study is presented in Sec. V-A.

IV. METHODOLOGY OF SELECTED USE CASES

The following section outlines the methodology applied in the three selected use cases. Each focuses on a specific challenge of future C-ITS systems, from ultra-reliable video streaming over precise localization to coordinated vehicle control.

A. RELIABLE VIDEO STREAMING FOR TELEOPERATION

For teleoperation, a reliable and low-latency video stream is crucial for safe operation. With this goal in mind, several works extended traditional constant bitrate encoding approaches. However, these approaches lack on-device congestion feedback for immediate reaction in the event of worse-than-expected network conditions. To overcome these limitations of state-of-the-art approaches, Reliable Intelligent Stream Encoding (RISE) was introduced in [65]. It combines proactive control of the video stream's bitrate within allowed limits using predictions and congestion feedback from on-device buffers at the sender. Compared to traditional approaches waiting for feedback from the stream receiver, RISE reduces latency and guarantees service availability without unnecessarily degrading video quality. To reduce delay during congestion-based adaptations, the transmit buffer size is additionally reduced (c.f. Appendix A). By integrating the *SEAMLESS* multi-link protocol, RISE can leverage multiple communication networks and technologies, reducing the probability of insufficient resources [66]. Commonly used software-based encoding might suffer from encoding delay and jitter due to other computational loads. As shown in Sec. II, some existing works also experienced a comparably high latency necessitating optimization.

1) Low-Latency Adaptations

For adaptability of the video streaming pipeline, the open source *GStreamer* [67] library is used for video processing and streaming. All used *GStreamer* elements have been fine-tuned by minimizing the latency while maintaining a sufficient video quality. This includes a deep dive into encoder and parameter selection, as detailed in Appendix A. For performance reasons, we use the *NVENC* hardware encoder, enabling low-latency, high-quality streams while reducing computational load. Hardware encoding delay is largely independent of the bitrate, allowing for the changes applied by RISE without affecting the delay. On the receiver side, the decoding and visualization pipeline has also been optimized for low latency. This includes using a GPU-based hardware decoder for the video stream and an *OpenGL* display sink.

For the latency evaluations, we configure a video bitrate below the channel capacity. In a real deployment the prediction and adaptation mechanisms of RISE would dynamically guarantee this. Lower channel quality or less scheduled resources would thus result in a reduced bitrate rather than increased latency, isolating the intrinsic system latency.

2) Teleoperation Latency Measurement Setup

The setup to assess the teleoperation E2E delay consists of time-synchronized *GStreamer* pipelines, custom ROS 2 and *GStreamer* latency measurements, and a G2G delay measurement system, summarized in Fig. 8. The G2G latency of the video stream of the scaled vehicle is measured using the setup proposed in [68]. The measurement system is calibrated to compensate for the intrinsic delay of the sensing hardware and measurements are conducted for at least 60 s.

To distinguish the contributions of all parts of the transmission pipeline, the delays of all elements used in the sender and receiver *GStreamer* pipelines are logged. However, this way, the exact network latency is still not captured. Due to the dependence of the delay on the exact payload and transmission characteristics, e.g., Internet Control Message Protocol (ICMP) latency measurements are not sufficient for measuring the video streaming network delay. Also, in mobile networks, there are significant differences in uplink and downlink delay due to scheduling procedures in the uplink. Thus, measurements of the video stream delay in the uplink and of control command delays in the downlink must be performed. For this, we developed custom *GStreamer* plugins to generate and read Real-time Transport Protocol (RTP) headers (cf. Fig. 8). These are filled with the current, synchronized timestamp t_v on video transmission. To reduce overhead to a minimum, the timestamp is injected as the number of milliseconds since midnight and only if the RTP marker bit is set (only once per video frame [69]). At the operator, the receiving timestamp t_c' is captured and sent back to the vehicle, along with the original transmission timestamp, via User Datagram Protocol (UDP). There, a third timestamp t_r is added, and thus the uplink latency τ_2 and downlink latency τ_4' can be calculated (cf. Table I):

$$\tau_2 = t_c - t_v, \quad \tau_4' = t_r - t_c', \quad \tau_4 = t_r - t_c. \tag{10}$$

Additionally, we measured the delay of ROS 2 control commands in the downlink τ_4 by setting the timestamp of the

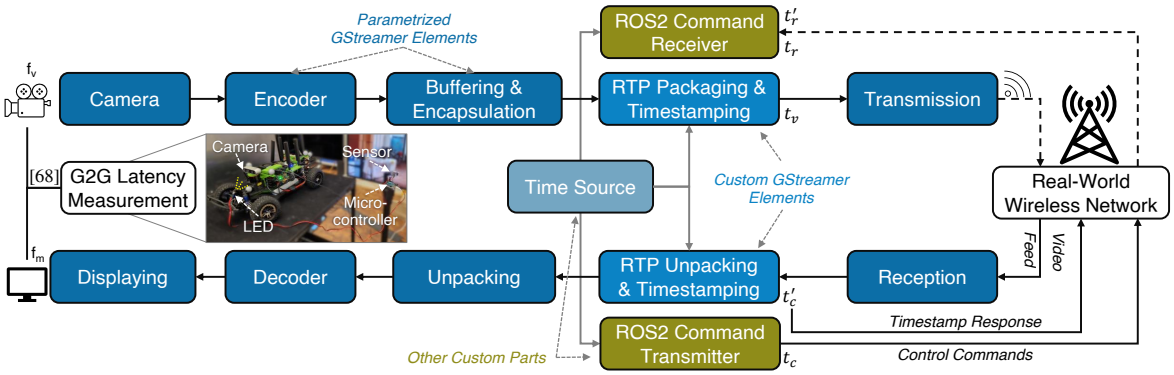


FIGURE 8. Video streaming setup and latency measurement methodology for quantifying subsystem and E2E latencies in teleoperation.

topic to t_c at the sender. We tested *Zenoh* over UDP and TCP with each over 100 analyzed control messages. The command was issued at a variable rate of 10–100 Hz, showing no difference in latency, in accordance with the results of [5] (and behaving similarly to τ'_4). Due to a smaller overhead, finally, UDP is used and a rate of 100 Hz. Delay measurements on a local host system showed that *Zenoh* introduces less than 1 ms of delay. Network delay measurements were validated using the Two-Way Active Measurement Protocol (TWAMP). We ensured correct time synchronization θ_d with well below 1 ms deviation, as described in Sec. III-B, as offsets would directly affect the measurement.

In addition to the network and encoder delays, the camera and the display contribute to the overall latency. The delay of the camera depends on the selected video frame rate f_v. Since the camera can only expose for less than one frame interval, the worst-case age of information at the camera is one frame time, and the median age of information may be estimated as half the frame time. Similar effects occur at the monitor, depending on its refresh rate f_m. Additionally, the monitor's response time τ_{resp} further delays the display of the video. We model the monitor reaction time δ_m as shown in (11).

$$\delta_m = \frac{1}{2 \cdot f_m} + \tau_{\text{resp}} \tag{11}$$

For the cameras, higher frame rates imply lower resolutions due to hardware constraints. We selected the highest respective frame rate for the latency evaluation. Measuring the G2G latency confirmed the relationship between frame rate and additional delay. Thus, latency can be balanced with resolution depending on current needs.

B. REALISTIC LOCALIZATION IN THE DIGITAL TRIPLET

In the scaled test field, a precise *Vicon* MoCap localization system is deployed to track the scaled vehicles with sub-mm accuracy. The MoCap setup monitors multiple retro-reflective spherical markers attached to each vehicle (cf. Fig. 5) with a rate of up to 200 Hz. This system replaces the GNSS-based localization used in the full-scale vehicle and is even more precise and reliable. However, while the MoCap system provides continuous and highly accurate position updates in the scaled test field, real-world GNSS systems may experience degraded availability or reduced update rates. In such cases, localization must rely on onboard sensors such as wheel odometry and the Inertial Measurement Unit (IMU). Those sensors are likewise integrated into the scaled vehicle (cf. Fig. 6), ensuring functional equivalence.

To achieve consistent behavior across both domains, an Extended Kalman Filter (EKF)-based sensor fusion is deployed on the scaled vehicle, aiming to replicate the behavior of the RTK-GNSS of the full-scale vehicle. The EKF combines fast, locally available data with slower, global position updates to maintain an accurate and continuous state estimate, even under intermittent localization signals. The position data are converted into GNSS-compliant message formats to emulate the real vehicle's localization interface. We use the ROS 2 robot localization package [70], to fuse multiple asynchronous sensor streams. Two EKF nodes are deployed that fuse the local IMU and odometry data, and global MoCap information. The EKFs are configured for planar motion, estimating the vehicle's x and y position, yaw angle, and their corresponding velocities. To balance estimation accuracy and computational efficiency on the embedded platform, the local EKF runs at 100 Hz, while the global EKF operates at 20 Hz.

For emulating realistic GNSS conditions, the MoCap signal is intentionally degraded by introducing undersampling and localization noise. This serves as a state mapping $\phi_{S \leftarrow R}$, allowing for systematic evaluation of the EKF accuracy and robustness under degraded conditions. Longer GNSS outages are not explicitly modeled, as these fall outside the nominal operational envelope of cooperative platooning and would probably trigger a controlled stopping or the use of local sensors. Further, common-mode GNSS biases are not considered, as they do not affect the relative spacing in the platoon.

Consequently, the scaled test field replicates real-world localization challenges more accurately, improving domain consistency compared to related work [3].

C. COOPERATIVE INTERACTION AND COMPUTATIONAL OFFLOADING THROUGH PLATOONING

Another capability of the *Digital Triplet* is the evaluation of platooning control for multiple vehicles in interaction of full-scale, scaled and simulated vehicles. Platooning serves as a representative and demanding application, as it requires tight coupling between vehicles, low-latency communication,

and consistent control performance. Platooning introduces challenges, such as corner-cutting effects, where cars exhibit increasingly rounded curves downstream of the platoon.

Hence, this work employs an NMPC-based algorithm that prevents cutting corners through its planning capabilities and considers hardware constraints including velocity, steering angle, and their respective change rates. Furthermore, the proposed algorithm enables custom fine-tuning of penalization for trajectory and headway tracking. Safety constraints are another vital point that can be implemented using NMPC as future additional constraints to provide safety guarantees. As the NMPC repeatedly calculates the best trajectory based on the current state, moderate model mismatch is corrected continuously. Further, the input and system state parameters are limited to prevent slip and significant dynamic differences between vehicles and domains (cf. Sec. III-C2). The detailed working principles of the employed NMPC, as well as robustness and string stability discussions, and detailed parameterization can be found in Appendix B.

Beyond cooperative control, the platooning scenario also enables the investigation of computational offloading as a key enabler for future 6G systems. The repeated optimization of the NMPC results in a considerable computational load, making it a suitable candidate for edge execution. We therefore evaluate the deployment on the edge and on the vehicles. This means that state updates and, in case of the edge server, also controller results need to be transferred over the mobile network, resulting in slight delays. The impacts of these delays are analytically analyzed in Appendix C, showing that they are manageable if limited and sufficient margins are provided. Thus, the presented setup can serve as a benchmark for evaluating distributed and centralized control paradigms under realistic network conditions.

We extended the platoon of three vehicles in [2] to six in this work to evaluate the behavior of larger platoons. For reproducibility, we recorded a reference track and replayed it as a virtual leader during evaluation.

V. SELECTED RESULTS AND INSIGHTS

The developed *Digital Triplet* system enables studying a variety of relevant topics in the field of C-ITS on three interconnected abstraction levels. In this section, we use the *Digital Triplet* to analyze the performance of the three case studies remote driving, localization, and platooning. Each case study highlights a specific capability of the *Digital Triplet*.

A. RESILIENT CHANNEL-ADAPTIVE TELEOPERATION

Future C-ITS depend on reliable connectivity. The increasing expansion and the rising demands of mass-deployed mobile applications can surpass the offered capacity of conventional FR1 systems necessitating mmWave connectivity. However, FR2 systems experience high path loss compared to FR1 systems in NLoS situations, which reduces the range and reliability of FR2 connections. This makes it crucial to extend the coverage, which can be achieved using RISs. As different configurations and scenarios of potential future deployments

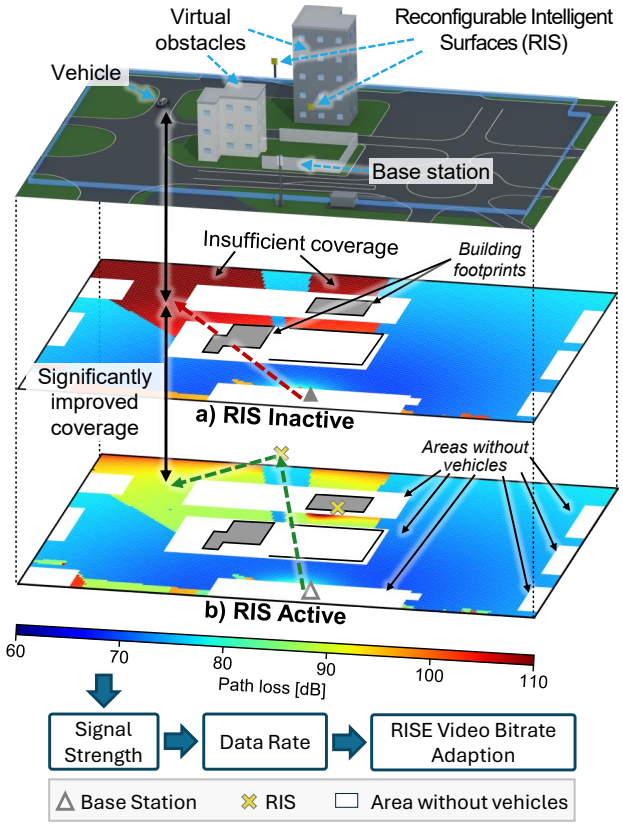


FIGURE 9. Co-simulated mmWave path loss with virtual obstacles (a) and RISs (b), extending the coverage of a FR2 base station into NLoS areas, combining connectivity and environment digital twins to emulate available data rates to test video bitrate adaptation mechanisms such as RISE [65].

can hardly be thoroughly tested in the real world, a simulated channel is emulated using the *Digital Triplet*. The path loss of the simulated channel is converted to a signal strength measured at the vehicle and an available data rate published via ROS 2 messages. RISE is used to adapt the video stream bitrate accordingly. In other configurations, virtual Spatially distributed Traffic and INterference Generator (vSTING) or similar concepts can be used to apply the data rate restrictions on the physical interface via *Linux* traffic shaping [71].

As shown in Fig. 9, a virtual mmWave base station is placed in the digital representation of the ATC. Due to two virtually placed buildings, areas are shadowed such that the attenuation becomes critically high. This attenuation could result in a connection loss, making remote driving infeasible. Deploying two RISs extends coverage into these NLoS areas, increasing the signal strength by over 20 dB, maintaining connectivity.

However, the RIS effectiveness depends on the incident wave angle relative to the reflector. This particularly impacts locations with steep angles, as seen in the left rear corner of Fig. 9(b). Additionally, the gain of the reflector is reduced if the vehicle is close. Both effects need to be taken into consideration for a successful coverage extension via RISs.

By combining adaptive video coding to proactively adapt the bitrate and actively increasing the signal strength in NLoS areas with RISs, the resilience of remote driving deployments can be improved drastically.

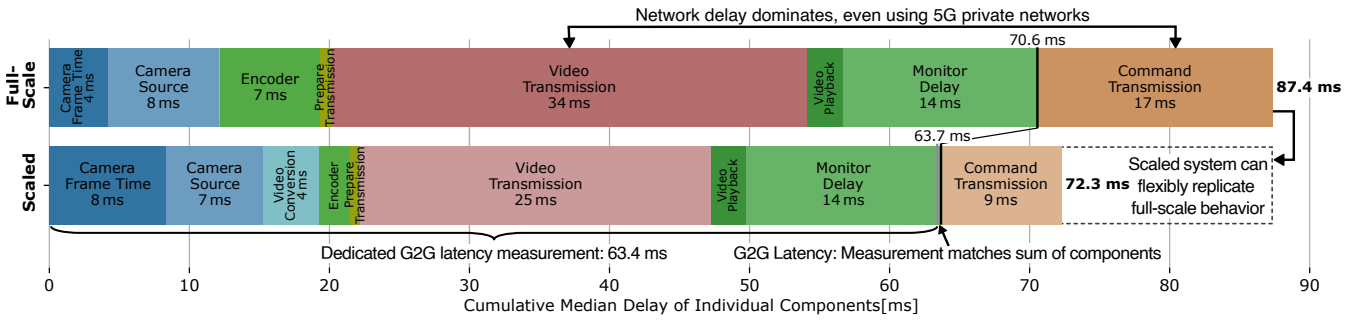


FIGURE 10. Cumulative teleoperation E2E-delay for the full-scale vehicle and the scaled vehicles in their respective test fields, both well below the 5GAA requirement of 120 ms [7, 8] (monitor and camera frame delays based on device parameters).

B. END-TO-END LOW-LATENCY TELEOPERATION

We performed G2G and subsystem latency measurements for our proposed setup and captured the individual delays of the remote driving system covering the video streaming as well as the command transmission, as described in Sec. IV-A.

The E2E delay of the full-scale and scaled setups is comparable, differing mainly in network latency, as can be seen in Fig. 10. As a result, the scaled vehicle can be used to emulate full-scale vehicle video streaming behavior. The authors of [72] used a similar scaled vehicle and reported a G2G latency of 241 ms. However, in [72], a local WiFi network with much lower latency compared to mobile networks is used. Still, the improvements proposed in this work reduce this delay by 74% to 63.7 ms. Thus, this reduction mainly results from the optimized streaming pipeline, rather than from differences in the communication infrastructure. The measured G2G delay of the scaled vehicle aligns closely with the sum of the individual delay contributions, validating the consistency of our measurements. Likewise, [73] reported a G2G delay below 120 ms for a full-scale vehicle using software encoding and a directly wired connection. Our optimized pipeline reduces this value to 71 ms (41% improvement), despite using a private mobile network and transmitting the data over the public internet. These latency gains improve the teleoperator's overall Quality of Experience (QoE), while simultaneously providing additional headroom. This headroom could, e.g., be used to compensate for the varying delay of the mobile network channel, whose characteristics fluctuate rapidly with channel conditions and user activity. Due to the even lower E2E delay, the scaled vehicle can be utilized to emulate full-scale remote driving. In the following, the delay distributions of components from camera and encoding (τ_1) to the downlink network delay (τ_4) are discussed. Latency distributions are visualized using violin plots to capture the full distribution including jitter effects.

Camera & encoding: Camera and encoding latency distributions for the full-scale and the scaled vehicle are displayed on the very left of Fig. 11. The cameras of the scaled and the full-scale vehicles offer different maximum resolution-frame-rate combinations, resulting in inherently different lower bounds on the capture delay. These hardware-imposed limits are reflected in the measured delay components and highlight the influence of camera capabilities on the achievable system-level latency. Differences in video preprocessing requirements, particularly if a format conversion is required, also influence the measured delay. In the full-scale vehicle, a dedicated camera interface enables direct hardware-accelerated video encoding without intermediate conversion, reducing the overall delay.

We observed a roughly linear relationship between the number of transmitted pixels per second and the delay of the camera source and encoder. Thus, although beneficial for operator perception, each increase in resolution and frame rate slightly increases the delay. This explains the higher encoder delay for the full-scale vehicle despite the more potent encoder hardware (cf. Table V), as it uses a higher resolution and frame rate. At the same time, higher video frame rates are reducing the exposure-induced delay at the camera itself. Overall, the reduced delay at the exposure outweighs the higher encoder delay, leading to a 10% reduction of τ_1 at the full-scale compared to the scaled vehicle.

Because hardware encoding is used, variations in the video bitrate do not significantly affect encoder delay, as detailed in Appendix A. However, the transmission delay τ_3 remains bitrate-dependent at a given channel capacity. Thus, the bitrate can be dynamically adjusted to the channel conditions without compromising real-time capabilities.

Uplink Video Transmission: For uplink and downlink transmission, we measured the delay across three network configurations: a public 4G mobile network, a private 5G NSA network with internet backhaul between the ATC and scaled test field (approximately 105 km apart), and a private 5G SA network peered directly on site. The respective channel conditions were chosen such that, in general, all networks could support the selected video bitrate and such remote driving would be feasible. In all cases, the mobile network uplink transmission is a dominant contributor to overall latency. However, there are significant differences between the tested network configurations. Public mobile networks are often primarily optimized for downlink throughput, leading to reduced uplink capacity under varying traffic conditions. As a result, the connection via the public mobile network yielded the highest delay with outliers over 100 ms, as shown in Fig. 11. The stochastic nature of other users' activity and the associated reduction in scheduled uplink resources are major contributors to uplink delay variations. A temporarily

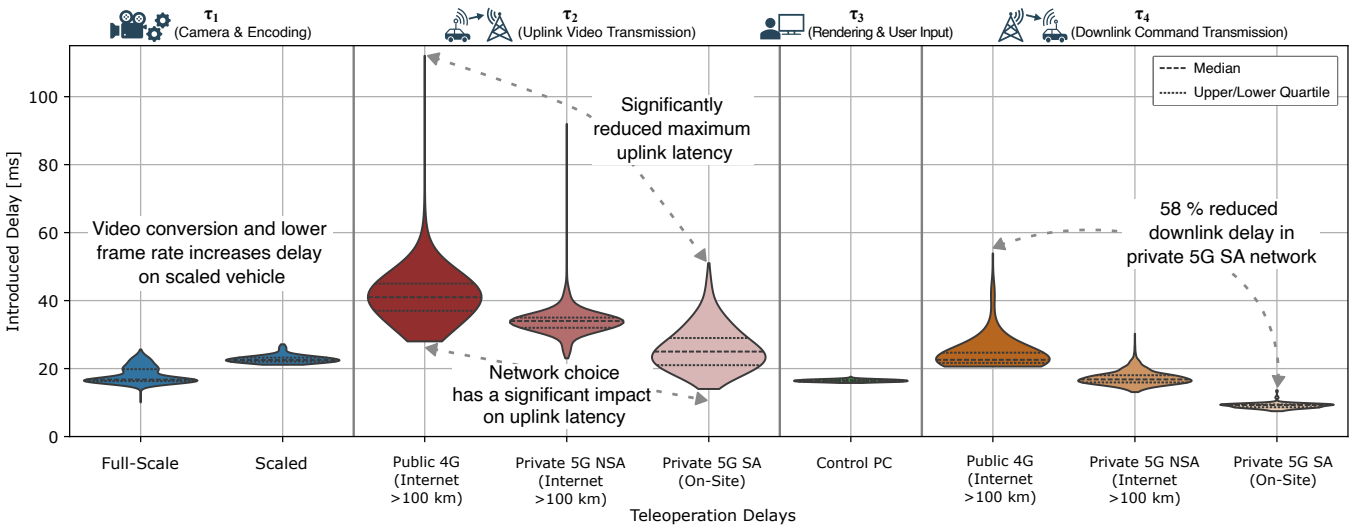


FIGURE 11. **Component-wise teleoperation E2E-delay distribution for the full-scale vehicle and the scaled vehicles in their respective test fields showing high dependence on the communication network (monitor and camera frame delays based on device parameters; outer 0.1% quantiles filtered).**

reduced uplink throughput below the video bitrate leads to delay spikes, which can, e.g., be prevented by slicing or using private networks. In our measurements, using a private mobile network reduced the median uplink delay by approximately 10 ms, and deploying a private 5G SA O-RAN system without an internet backhaul provided an additional improvement of roughly 10 ms. Although significantly reduced, variations in the transmission delay still persist, especially in the uplink. These can be explained by the central reactive uplink scheduling mechanisms of 4G and 5G systems [74]. Packet loss was negligible in the evaluated network configurations.

Rendering & User Input: Playback delays have been reduced to be comparatively small relative to other pipeline components. Rendering latency depends on display response time and frame rate, as well as the decoding pipeline. The authors of [5] reported a minimum rendering delay of 16.6 ms when using *rqt-image-view*. By eliminating ROS 2 for video playback, we can use hardware-accelerated decoding and rendering and avoid the serialization and deserialization overhead of ROS 2. This reduces the video playback delay to approximately 2.5 ms. An additional monitor delay of 14 ms is introduced, due to the used monitor type.

Downlink Command Transmission: Downlink delays exhibit similar trends across network types as in the uplink, with even greater relative improvements when transitioning to a private 5G SA network. Delay reductions of up to 58% are achieved, influenced by factors such as the subcarrier spacing (cf. Table II). The command transmission in the downlink is significantly faster than the uplink video transmission (cf. Fig. 11). The main cause of this smaller delay is not the lower data rate of control commands, but the absence of scheduling delays. Data in the downlink can be transmitted almost immediately without waiting for a transmission grant. Overall, communication delay accounts for a substantial portion of the total E2E latency, underscoring the importance of selecting an appropriate communication infrastructure.

Compared to the related work reporting the overall lowest latency parts [14], several delay components are further improved. While the use of 5G SA improves the network delays τ_2 and τ_4 by 28 ms, optimizations of the video pipeline decrease the processing delays by over 35 ms (cf. Table I). Thus, latency gains are distributed across multiple components, with the optimized streaming pipeline contributing even more than the communication infrastructure. While individual components, such as the video playback delay τ_3, may be further decreased by using even faster displays, the overall playback delay is already comparably small. Instead, the mobile network transmission is the dominant factor in the optimized pipeline. At the same time, current 5G network delays remain above the strict communication delay requirement of under 10 ms defined in [9]. Nevertheless, unlike most related work, the E2E delay of the proposed setup reaches the latency limits specified in [13] for both vehicle types.

C. PRECISE AND RELIABLE LOCALIZATION

To assess whether our localization approach meets the accuracy and reliability requirements for platooning, we evaluate the sensor fusion framework described in Sec. IV-B. The MoCap system in the scaled test field is used as ground truth, providing globally referenced position data at 120 Hz. To emulate GNSS-like measurements, the MoCap data is downsampled to update rates from 1 to 20 Hz and augmented with noise matching the real vehicle's GNSS accuracy. Wheel odometry and IMU measurements are provided at 50 Hz and 100 Hz, respectively. We recorded the sensor data while driving a scaled vehicle in the test field. This dataset is replayed, varying only the update rate of the simulated GNSS-like position updates. For each run, the estimated vehicle pose obtained from the EKF is compared against the ground truth from the MoCap system. Fig. 12 shows the lateral and longitudinal position errors of the EKF estimates and the raw emulated GNSS. In addition, 99% percentiles and the respective RMSEs are shown to capture both typical and worst-case performance.

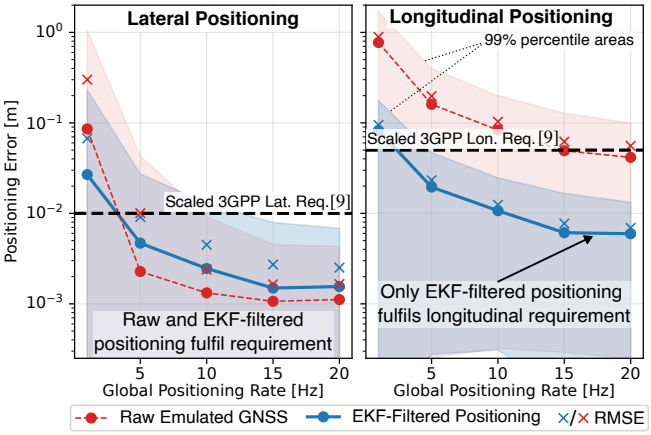


FIGURE 12. **EKF-based position estimation accuracy at different global position update rates compared to raw GNSS-like MoCap ground truth showing the EKF-approach reaching 3GPP requirements [9] at lower rates.**

The results demonstrate a clear dependence of localization accuracy on the global position update rate, with diminishing returns beyond approximately 15 Hz. Due to the Ackermann steering, lateral deviations are inherently limited. As a result, both the raw GNSS and the EKF-filtered estimates satisfy the stricter lateral requirement of below 0.1 m (1 cm scaled) defined in [9], provided a sufficient update rate. In contrast, longitudinal positioning proves more challenging. The raw GNSS approach cannot reliably fulfil the requirement at realistic update rates of $\leq$20 Hz, showing higher variability and worst-case errors. The EKF exhibits less variability and lower errors, remaining below the longitudinal requirement even at the 99% quantile, and consistently outperforms the raw GNSS approach across all evaluated frequencies. This performance gain is especially important at lower frequencies, where positioning errors are highest. The RMSE shows only small deviations from the median, indicating the absence of significant outliers and confirming the robustness of the estimation. Based on these results, a global position update rate of 20 Hz, combined with high-rate local sensing and EKF-based sensor fusion, is sufficient for platooning applications.

D. SHOWCASING COOPERATIVE INTERACTION AND COMPUTATIONAL OFFLOADING USING PLATOONING

Reliable and precise localization, and low-latency wireless communications are the foundation for cooperative platooning applications. As illustrated in Fig. 13, five scaled vehicles perform platooning while following a virtual leader's trajectory, with their positions visualized via laser projections. By transforming the leading full-scale vehicle into the scaled test field, platooning with mixed scaled, full-scale and virtual vehicles is performed. This experiment demonstrates the core capability of the *Digital Triplet*, synchronizing interactions between vehicles across different domains, which allows comparing platooning algorithms across domains.

Previously, a CACC-based platooning solution was implemented and evaluated in a highway scenario [2]. While computationally efficient, the CACC-based approach has limitations in urban scenarios with sharp turns compared to NMPC approaches with inherent planning capabilities (cf. Sec. IV).

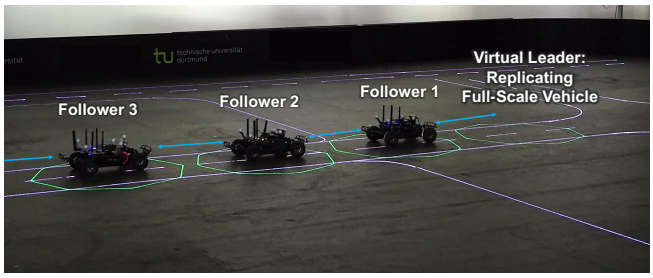
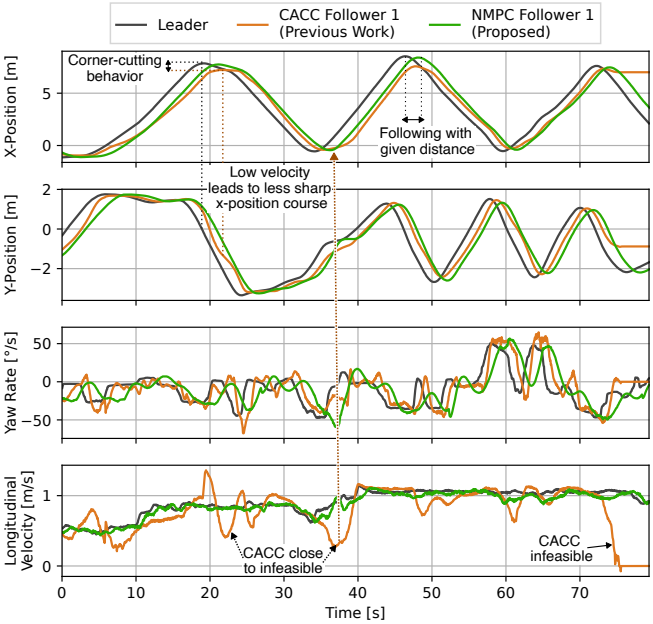
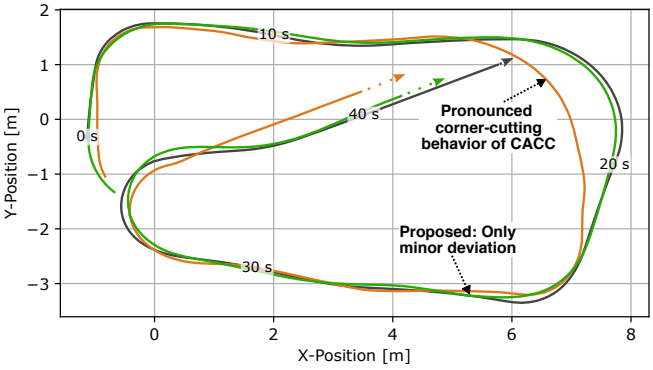


FIGURE 13. **Active NMPC platooning with five scaled vehicles following a virtual sixth vehicle in the scaled environment, driving the trajectory of the full-scale vehicle.**



(a) Reference trajectory with NMPC and CACC trajectories



(b) 2D plot of part of the reference trajectory and the NMPC and CACC path

FIGURE 14. **Evaluation of CACC and NMPC platooning along a reference trajectory for one follower, showing more precise operation of the NMPC.**

A performance comparison between the CACC-based platooning approach from [45] and the proposed NMPC controller is illustrated in Fig. 14. Within the evaluation, both algorithms are configured to follow the same reference trajectory recorded by a teleoperated leader vehicle. The plots in Fig. 14a show the evolution of the position, yaw rate, and ve-

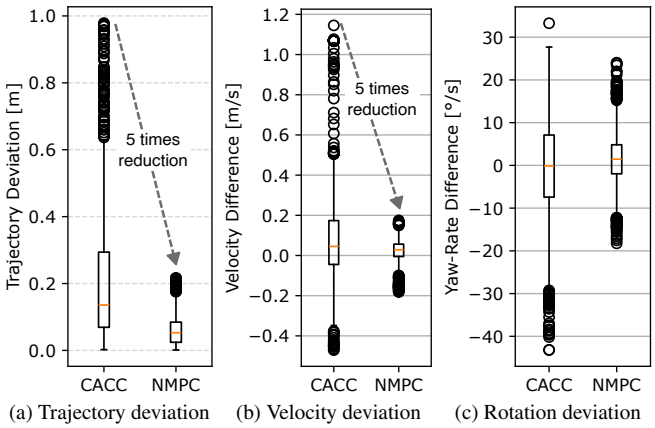


FIGURE 15. Comparison of CACC and NMPC performance following the same reference trajectory (Points outside the 1.5 times interquartile range are shown as outliers).

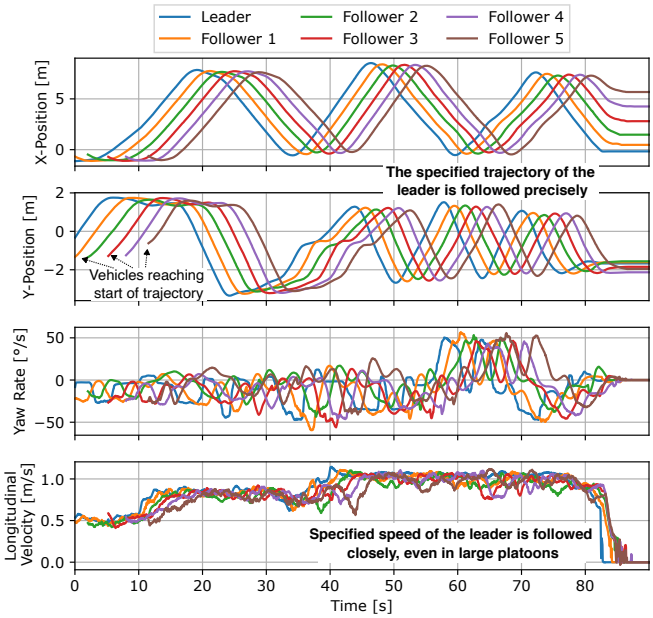


FIGURE 16. Visualization of a NMPC platoon of five vehicles following a reference trajectory confirming precise trajectory tracking in larger platoons.

locity of the first follower vehicle over time, whereas Fig. 14b depicts the corresponding two-dimensional trajectories.

The results highlight fundamental behavioral differences between the two control strategies. The CACC follower is generally capable of maintaining the trajectory under mild curvature, but its performance degrades significantly at sharp turns. Around 15–20 s, the CACC begins to exhibit pronounced corner-cutting behavior and reacts to high-curvature sections by reducing the velocity. This behavior encourages further corner-cutting at the next turn. Near the final turn, the CACC controller reaches an infeasible state and halts, failing to complete the trajectory due to the heading difference to the leader exceeding the limit of $\pm 90°$. In contrast, as visible in both the time-domain and the 2D trajectory plots, the NMPC follower closely tracks the reference path with minimal lateral deviation and a matching velocity, even during sharp turns.

These findings are further quantified in Fig. 15, which compares the deviations of both controllers from the reference trajectory in terms of lateral distance, velocity, and yaw rate using box plots including outliers. The results confirm that NMPC achieves up to a fivefold reduction in trajectory deviation compared to CACC, while maintaining significantly better adaptation to the velocity and orientation of the leader. This improvement stems from the model-based planning, as described in Sec. IV, which jointly minimizes path tracking error and control effort. The approach also shows tolerance to mobile network-induced effects as delay and jitter.

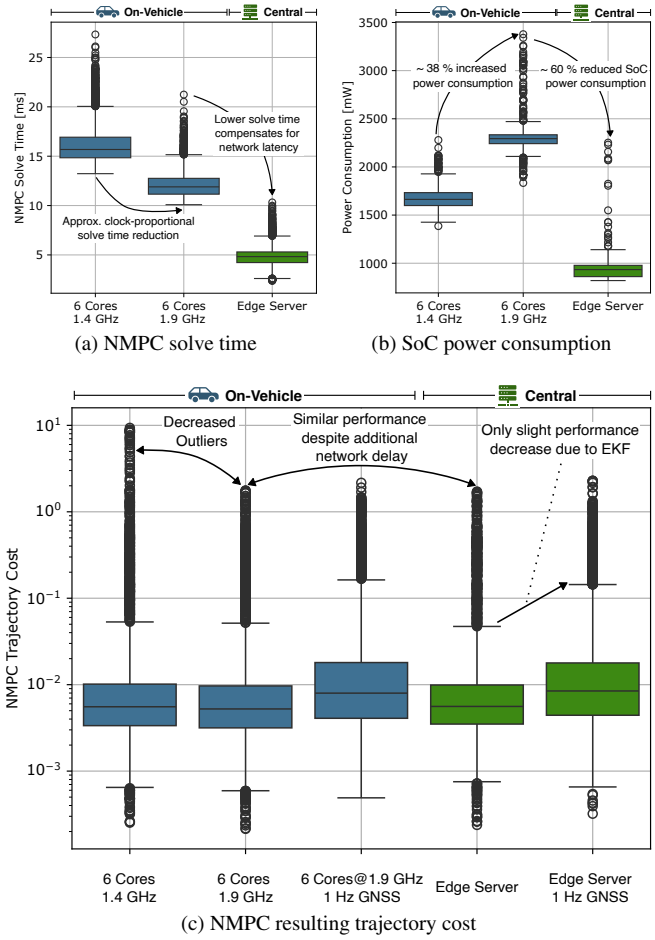
1) Experimentally Observed Platooning Stability

The scalability and stability of the proposed approach are illustrated in Fig. 16, depicting a platoon comprising the leader vehicle and five NMPC followers. All vehicles closely follow the leader trajectory throughout the scenario. Six slightly time-shifted curves can be observed, representing the sequential arrival of each follower at the same spatial positions and velocities as the leader, offset by the desired inter-vehicle spacing. The observed stable tracking performance along the vehicle chain indicates that the leader's velocity and steering angle fluctuations are not amplified downstream. Consequently, the results demonstrate that the NMPC approach remains robust and stable even for extended platoons operating under realistic communication and actuation constraints, confirming the analytical evaluation in Appendix B.

2) Centralized Control in Edge-Cloud Deployment

The performance of the NMPC approach was further analyzed by comparing its execution on the vehicle's onboard computing unit with limited computational resources to execution on a high-performance edge server connected via a 5G SA network. The results regarding the computation time, SoC power consumption and the internal NMPC trajectory cost J under different deployment configurations are shown in Fig. 17. The box plots are aggregated over all vehicles.

Fig. 17a shows the NMPC solve time for three setups: onboard execution at nominal and overclocked frequencies, and remote execution on the edge server. The onboard solver requires approximately 16 ms per optimization cycle at the default clock frequency, decreasing proportionally with overclocking to around 12 ms. Thus, NMPC-induced latency can be improved at the cost of an approximately 38% increased power consumption due to overclocking, as shown in Fig. 17b. When offloaded to the edge server, the solve time is further reduced to around 5 ms, while the SoC power consumption is decreased by 60% compared to the overclocked configuration. If also considering the additional downlink delay τ_{DL}, the median effective solve time of the edge cloud controller $t'_{c,eff}$ is 14 ms with a lower standard deviation than the local setups. At the same time, the states used in the edge controller are less delayed than in the local deployment (cf. Appendix C). As a result, a comparable performance to the overclocked local deployment is reached, as shown in the resulting NMPC trajectory cost in Fig. 17c. While



(a) NMPC solve time (b) SoC power consumption

(c) NMPC resulting trajectory cost

FIGURE 17. Performance comparison of edge-cloud and on-vehicle execution of the NMPC controller showing comparable performance. Further, the implementation shows robustness to a reduced clock speed and a low global positioning update rate.

all discussed configurations yield a similar median cost, the faster execution during overclocking and in the edge server reduce outliers, improving the worst-case performance.

As the vehicle states are the input to the platooning controller, deferred position updates can also affect performance like communication delays and high solve times. To quantify this effect, we reduced the global localization update rate to 1 Hz, which would result in localization errors of up to 2 m at maximum velocity. However, location state estimates by the EKF provide needed updates for the NMPC, which is continuously adapting the planned trajectory to possible induced inaccuracies. While the median trajectory cost and its interquartile range increase, this configuration shows robustness against lower global update rates. Thus, the EKF can provide sufficiently precise state estimations for over 1 s.

The discussed results underline the critical influence of fast communication, computation and localization on platooning performance. By comparing distributed and centralized NMPC deployments, it becomes evident that a careful co-design of communication and control, as discussed in [74], is essential. The ability to balance computational load between the vehicle and edge cloud resources enables adaptive optimization based on network conditions and mission requirements. The study shows that edge execution is feasible using 5G SA, even for latency-critical platooning. In the future, further repeated trial studies could refine the results gained in this case study.

However, the experiments demonstrate that the *Digital Triplet* enables synchronized interaction across real, scaled, and simulated domains, which cannot be conducted with conventional testbeds that rely on a single abstraction level.

VI. CONCLUSION AND OUTLOOK

In this paper, we presented our *Digital Triplet* concept for the evaluation of future 6G C-ITS concepts. By combining the three interconnected layers, real-world, scaled replica, and simulation, the *Digital Triplet* enables realistic, reproducible, and scalable testing across diverse deployment scenarios.

Through multiple case studies, we demonstrated the versatility and effectiveness of the proposed framework. For resilient low-latency remote driving, the integration of the RISE concept for adaptive video streaming improves coverage and perceptual quality, while video pipeline adaptations showed significant reductions in E2E delay compared to state-of-the-art solutions. Measurements confirmed the advantages of 5G standalone private networks over public 4G and 5G NSA. Furthermore, by using the *Digital Triplet's* ability to apply metrics of the virtual domain to the real world, we evaluated the potential of deploying future large-scale RISs to extend the coverage of FR2-based remote driving. This combines controlled connectivity with realistic mobility in scaled or full-scale environments and can be refined in future work to assess hardware-specific effects more precisely.

The scaled part of our implementation of the *Digital Triplet* consists of *RoboRacer* vehicles, whose hardware and software were extended to depict full-scale vehicles even more accurately. This includes adding mobile network connectivity directly, simplifying the system architecture and reducing latency compared to previous work. For the scaled vehicles, we demonstrated reliable localization even at reduced simulated GNSS update rates. The feasibility of NMPC-based platooning was demonstrated for on-vehicle and edge-cloud deployment, showing that real-time control can be achieved with minimal latency while reducing computational load through task offloading. By integrating simulated, scaled, and real vehicles within one continuum, the *Digital Triplet* proves to be a powerful enabler for cost-efficient and comprehensive evaluation of C-ITS applications.

In the future, we plan to further leverage the parameter space of *O-RAN* base stations to evaluate and improve network capabilities. By extending a previously developed predictive scheduling approach [75], proactive uplink resource allocation can be studied for remote driving. This has the potential to significantly reduce the total delay of remote driving, as the uplink transmission is a major contributor to the overall E2E latency.

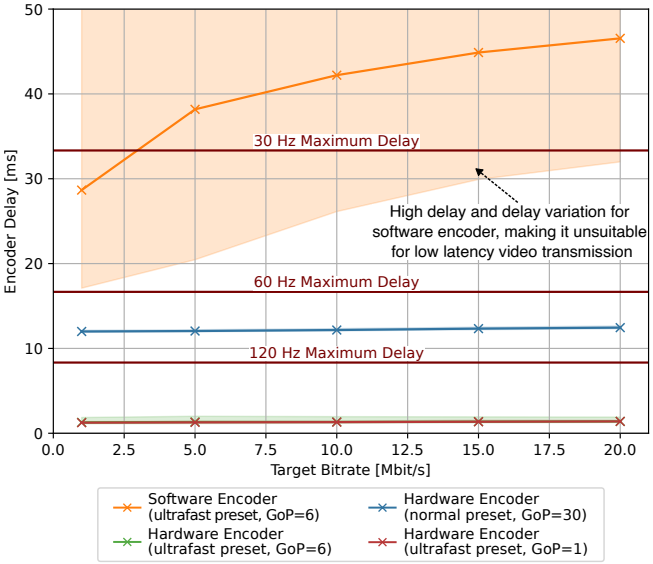


FIGURE 18. Encoder delays and 95% percentiles for hardware and software encoders on the *Jetson AGX Orin* showing the advantage of hardware encoders yielding a lower and bitrate-independent delay.

APPENDIX A
VIDEO STREAMING QUALITY AND DELAY OPTIMIZATION

This appendix provides further insights into video coder selection concerning video quality and delay optimization. Remote driving has strict latency constraints necessitating careful optimization of the video pipeline. First, we restricted the encoders to intra- and predictive frames, as bidirectional frames introduce additional latency due to backward predictions. Thus, the encoder and decoder delay is reduced. Optimizing the video pipeline also involves deciding between software and hardware encoders.

While software encoders usually offer more features to improve compression, this comes with the cost of a high computational load. On the deployed systems, this results in excessive delay, necessitating the disabling of some of these features and optimizing for faster encoding (i.e., *zerolatency*, *ultrafast*). Even with these optimizations, the software encoder latency is significantly higher than for hardware encoders, and the delay fluctuates strongly, as shown in Fig. 18. Additionally, the encoder delay increases with the video bitrate, such that already at 5 Mbit/s, even a median frame rate of 30 Hz cannot be sustained. Using hardware encoders, however, yields a consistently lower and bitrate-independent encoding delay, enabling higher frame rates.

The hardware encoder's default configuration already maintains a frame rate of over 60 Hz with only marginal delay fluctuations. Using GPU-compatible data types reduces system-to-GPU memory transfers. At the same time, GPU video encoding is fast and is not significantly affected by other computational loads. By switching to the hardware encoder's *ultrafast* preset, even lower latency is achieved, enabling frame rates over 120 Hz. Consequently, the highest available frame rate of the vehicle camera can be used, also reducing the frame rate-induced delay.

TABLE V. Video Streaming Parameterization for Latency Evaluation

Parameter	Full-Scale	Scaled 1/10
Camera Interface	GMSL2	USB 3.0
Resolution	$960 \times 600p$	$480 \times 360p$
Frame Rate	120 Hz	60 Hz
Encoder Hardware	Jetson AGX Orin	Jetson Xavier NX
Codec	Ampere h265 NVENC	Volta h265 NVENC
Group of Pictures	6	6
Bitrate	10 Mbit/s	10 Mbit/s
Streaming Protocol	Real-time Transport Protocol	
Transport Protocol	User Datagram Protocol	
UDP Buffer	12.5 kB	
Decoder Hardware	NVIDIA GeForce RTX 4080 Super	
Decoder	h265 NVDEC	
Display	LG 55NANO869PA	
Display Setup	$1920 \times 1080p$, 120 Hz	

For the optimized hardware encoder, we utilize a Group of Pictures (GoP) of six compared to 30 in the default configuration. Full parameterization of the video streaming pipeline for the scaled and the full-scale vehicle is listed in Table V. Although increasing the number of predictive frames can enhance compression efficiency, fewer predictive frames reduce worst-case recovery times after frame loss. While the recovery time for the default configuration at 120 Hz is around 250 ms, exceeding the maximum tolerated latency for remote driving, at a GoP of 6, it is reduced to 50 ms. In the case of only I-frames, the recovery time is only one frame. While this also reduces jitter slightly, no improvement in median encoder delay occurs, consistent with G2G measurements. At the same time, the required bitrate to achieve the same quality is higher, as shown in Fig. 19.

When comparing video quality, the Video Multi-Method Assessment Fusion (VMAF) score combines multiple conventional scores, such as the Peak Signal-to-Noise Ratio (PSNR), into a single score that correlates with human perception [76]. In Fig. 19, we compare the VMAF score of encoders on a video of a front-facing camera in a car driving in a campus scenario also used in [65]. This ensures a realistic video feed and fair comparison between encoders. Due to the software encoder's required latency optimizations, it yields a lower VMAF score than the hardware encoders across all bitrates. Thus, the proposed use of hardware encoders does improve latency and video quality at the same time. The hardware encoder with the normal preset performs best, closely followed by the ultrafast preset. These differences become less significant with higher bitrates. As the ultrafast preset offers the highest refresh rate while only slightly worse in quality, we choose to use it throughout this paper.

For remote driving, the most important metric is not the highest possible quality, e.g., VMAF score, but the operator's ability to react correctly and in time based on the video stream. However, VMAF is widely used to assess video quality for comparison between encoders and correlates with operator perception [77]. Thus, selecting the most suitable encoder is viable based on quality metrics such as VMAF.

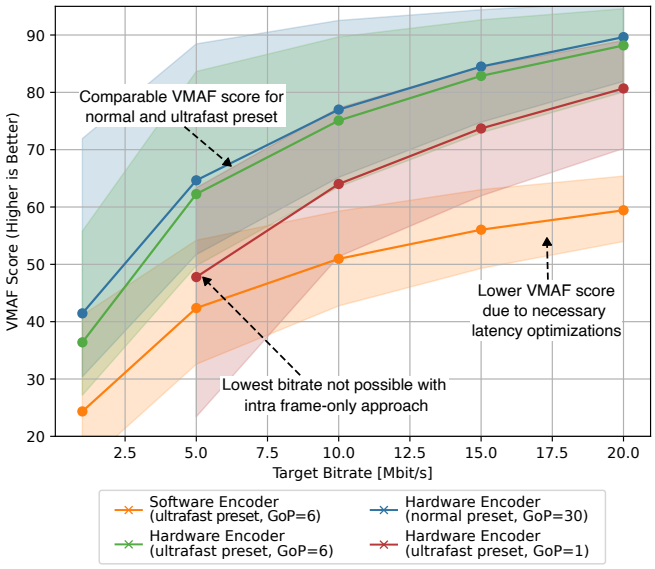


FIGURE 19. Evaluation of VMAF score for different encoders and configurations on the *Jetson AGX Orin* underlining the advantage of utilizing hardware encoders. The 95% percentile areas show that the relationship between image quality and the encoder's bitrate follows a distribution, rather than a clear mapping.

To judge whether the operator can use the video feed to drive safely, the content complexity must also be considered. This complexity is dependent on the exact driving scenario.

APPENDIX B
NONLINEAR MODEL PREDICTIVE CONTROL APPROACH

The proposed NMPC scheme for combined longitudinal and lateral platooning solves an Optimal Control Problem (OCP) over the finite horizon N with the following internal states and control inputs:

$$x_k = \begin{bmatrix} p_x & p_y & p_\theta & v & \delta \end{bmatrix}_k^T \in \mathbb{X}, \quad u_k = \begin{bmatrix} a & \dot{\delta} \end{bmatrix}_k^T \in \mathbb{U}, \quad (12)$$

where x_k is the state vector at time step k comprising the 2D pose $\boldsymbol{p}_k$ in Cartesian coordinates with x, y components and heading θ, velocity v_k as well as steering angle δ_k. The control input vector u_k is composed of the acceleration $\dot{v}_k = a_k$ and steering angle rate $\dot{\delta}_k$. Using these states, the objective function of the OCP consists of three components: trajectory adherence of the leading reference trajectory $\mathbf{r}$, an input penalization and the terminal reference state r_N condition, i.e., the desired position and velocity at the end of the horizon. The reference trajectory $\mathbf{r} = [r_0, r_1, \ldots, r_N]$ is generated based on the leading vehicle's trajectory, a desired inter-vehicle distance η, and the current velocity. The Lagrange term l_k of the cost function at time step k is defined as

$$l_k(x_k, u_k) = (x_k - r_k)^T \mathbf{Q}(x_k - r_k) + u_k^T \mathbf{R} u_k. \quad (13)$$

This linear least-squares cost function is minimized over the finite horizon N to achieve an optimal trajectory via the input vector $u_k, k \in [0, N-1]$. The Mayer term or terminal cost M is calculated as

$$M(x_N) = (x_N - r_N)^T \mathbf{S}(x_N - r_N). \quad (14)$$

Hence, the full cost is given by

$$J(\mathbf{x}, \mathbf{u}) = \sum_{k=0}^{N-1} l_k(x_k, u_k) + M(x_N). \quad (15)$$

Here, $\mathbf{Q} = \mathrm{diag}(0.5, 0.5, 0, 0, 0)$, $\mathbf{R} = \mathrm{diag}(0.2, 0.075)$, and $\mathbf{S} = \mathrm{diag}(500, 500, 0, 100, 0)$ are the empirically determined positive semi-definite weight matrices for the state stage error, the input penalty, and the terminal condition error. In $\mathbf{Q}$, only location deviations are penalized, as velocity and heading errors are already penalized through their induced deviation along the N points of the trajectory. The weights are selected, such that $\|\mathbf{Q}\| > \|\mathbf{R}\|$, as $\mathbf{R}$ only concerns driving behavior. While individual weights in the weight matrices need to be balanced relative to each other, weights in $\mathbf{S}$ need to be larger than N times the weights in $\mathbf{R}$, as reaching the terminal pose and velocity is the most important. Finally, optimal control actions $\mathbf{u}^\star$ and system states $\mathbf{x}^\star$ are derived via minimization of J subject to system dynamics and constraints given in Table III as follows:

$$(\mathbf{x}^\star, \mathbf{u}^\star) = \underset{\mathbf{x},\mathbf{u}}{\arg\min}\, J(\mathbf{x}, \mathbf{u}) \quad (16a)$$

$$\text{s. t.} \quad x_{k+1} = f(x_k, u_k) \quad \forall k = 0, \ldots, N-1 \quad (16b)$$

$$x_0 = \bar{x}_0 \quad (16c)$$

$$0 \le v_k \le v_{\max} \quad \forall k = 0, \ldots, N \quad (16d)$$

$$-\delta_{\max} \le \delta_k \le \delta_{\max} \quad \forall k = 0, \ldots, N \quad (16e)$$

$$a_{\min} \le a_k \le a_{\max} \quad \forall k = 0, \ldots, N-1 \quad (16f)$$

$$-\dot{\delta}_{\max} \le \dot{\delta}_k \le \dot{\delta}_{\max} \quad \forall k = 0, \ldots, N-1. \quad (16g)$$

During evaluation, $\mathbf{x}$ and $\mathbf{u}$ are limited as follows to prevent slip and significant dynamic differences between vehicles and domains (cf. Sec. III-C2): $v_{\max} = 2\,\mathrm{m/s}$, $a_{\min} = -3\,\mathrm{m/s^2}$, $a_{\max} = 2.5\,\mathrm{m/s^2}$, $\delta_{\max} = 0.35\,\mathrm{rad}$, and $\dot{\delta}_{\max} = 2.5\,\mathrm{rad/s}$.

The nonlinear kinematic bicycle model is used as the system model. With sufficiently limited lateral acceleration, this simplified model has proven to be accurate for most applications in autonomous vehicles [52]. The transfer function of the system model in a global frame can be written as:

$$f(x, u) = x + \begin{bmatrix} v \cdot \cos(p_\theta) \\ v \cdot \sin(p_\theta) \\ v/L_w \cdot \tan(\delta) \\ a \\ \dot{\delta} \end{bmatrix} \cdot \Delta t \quad (17)$$

with known wheelbase L_w, and the discretization time step Δt. The nonlinear system model is linearized using a fourth-order Runge-Kutta (RK4) integration scheme. This approach approximates the system dynamics over the discretization interval. The transfer function is initialized with $x_0 = \bar{x}_0$, where $\bar{x}_0$ is the observed system state. By using the *acados* framework [78], which offers fast solvers for NMPC, the execution of the algorithm is accelerated. This allows the NMPC to be deployed directly on the scaled vehicles as well as on powerful edge servers.

APPENDIX C
STRING STABILITY AND DELAY CONSIDERATIONS

This appendix provides the main arguments for stability of the proposed NMPC approach under the stated assumptions. There exist several works with similar NMPC approaches, proving recursive feasibility and closed-loop stability definitions [79–82]. Therefore, here, we focus on string stability.

A. STRING STABILITY ANALYSIS

We employ a constant time headway inter-vehicle distance policy

$$d_i(t) = \delta_0 + \Delta_t \cdot v_i(t) \tag{18}$$

for vehicle i with velocity $v_i(t)$ as the target for the platoon, where $\Delta_t > 0$ is the time headway.

Since all vehicles aim to follow the same reference trajectory and only the longitudinal target position along the trajectory depends on previous vehicles, we analyze the platoon dynamics in one-dimensional space. We assume that the platoon is initially in equilibrium, i.e., $e_i(0) = 0$ and $v_i(0) = v_0$ for all vehicles i. Linearizing the longitudinal vehicle dynamics around constant velocity v_0 yields the following model:

$$\dot{s}_i(t) = v_i(t), \qquad \dot{v}_i(t) = a_i(t). \tag{19}$$

Let $s_i(t)$ be the current longitudinal position of vehicle i along the trajectory. The spacing error $e_i(t)$ is defined as

$$e_i(t) = s_{i-1}(t) - s_i(t) - d_i(t). \tag{20}$$

The platoon is said to be string stable in the L_2 sense if there exists $0 < p < 1$ such that $\|e_i\|_2 \leq p \cdot \|e_{i-1}\|_2$ for all i. For linear systems this is equivalent to

$$\|G\|_\infty = \sup_{\omega \in \mathbb{R}_{\geq 0}} |G(j\omega)| \leq 1, \tag{21}$$

where $G(s) = E_i(s)/E_{i-1}(s)$ is the transfer function from predecessor spacing error $E_{i-1}(s)$ to the follower spacing error $E_i(s)$.

Near the equilibrium point, the nonlinear dynamics of the NMPC can be locally linearized. In this region, the NMPC with cost matrices $\mathbf{Q}$ and $\mathbf{S}$ behaves like a linear quadratic regulator (LQR). For the used double-integrator vehicle model, the resulting optimal feedback law reduces to a proportional-derivative (PD) structure of the form

$$a_i(t) = k_p \cdot e_i(t) + k_d \cdot \dot{e}_i(t), \tag{22}$$

where $k_p > 0$ and $k_d > 0$ are effective gains determined by the weight matrices and prediction horizon. Evaluating the transfer function $G(s)$ yields the conservative string stability condition

$$\Delta_t \geq \sqrt{2/k_p}. \tag{23}$$

Hence, the time headway Δ_t must be chosen sufficiently large relative to the effective proportional gain k_p induced by the NMPC. Under this condition, disturbances do not amplify along the platoon, and the closed-loop system is string stable. In a simulation of a platoon with given $e(t)$ and $\dot{e}(t) = 0$, we identified $k_p \approx a(t)/e(t)$ of the NMPC in our setup to be 3.88 1/s^2 with $\Delta_t = 0.8$ s, fulfilling (23).

B. INFLUENCE OF DELAYS ON FEASIBILITY

In our proposed setup, we use 4G and 5G mobile networks for inter-vehicle status exchanges. In contrast to other systems like WiFi, which in general use contention-based medium access, mobile networks employ centralized scheduling at the base station, preventing collisions. Further, mobile networks mitigate packet loss effects by retransmissions on lower layers. Still, they introduce communication delays τ_n in addition to the NMPC controller delay τ_c. However, the introduced delays are significantly smaller than the set time delay Δ_t. The NMPC controller delay τ_c is less than 16 ms and the E2E network delay τ_n in the scaled testbed is below 55 ms, even for uplink video streaming. Thus, these delays may be approximated as linear transfer functions $F_c(s)$ and $F_{n,E2E}(s)$. Due to the network delay, the observed error $\hat{e}_i(t)$ is a time-delayed version of the true error $e_i(t)$, i.e.,

$$\hat{e}_i(t) = \hat{s}_{i-1}(t) - s_i(t) - d_i(t), \tag{24}$$

where $\hat{s}_{i-1}(t)$ denotes the delayed observation of the preceding vehicle in case of the distributed configuration. Packet losses or jitter, e.g., due to retransmissions, result in delayed state updates, and thus can be modeled as additional delay in the observation. Padé or first-order lag approximations of $F_{n,E2E}(s)$ result in the transfer function of the delayed spacing errors becoming $\hat{G}(s) = F_n(s)G(s)$ with $|F_n(jw)| \leq 1$, thus not negatively affecting the string stability condition (21). Further, due to velocity and acceleration restrictions, $v_{\max}$ and $a_{\max}$, the additional latency-induced spacing error $\delta_e(t) = |e_i(t) - \hat{e}_i(t)|$ is bounded by

$$\delta_e(t) \leq \tau_n v_i + \tfrac{1}{2} a_{\max} {\tau_n}^2, \quad \delta_e(t) \leq \tau_n v_{\max}. \tag{25}$$

Thus, due to the given time headway inter-vehicle distance, these deviations can be absorbed. In the edge deployment, all states are delayed, but only by the uplink delay component τ_{UL}. This results in the observed error $\hat{e}_i(t) = e_i(t - \tau_{UL})$ leading to a similar expression as in (25) with τ_{UL}.

In case of a significant controller delay τ_c, its corresponding transfer function $F_c(s)$ is in the numerator and the denominator of $G(s)$. This creates more complex conditions for string stability. If using a first-order lag to model the controller delay as the only delay, the condition $(1+\Delta_t k_d)^2 - 2\tau_c(k_d + \Delta_t k_p) \geq 0$ needs to be met in addition to (23) for a conservative stability check. For the given setup, this additional condition is fulfilled for every $k_d \geq 0$. In case of the edge deployment, further downlink delay τ_{DL} is added to the controller delay τ_c', resulting in an effective controller delay $\tau_{c,eff}' = \tau_c' + \tau_{DL}$.

The above analysis establishes local string stability of the linearized closed-loop system under nominal assumptions. In practice, additional nonlinear effects such as further delays, actuator limits, and heterogeneous vehicle dynamics may affect disturbance propagation. The experimental results with a six-vehicle platoon indicate that the proposed approach remains stable under these practical conditions.

ACKNOWLEDGMENT

The authors would like to thank the Chair of Material Handling and Warehousing (FLW) at TU Dortmund University for providing access to their indoor test field. They also gratefully acknowledge the contributions of the project group *SWIFTdrive*, whose prior work laid a foundation for parts of this study.

REFERENCES

[1] H. Schippers, C. Schüler, B. Sliwa, and C. Wietfeld, "System modeling and performance evaluation of predictive QoS for future tele-operated driving," in *Proc. IEEE International Systems Conference (SysCon)*, Apr. 2022.

[2] C. Krieger, H. Teper, J. Freytag, I. F. Priyanta, P. Schulte, M. Roidl, J.-J. Chen, and C. Wietfeld, "Integration of scaled real-world testbeds with digital twins for future AI-enabled 6G networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2023.

[3] A. Mokhtarian, J. Xu, P. Scheffe, M. Kloock, S. Schäfer, H. Bang *et al.*, "A survey on small-scale testbeds for connected and automated vehicles and robot swarms: A guide for creating a new testbed [survey]," *IEEE Robotics & Automation Magazine*, vol. 32, no. 3, pp. 146–163, Sep. 2025.

[4] H. Lee, J. Park, C. Koo, J.-C. Kim, and Y. Eun, "Cyclops: Open platform for scale truck platooning," in *Proc. International Conference on Robotics and Automation (ICRA)*, May 2022.

[5] T. Kerbl, D. Brecht, N. Gehrke, N. Karunainayagam, N. Krauss, F. Pfab *et al.*, "TUM Teleoperation: Open source software for remote driving and assistance of automated vehicles," in *2025 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2025, pp. 2593–2600.

[6] N. Buckman, A. Hansen, S. Karaman, and D. Rus, "Evaluating autonomous urban perception and planning in a 1/10th scale minicity," *Sensors*, vol. 22, no. 18, Sep. 2022.

[7] T. Linget, "C-V2X use cases volume II: Examples and service level requirements," 5GAA Automotive Association, Tech. Rep., Oct. 2020.

[8] ——, "Tele-operated driving (ToD): System requirements analysis and architecture," 5GAA Automotive Association, Tech. Rep., Sep. 2021.

[9] 3rd Generation Partnership Project (3GPP), "5G; Service requirements for enhanced V2X scenarios," 3GPP, Technical Specification (TS) 22.186, Oct. 2025, version 19.0.0.

[10] 5G-MOBIX, "5G for cooperative & connected automated MOBIility on X-border corridors," Tech. Rep.

[11] A. Schimpe, I. G. Vazquez, A. Pfadler, K. T. Floess, J. Löfhede, H. Zhao *et al.*, "Deliverable D2.1 Test Case Definition and Trial site Description Part 1," Fifth Generation Cross-Border Control, Tech. Rep., 2020.

[12] T. Floess, J. Löfhede, M. Muehleisen, A. Pfadler, A. Schimpe, R. Sedar *et al.*, "Deliverable D2.2 test case definition and test site description for second round tests and trials," Fifth Generation Cross-Border Control, Tech. Rep., 2020.

[13] Bundesministerium der Justiz und für Verbraucherschutz, "Verordnung über Ausnahmen von straßenverkehrsrechtlichen Vorschriften für ferngelenkte Kraftfahrzeuge (Straßenverkehr-Fernlenk-Verordnung – StVFernLV)," Published in Bundesgesetzblatt I, No. 176. Available: https://www.recht.bund.de/bgbl/1/2025/176/VO.html, Jul. 2025.

[14] J.-M. Georg, J. Feiler, S. Hoffmann, and F. Diermeyer, "Sensor and actuator latency during teleoperation of automated vehicles," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, Las Vegas, NV, USA, Oct. 2020.

[15] A. Belogolovy, D. Dasalukunte, R. Dorrance, E. Stupachenko, and X. Zhang, "Low latency communication over commercially available LTE and remote driving," arXiv:2209.09794, Sep. 2022.

[16] M. Testouri, G. Elghazaly, F. Hawlader, and R. Frank, "5G-enabled teleoperated driving: An experimental evaluation," in *Proc. IEEE Int. Conf. on Models and Technologies for Intelligent Transp. Systems (MT-ITS)*, 2025.

[17] G. Kakkavas, K. N. Nyarko, C. Lahoud, D. Kühnert, P. Küffner, M. Gabriel *et al.*, "Teleoperated support for remote driving over 5G mobile communications," in *Proc. IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, Sep. 2022.

[18] G. Kakkavas, M. Diamanti, K. Nseboah Nyarko, M. Gabriel, V. Karyotis, K. Mößner, and S. Papavassiliou, "Realistic field trial evaluation of a teleoperated support service for remote driving over 5G," in *Proc. IEEE Conf. on Standards for Commun. and Networking (CSCN)*, Feb. 2022.

[19] R. Bennett, R. Kapp, T. R. Botha, and S. Els, "Influence of wireless communication transport latencies and dropped packages on vehicle stability with an offsite steering controller," *IET Intell. Transport Syst.*, Mar. 2020.

[20] S. Neumeier, E. A. Walelgne, V. Bajpai, J. Ott, and C. Facchi, "Measuring the feasibility of teleoperated driving in mobile networks," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, Jun. 2019.

[21] S. Neumeier, P. Wintersberger, A.-K. Frison, A. Becher, C. Facchi, and A. Riener, "Teleoperation: The holy grail to solve problems of automated driving? Sure, but latency matters," in *Proc. ACM International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Utrecht, Netherlands, Sep. 2019.

[22] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "High latency unmanned ground vehicle teleoperation enhancement by presentation of estimated future through video transformation," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 2, p. 48, Oct. 2022.

[23] C. Schüler, T. Gebauer, M. Patchou, and C. Wietfeld, "QoE evaluation of real-time remote operationwith network constraints in a system-of-systems," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2022.

[24] S. Avanzini, "Deep dive: How to break the congestion barrier - Achieving low latency with high throughput for safe teledriving," Vay, accessed: Apr. 3, 2026. [Online]. Available: https://vay.io/deep-dive-how-to-break-the-congestion-barrier-achieving-low-latency-with-high-throughput-for-safe-teledriving

[25] "Teleoperation - The future of machine control across industries," RoboAuto, Brno, CZECHIA, Tech. Rep., Oct. 2024, accessed: Nov. 16, 2025. [Online]. Available: https://roboauto.tech

[26] Soliton Systems Europe N.V. , "Ultra low latency video streaming - innovating with ultra low latency live streaming," accessed: Dec. 18, 2025. [Online]. Available: https://blog.solitonsystems.com/blog/innovating-with-ultra-low-latency-live-streaming

[27] F. J. Jiang, J. Mårtensson, and K. H. Johansson, "Safe teleoperation of connected and automated vehicles," in *Cyber-Physical-Human Systems: Fundamentals and Applications*. John Wiley & Sons, Ltd, Jun. 2023, ch. 10, pp. 251–272.

[28] I. Otani, Y. Yaguchi, K. Nakamura, and K. Naruse, "Quantitative Evaluation of Streaming Image Quality for Robot Teleoperations," *Artificial Life and Robotics*, vol. 24, no. 2, pp. 230–238, Jun. 2019.

[29] M. Lucas-Estañ, B. Coll-Perales, M. I. Khan, S. S. Avedisov, O. Altintas, J. Gozalvez, and M. Sepulcre, "Support of teleoperated driving with 5G networks," in *Proc. IEEE Veh. Technol. Conf. (VTC2023-Fall)*, Oct. 2023.

[30] M. Lucas-Estañ, B. Coll-Perales, M. I. Khan, J. Gozalvez, S. S. Avedisov, O. Altintas, and M. Sepulcre, "5G network architecture and configuration choices to support teleoperated driving at scale," in *Proc. IEEE Vehicular Technology Conference (VTC2024-Fall)*, Oct. 2024.

[31] J. Žádník, M. Mäkitalo, J. Vanne, and P. Jääskeläinen, "Image and video coding techniques for ultra-low latency," *ACM Computing Surveys*, vol. 54, no. 11s, Jan. 2022.

[32] M. Hofbauer, C. B. Kuhn, M. Khlifi, G. Petrovic, and E. Steinbach, "Traffic-aware multi-view video stream adaptation for teleoperated driving," in *Proc. IEEE Vehicular Technology Conference (VTC-Spring)*, Helsinki, Finland, Jun. 2022.

[33] S. Neumeier, V. Bajpai, M. Neumeier, C. Facchi, and J. Ott, "Data rate reduction for video streams in teleoperated driving," *IEEE Trans. on Intelligent Transp. Systems*, vol. 23, no. 10, pp. 19 145–19 160, Oct. 2022.

[34] M. Hofbauer, C. B. Kuhn, G. Petrovic, and E. Steinbach, "Adaptive multi-view live video streaming for teledriving using a single hardware encoder," in *Proc. IEEE International Symposium on Multimedia (ISM)*, Dec. 2020.

[35] N. Adwani, K. Silvestrini-Cordero, R. Rojas-Cessa, T. Han, and C. Wang, "A special-purpose video streaming codec for internet-based remote driving," in *Proc. IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Jul. 2024.

[36] N. Chakraborty, Y. Fang, A. Schreiber, T. Ji, Z. Huang, A. Mihigo *et al.*, "Towards real-time generation of delay-compensated video feeds for outdoor mobile robot teleoperation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2025.

[37] A. Alam, B. Besselink, V. Turri, J. MåRtensson, and K. H. Johansson, "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency," *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 34–56, 2015.

[38] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, 2000.

[39] Z. Seifaei, T. Reißland, and N. Franchi, "Communications-control co-design in cooperative platooning: Quantifying the stability effects of communications delay and loss," in *Proc. IEEE Vehicular Technology Conference (VTC2025-Fall)*, Chengdu, China, 2025, p. 7.

[40] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7907–7922, 2019.
[41] C. Hong, H. Shan, M. Song, W. Zhuang, Z. Xiang, Y. Wu, and X. Yu, "A joint design of platoon communication and control based on LTE-V2V," *IEEE Trans. on Veh. Technology*, vol. 69, no. 12, pp. 15 893–15 907, 2020.
[42] J. M. Kennedy, J. Heinovski, D. E. Quevedo, and F. Dressler, "Centralized model predictive control with human-driver interaction for platooning," *IEEE Trans. on Veh. Technology*, vol. 72, no. 10, pp. 12 664–12 680, 2023.
[43] C. Quadri, V. Mancuso, M. A. Marsan, and G. P. Rossi, "Edge-based platoon control," *Computer Communications*, vol. 181, pp. 17–31, 2022.
[44] P. Wijnbergen, M. Jeeninga, R. de Haan, and E. Lefeber, "Longitudinal and lateral control of vehicle platoons: A unifying framework to prevent corner cutting," *Automatica*, vol. 177, 2025.
[45] A. Bayuwindra, J. Ploeg, E. Lefeber, and H. Nijmeijer, "Combined longitudinal and lateral control of car-like vehicle platooning with extended look-ahead," *IEEE Trans. on Control Systems Technology*, vol. 28, no. 3, pp. 790–803, May 2020.
[46] Y. Díaz, J. Dávila, and M. Mera, "Leader-follower formation of unicycle mobile robots using sliding mode control," *IEEE Control Systems Letters*, vol. 7, pp. 883–888, 2023.
[47] A. Bayuwindra, L. Wonohito, and B. R. Trilaksono, "Design of DDPG-based extended look-ahead for longitudinal and lateral control of vehicle platoon," *IEEE Access*, vol. 11, pp. 96 648–96 660, 2023.
[48] A. Albers and T. Düser, "Implementation of a vehicle-in-the-loop development and validation platform," in *FISITA World Automotive Congress*, Jun. 2010.
[49] J.-P. Busch, L. Ostendorf, G. Linden, L. Reiher, T. Beemelmanns, B. Lampe, T. Woopen, and L. Eckstein, "karl. – A research vehicle for automated and connected driving," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, Detroit, USA, Jun. 2026.
[50] RoboRacer Autonomous Racing Community, "RoboRacer (formerly F1TENTH): an open-source autonomous racing platform," accessed: Jul. 7, 2025. [Online]. Available: https://github.com/f1tenth
[51] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1TENTH: An open-source evaluation environment for continuous control and reinforcement learning," in *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, ser. Proceedings of Machine Learning Research (PMLR), vol. 123, Dec. 2020.
[52] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017.
[53] Quectel, "LTE&5G Linux USB driver user guide," Quectel Wireless Solutions Co., Ltd., Tech. Rep. Rev. 2.0, Dec. 2019.
[54] 3rd Generation Partnership Project (3GPP), "5G; study on channel model for frequencies from 0.5 to 100 GHz (release 18)," 3GPP, Technical Report (TR) TR 38.901, May 2024, version 18.0.0.
[55] A. Abhishek, S. Mukherjee, M. Aahan, M. Kohli, R. Feick, R. Valenzuela *et al.*, "Around-corner and over-top 28 GHz measurement in Manhattan: Path loss and AoA for MU-MIMO," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, May 2025.
[56] O. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 581–585, May 2020.
[57] S. Häger, S. Böcker, and C. Wietfeld, "Reflection modeling of modular passive IRS geometries," *IEEE Wireless Communications Letters*, vol. 14, no. 5, pp. 1366–1370, May 2025.
[58] E. Björnson, Ö. T. Demir, and L. Sanguinetti, "A primer on near-field beamforming for arrays and reconfigurable intelligent surfaces," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Oct. 2021.
[59] R. Deban, H. Boutayeb, K. Wu, and J. Conan, "Deterministic approach for spatial diversity analysis of radar systems using near-field radar cross section of a metallic plate," *IEEE Trans. on Antennas and Propagation*, vol. 58, no. 3, pp. 908–916, Mar. 2010.
[60] S. Häger, M. Kaudewitz, F. Schmickmann, S. Böcker, and C. Wietfeld, "Field performance evaluation of a mechatronic reflector system in a private mmWave network environment," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 5005–5029, 2025.
[61] D. K. Pekcan and E. Ayanoglu, "Achieving optimum received power for discrete-phase RISs with elementwise updates in the least number of steps," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2706–2722, 2024.
[62] M. Danger, C. Arendt, H. Schippers, S. Böcker, M. Muehleisen, P. Becker, J. B. Caro, G. Gjorgjievska, M. A. Latif, J. Ansari, N. Beckmann, N. König, R. Schmitt, and C. Wietfeld, "Performance evaluation of IRS-enhanced mmWave connectivity for 6G industrial networks," in *Proc. IEEE International Symposium on Measurements & Networking (M&N)*, Jul. 2024.
[63] 3rd Generation Partnership Project (3GPP), "5G; NR; user equipment (UE) radio access capabilities (release 16)," 3GPP, Technical Specification (TS) 38.306, Oct. 2025, version 19.0.0.
[64] Unity Technologies, "ROS-TCP-Endpoint for Unity (ROS2 v0.7.0)," 2022, accessed: Feb. 2, 2026. [Online]. Available: https://github.com/Unity-Technologies/ROS-TCP-Endpoint/releases/tag/ROS2v0.7.0
[65] H. Schippers, T. Gebauer, K. Heimann, and C. Wietfeld, "RISE: Multi-link proactive low-latency video streaming for teleoperation in fading channels," in *Proc. IEEE Vehicular Technology Conference (VTC-Spring)*, Oslo, Norway, Jun. 2025.
[66] T. Gebauer, M. Patchou, and C. Wietfeld, "SEAMLESS: Radio metric aware multi-link transmission for resilient rescue robotics," in *Proc. IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR)*, Nov. 2023.
[67] "GStreamer: Open source multimedia framework," accessed: Nov. 16, 2025. [Online]. Available: https://gstreamer.freedesktop.org/
[68] C. Bachhuber and E. Steinbach, "A system for high precision glass-to-glass delay measurements in video communication," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Sep. 2016.
[69] Y.-K. Wang, R. Even, and S. Wenger, "RTP payload format for high efficiency video coding (HEVC)," Internet Request for Comments, Internet Engineering Task Force, RFC 7798, Mar. 2016. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7798
[70] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Proc. International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, Jul. 2014.
[71] M. Patchou, J. Tieman, C. Arendt, S. Böcker, and C. Wietfeld, "Realtime wireless network emulation for evaluation of teleoperated mobile robots," in *Proc. IEEE Symposium on Safety, Security, and Rescue Robotics (SSRR)*. Sevilla, Spain: IEEE, Nov. 2022.
[72] A. Schimpe, J. Feiler, S. Hoffmann, D. Majstorović, and F. Diermeyer, "Open source software for teleoperated driving," in *Proc. International Conference on Connected Vehicle and Expo (ICCVE)*, Mar. 2022.
[73] A. Schimpe, S. Hoffmann, and F. Diermeyer, "Adaptive video configuration and bitrate allocation for teleoperated vehicles," in *Proc. IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, Jul. 2021.
[74] D. Overbeck, N. A. Wagner, R. Wiebusch, J. Püttschneider, T. Faulwasser, and C. Wietfeld, "Data-driven proactive uplink slicing enabling real-time control within an Open RAN testbed," in *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, May 2024.
[75] N. A. Wagner and C. Wietfeld, "O-RACES: Proactive AI-driven scheduling in Open RAN for 6G-networked humanoid robots," in *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, May 2026.
[76] Netflix, Inc., "VMAF - video multi-method assessment fusion," accessed: Feb. 2, 2026. [Online]. Available: https://github.com/Netflix/vmaf
[77] I. B. Frontelo, P. P. García, and M. Orduna, "Video quality assessment dataset with advanced AI and QoE metrics for teleoperated driving," in *Proc. 17th International Conference on Quality of Multimedia Experience (QoMEX)*, 2025.
[78] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli *et al.*, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, pp. 147–183, Oct. 2021.
[79] H. Sun, D. Tavernini, P. Gruber, B. Li, W. Zhuang, and U. Montanaro, "Distributed model predictive control for vehicle platooning with coupled constraints under asynchronous communication protocols," *IEEE Trans. on Veh. Technology*, 2025.
[80] Q. Chen, D. J. Sun, J. Xia, X. Liu, and C. An, "String stable and robust optimal longitudinal control for vehicle platoons: a finite time disturbance observer-based tube MPC," *Nonlinear Dynamics*, vol. 113, pp. 16 593–16 610, 2025.
[81] M. H. Shaham and T. Padır, "Distributed model predictive control for heterogeneous platoons with affine spacing policies and arbitrary communication topologies," in *Proc. IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024, pp. 2914–2921.
[82] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2017.

AUTHORS

HENDRIK SCHIPPERS (Student Member, IEEE) received the M.Sc. degree in electrical engineering and information technology from TU Dortmund University, Germany, in 2021. He is currently pursuing the Ph.D. degree at the Chair of Communication Networks (ComNets) at TU Dortmund University. His research interests cover cellular 5G and beyond networks, focusing on the reliable prediction of connectivity quality for resource- and energy-efficient communication.

NIKLAS A. WAGNER (Student Member, IEEE) received the M.Sc. in electrical engineering and information technology from TU Dortmund University, Germany, in 2023. In 2023, he was a visiting researcher at the University of Oulu, Finland at ICON group. He is currently pursuing a Dr.-Ing. degree with the Chair of Communication Networks (ComNets) at TU Dortmund University. His research interest is in heterogeneous critical communications toward an Open 6G RAN.

LENNART REIHER received the M.Sc. degree in Computational Engineering Science from RWTH Aachen University, Aachen, Germany, in 2021. He is currently Group Leader Connectivity for the Vehicle Intelligence & Automated Driving department at the Institute for Automotive Engineering (ika) at RWTH Aachen University. His research interests cover full-stack development for automated and connected driving, particularly in the context of (edge-)cloud connectivity.

HARUN TEPER (Member, IEEE) received the B.Sc. and M.Sc. degrees from TU Dortmund University (TUD), in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the Design Automation for Embedded Systems Group. His research interests include real-time systems, ROS 2, end-to-end latency analysis, latency optimization, and autonomous driving.

LUKAS OSTENDORF received the M.Sc. degree in Automation Engineering from RWTH Aachen University, Aachen, Germany, in 2023. He is currently a researcher at the Vehicle Intelligence & Automated Driving department at the Institute for Automotive Engineering (ika) at RWTH Aachen University. His research interest is in the end-to-end development of methods and systems for automated and connected mobility.

JIAN-JIA CHEN (Senior Member, IEEE) received the B.S. degree from the Department of Chemistry, National Taiwan University, in 2001, and the Ph.D. degree from the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, in 2006. He was a Junior Professor with the Department of Informatics, Karlsruhe Institute of Technology (KIT), Germany, from May 2010 to March 2014. He is currently a Professor with the Department of Informatics, TU Dortmund University, Germany. His research interests include real-time systems, embedded systems, energy-efficient scheduling, power-aware designs, temperature-aware scheduling, and distributed computing. He received the European Research Council (ERC) Consolidator Award, in 2019. He has received more than ten best paper awards and outstanding paper awards and has involved in technical committees in many international conferences.

LUTZ ECKSTEIN studied mechanical engineering and received the Dr.-Ing. degree from the University of Stuttgart, Stuttgart, Germany, in 2000. During his time in industry, he worked on active safety systems and human-machine interactions for DaimlerChrysler AG and BMW AG. He is currently a Full Professor and the Head of the Institute for Automotive Engineering (ika) at RWTH Aachen University, Aachen, Germany, since 2010. His research is holistically covering the entire vehicle and mobility ecosystem, from vehicle concepts, dynamics, and acoustics over energy and powertrain systems to automated driving and traffic psychology. In 2021, he was appointed to the Scientific Advisory Board of the German Federal Ministry of Transport (BMV). In 2023, he was elected President of the Association of German Engineers e.V. (VDI). Together with these two roles, he is shaping the future of mobility on national and international level.

CHRISTIAN WIETFELD (Senior Member, IEEE) received the Dr.-Ing. and Dipl.-Ing. degrees from RWTH Aachen University, Aachen, Germany. He is currently a Full Professor and the Head of the Chair of Communication Networks (ComNets) at TU Dortmund University, Dortmund, Germany. For more than 30 years, he has been a coordinator of and a contributor to large-scale research projects on Internet-based mobile communication systems within both academia and industry at RWTH Aachen (1992-1997), Siemens AG (1997-2005), and TU Dortmund (2005 to date). His current research interests include the design and performance evaluation of 5G and 6G communication networks for cyber-physical systems in energy, transport, robotics, and emergency response. In 2025, Prof. Wietfeld was appointed as Fellow of the Information Technology Society ITG of the German Association of Electrical Engineers (VDE). He is a member of the Technical Editorial Board of the IEEE Wireless Communications Magazine. In addition to more than 20 IEEE Best Paper Awards, he received the ITU-T Outstanding Contribution Award for his work on the standardization of next-generation mobile network architectures.