# A Real-World Testbed for Joint Communication and Control in 5G-Enabled IIoT Networks

Filippo Bragato
*WiLab/CNIT and University of Padova*
Padova, Italy
filippo.bragato@dei.unipd.it

Seyedali Hosseinishamoushaki
*WiLab/CNIT and University of Padova*
Padova, Italy
seyedali.hosseinishamoushaki@dei.unipd.it

Marco Giordani
*WiLab/CNIT and University of Padova*
Padova, Italy
marco.giordani@unipd.it

Malte Schellmann
*Huawei Technologies*
Munich, Germany
malte.schellmann@huawei.com

Josef Eichinger
*Huawei Technologies*
Munich, Germany
joseph.eichinger@huawei.com

Michele Zorzi
*WiLab/CNIT and University of Padova*
Padova, Italy
zorzi@dei.unipd.it

***Abstract*—The integration of 5G networks into Industrial Internet of Things (IIoT) robotic applications represents a significant step toward highly flexible, reconfigurable, scalable, and intelligent manufacturing environments. In this scenario, the performance of the control system directly depends on the underlying communication network, a paradigm referred to as Joint Communication and Control (JCC), making it essential to understand the interaction between the two systems under realistic operating conditions. In this paper, we present an experimental testbed for JCC to evaluate the impact of packet loss and latency on an external control system guiding a commercial industrial mobile robot along a predefined trajectory. Specifically, the robot transmits high-rate sensing data over a real private 5G network based on Open Air Interface (OAI). Based on this data, the remote controller, implemented using ROS 2, generates and returns proper driving commands accordingly. The performance is analyzed under different network load conditions introduced by a secondary 5G user. Real-world experimental results indicate that the system can tolerate moderate latency with negligible degradation. However, in case of severe packet loss, the control error increases significantly, and can ultimately compromise the stability of the control loop.**

***Index Terms*—Industrial Internet of Things (IIoT); Joint Communication and Control (JCC); experimental testbed.**

## I. Introduction

The integration of 5G New Radio (NR) networks into Industrial Internet of Things (IIoT) applications is promoting the transition from onboard control, where sensing and computation are co-located within the same platform, to networked control architectures, in which control functions are partially or fully offloaded to remote edge servers. This paradigm, referred to as Joint Communication and Control (JCC), enables more scalable and flexible industrial automation, but also introduces a strong dependency between the control performance and the underlying communication network [1]. In particular, network impairments, such as latency, jitter, and packet loss, directly affect the stability of the control application. As such, communication and control can no longer be designed independently, but must be jointly optimized, which requires a rigorous understanding of the interaction between the two [2], [3].

Despite the large body of literature on JCC and its potential for industrial automation, most existing works, e.g., [4]–[7], rely on analytical models or simulations to evaluate the impact of communication on control [8], providing preliminary insights into stability conditions and working regimes. More recent studies have investigated the feasibility of 5G for industrial automation and robotics, and quantified the latency and reliability requirements necessary to support real-time control applications [9]–[11]. Similarly, in our previous work [12], we developed a simulation framework to numerically evaluate the relationship between communication performance, particularly packet loss, and control accuracy. While these approaches provide valuable insights, they often fail to capture the complexity of real-world deployments, including protocol stack components, hardware constraints, software integration, and implementation effects. As a result, experimental validation on real platforms remains limited. Some works have considered 5G testbeds for industrial applications, demonstrating the feasibility of wireless control and providing initial performance evaluations [13]–[15]. However, these studies focus primarily on the communication layer, and rely on simplified or idealized controller implementations. Conversely, other contributions mainly optimize the robotic applications, but depend on emulated network conditions rather than on fully integrated 5G systems [16], [17].

In this paper, we address this gap by presenting an experimental testbed for JCC based on a fully operational private 5G network implemented using Open Air Interface (OAI) platforms. Specifically, we consider a closed-loop control system where a commercial industrial Automated Guided Vehicle (AGV) generates and transmits sensor data over the 5G network to an external controller implemented in Robot Operating System 2 (ROS 2). Then, the controller processes the received data, and generates control commands for the AGV to ensure that it accurately follows a predefined trajectory. We experimentally evaluate the impact of communication impairments, particularly latency and packet loss, on the tracking accuracy under different network load conditions. To

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

emulate latency-critical control applications, we also consider a scenario where packets that accumulate delays above a predefined threshold are discarded, reflecting strict real-time constraints. Our results reveal that, while moderate latency can be tolerated by acting on the controller parameters, packet loss due to delayed packets may increase the tracking error, and compromise the stability of the closed loop.

The remainder of the paper is organized as follows. In Sec. II we describe our experimental setup. In Secs. III and IV we present our experimental methodology and results, respectively. In Sec. V we conclude the paper with suggestions for future work.

## II. Experimental Setup

In this section we describe the main components of our experimental setup, illustrated in Fig. 1, including the scenario (Sec. II-A), the hardware infrastructure (Sec. II-B), and the controller architecture (Sec. II-C).

### A. Scenario Description

We consider a smart-factory-like indoor scenario where an AGV is required to follow a predefined trajectory based on driving commands received from an external (remote) control unit, co-located with a 5G Next Generation Node Base (gNB). Since the AGV uses the 5G wireless network to interact with the control unit, this scenario is an example of JCC. Specifically, the AGV periodically transmits sensing and odometry data to the 5G gNB controller for processing. The gNB controller, in turn, uses the received data to compute the absolute position and orientation of the AGV with respect to the factory floor. Based on this information, it derives and sends driving control commands to the AGV to ensure that it follows the desired trajectory. These bidirectional transmissions make the system sensitive to communication errors, which may affect driving operations. In addition to the AGV and the gNB, our scenario includes another 5G User Equipment (UE), e.g., a robot or industrial machine in the factory floor, generating background traffic, to consider a more congested and realistic network.

### B. Hardware and Network Infrastructure

Our setup consists of the following hardware components.

*a) AGV (Primary UE):* The AGV consists of a Husarion ROSbot XL autonomous platform, which is a differential drive robot equipped with a 360-degree Light Detection and Ranging (LiDAR) sensor for localization and perception of the environment, an inertial measurement unit (IMU), and wheel encoders. The AGV can drive autonomously via an onboard native ROS 2 controller, that we extended to also support teleoperated driving via our remote control system, as described in Sec. II-C. The LiDAR is an RPLIDAR S3, which does not provide a point cloud of the environment, but rather a set of distance measurements at different angles. The IMU provides information about the orientation of the AGV, while the wheel encoders provide information about the distance traveled by each wheel. The AGV is interfaced via a wired Ethernet connection with an OAIBOX 40 Software Defined Radio (SDR) acting as the Primary 5G NR UE. Specifically, the OAIBOX 40 is used to transmit LiDAR and odometry data to the 5G NR gNB, and receive and execute linear and angular velocity commands (`cmd_vel`).

*b) 5G NR network:* The 5G Core (5GC) and 5G gNB are implemented using an OAIBOX MAX SDR, equipped with an Ettus USRP X410 for physical-layer signal processing. The OAIBOX provides a complete implementation of the 3GPP 5G NR protocol stack based on OAI. Notably, the baseline OAI stack natively implements only a Proportional Fair (PF) scheduler, which is similar to 5G NR Grant-Based Scheduling (GBS). Beyond the RAN functionality, this node acts as the remote controller for the AGV, hosting the path planning nodes and the logic for the trajectory execution.

The 5G gNB is configured to operate in the n77 frequency band, with a carrier frequency of $3.7\,\mathrm{GHz}$ and a bandwidth of $40$ MHz. A specific Time Division Duplex (TDD) pattern is implemented, following a DDFUU (Downlink-Downlink-Flexible-Uplink-Uplink) slot structure. This configuration provides a balanced allocation between downlink and uplink resources, which is critical for industrial robotics where uplink sensor data is as vital as downlink control commands, while the F slots can be dynamically assigned to either direction depending on traffic demands.

*c) Secondary UE:* A second OAIBOX 40 is used as an independent UE within the same radio cell, serving as a controlled source of network traffic and interference. Specifically, this node generates *iperf3* User Datagram Protocol (UDP) floods, forcing the gNB scheduler to distribute the (limited) available Resource Blocks (RBs) between the AGV and the rest of the network. This approach provides valuable insights into the scalability performance of our experimental platform under diverse traffic conditions.

### C. Control System Architecture

The control system is implemented in ROS 2. It is responsible for managing the behavior of the AGV, and ensures that it follows the desired trajectory. It consists of three components.

*a) Map creator:* It processes the LiDAR data to create a 2D map of the environment. This is done using the Simultaneous localization and mapping (SLAM) technique, which allows to create a map of the environment while simultaneously localizing the AGV within that map. To do so, we use the `slam_toolbox` ROS 2 package, which provides various SLAM algorithms. The output of the map creator is a 2D occupancy grid map, which represents the environment as a grid of squared cells of size $5 \times 5$ cm$^2$, that can be either occupied, free, or unknown.

*b) Localization system:* It uses the map created by the map creator, along with the LiDAR and odometry data from the wheel encoders, to estimate the position and orientation of the AGV. This is done using a particle filter, which is a probabilistic method for estimating the state of a system based on noisy measurements. To do so, we use the `nav2` ROS 2 package, which provides various navigation and localization

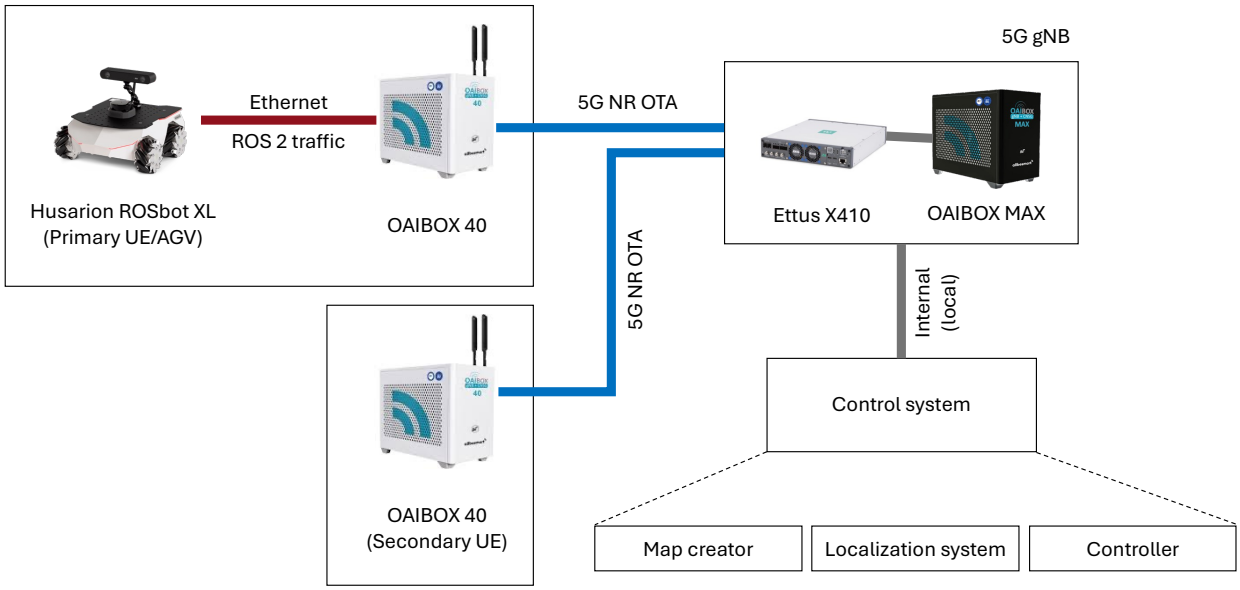



Fig. 1: General overview of the proposed JCC testbed.

algorithms via Adaptive Monte Carlo Localization (AMCL). The output of the localization system is an estimate of the position and orientation of the AGV within the map.

*c) Controller:* It computes control commands based on the estimated position and orientation of the AGV from the localization system and the desired trajectory. To do so, we use two simple Proportional-Integral–Derivative (PID) controllers, which compute the linear and angular velocities that the AGV should set to minimize the error $e(t)$ between its current position and orientation and the desired trajectory. The output of each PID controller can be described as

$$y(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}, \qquad (1)$$

where $K_p$, $K_i$ and $K_d$ are the proportional, integral and derivative gains of the controller, respectively, and were tuned empirically, and their values are reported in Sec. III. The control commands are published on the `/cmd_vel` topic, to which the AGV subscribes to receive the control inputs.

The controller uses a lookahead-based trajectory tracking strategy, as described in [12], where the reference point for the calculation of the optimal driving commands is not the robot's current position along the path, but a waypoint corresponding to its expected position after a fixed time horizon. The lookahead parameter $\tau_a$ determines how far ahead along the reference trajectory the controller computes its tracking objective: increasing $\tau_a$ allows the controller to anticipate possible upcoming changes in the path, while decreasing $\tau_a$ provides more accurate tracking, though only relative to the current segment of the path.

All control components operate as local software processes at the gNB side, so they are also implemented on the OAIBOX MAX. Notably, they interact internally with the OAI stack, thereby enabling closed-loop robotic control over the 5G link.

## III. Experimental Methodology

*a) Scenario:* The experimental setup consists of a Husarion ROSbot XL AGV (i.e., the Primary UE) sending LiDAR and odometry data to the 5G gNB (i.e., the OAIBOX MAX), which in turn sends back to the AGV proper control commands. We run experimental tests in a real-world large-scale scenario, where the robot moves at a speed of up to 1 m/s in an area of approximately $8 \times 8$ m$^2$. The objective of our tests is to quantify how communication latency and packet loss affect the control performance in case of network congestion.

*b) Traffic:* The AGV generates LiDAR packets of size 125 and 250 KB at a rate of 10 Hz, resulting in an average data rate $R_{\text{AGV}}$ of 10 and 20 Mbps, respectively. In parallel, the Secondary UE generates additional UDP data at a configurable data rate $R_{\text{UE}}$ from 0 (no concurrent traffic) to 20 Mbps, to emulate different network congestion levels.

*c) Controller:* The parameters of the PID controllers are tuned heuristically based on the performance of the AGV when the target position or orientation are fixed. Specifically, with reference to Eq. (1), we set $K_p = 1.5$, $K_i = 0$ and $K_d = 0.5$ for the linear velocity controller, and $K_p = 1.5$, $K_i = 0$ and $K_d = 0.6$ for the angular velocity controller.

*d) Experimental procedure:* In each experiment, the AGV is required to follow a predefined eight-shape trajectory based on the received control commands. The trajectory and controller parameters are kept fixed across all experiments to ensure consistency. We run closed-loop experiments for a duration from 30 to 60 s. During this time, sensor data, control commands, and network-level information are continuously exchanged over the 5G network. Each experiment is executed five times, i.e., the AGV completes five laps along the same trajectory, to capture different variability conditions in dy-

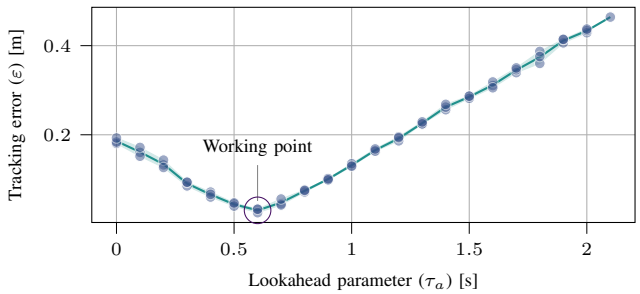



Fig. 2: Measurements and interpolation of $\varepsilon$ of the controller as a function of the lookahead parameter $\tau_a$.

namic wireless networks. The collected data is then aggregated to obtain statistically reliable results.

*e) Performance metrics:* We evaluate both control and communication performance. Regarding the performance of the control system, we measure the tracking error $\varepsilon$, i.e., the Root Mean Squared Error (RMSE) between the position and orientation of the AGV along the current trajectory $p$ and the optimal/desired trajectory $p^*$. Position and orientation are measured based on LiDAR perception and odometry data, as well as on control commands. This data is collected using dedicated ROS 2 topics, and stored in a database for analysis.

Regarding the performance of the 5G NR network, we compute the End-to-End (E2E) latency, measured using timestamps generated within the ROS 2 framework, knowing that all network nodes (AGV, UE, and gNB) are time-synchronized prior to and during each experimental run. We also compute the packet loss, reflecting the ratio between transmitted and successfully received packets.

## IV. Experimental Results

In this section we validate the accuracy of the proposed JCC testbed. In Sec. IV-A we determine the optimal lookahead parameter of the controller under ideal network conditions. Then, in Sec. IV-B we evaluate the control performance as a function of the 5G communication network metrics.

### A. Tuning of the Lookahead Parameter

As far as the controller is concerned (see Sec. II-C), proper tuning of the lookahead parameter $\tau_a$ is essential to balance responsiveness and stability. Indeed, we conducted a set of preliminary experimental tests using the Husarion ROSbot XL under ideal network conditions, i.e., the controller is implemented and executed directly onboard the robot, thereby eliminating the effect of latency, jitter, packet loss, or other disturbances from the 5G network. The goals are to: (i) validate our controller implementation; (ii) tune the controller, and identify the optimal value of $\tau_a$ to use in subsequent experiments; and (iii) establish an upper performance bound for the system. These results serve as a baseline reference, to isolate and evaluate the intrinsic tracking performance of the control algorithm, before we introduce the influence of the 5G network.

In Fig. 2 we evaluate the impact of $\tau_a$ on the tracking error between the reference trajectory and the actual (executed)

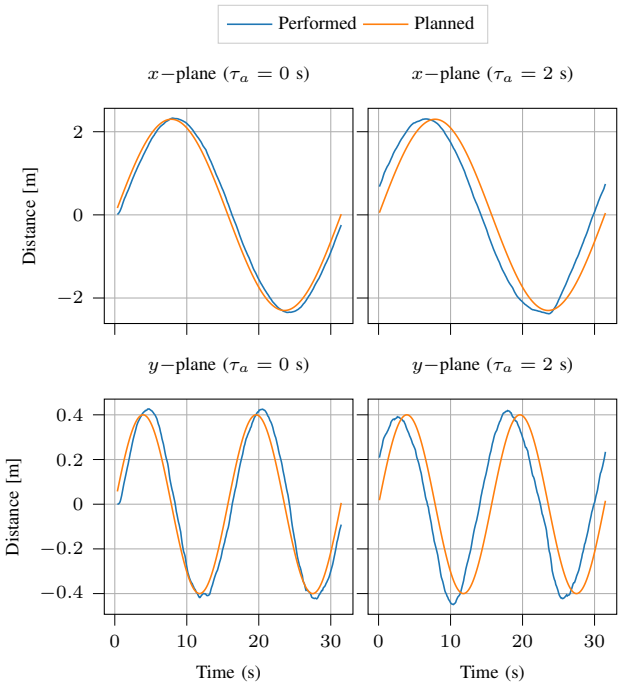
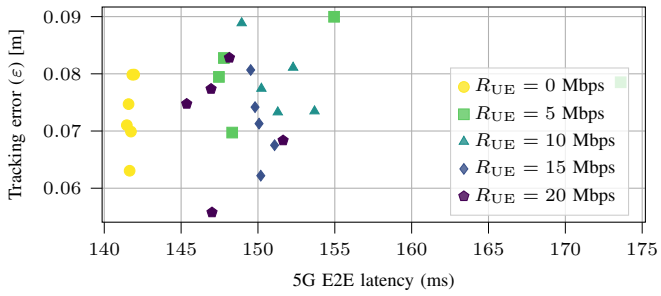



Fig. 3: Position of the robot over time (planned/expected in orange, actual/performed in blue) for $\tau_a = 0$ s (right) and $\tau_a = 2$ s (left).

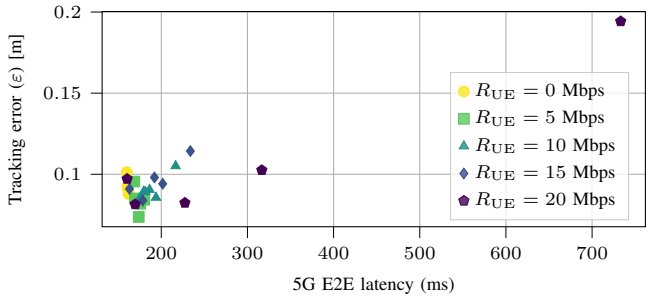
trajectory of the robot. A clear minimum is observed, so there exists an optimal lookahead value $\tau_a^* \simeq 0.6$ s. For $\tau_a < \tau_a^*$, the controller provides insufficient anticipation of the robot dynamics along the reference trajectory, which can result in phase lags (in control theory terminology). On the other side, for $\tau_a > \tau_a^*$, the controller anticipates the future trajectory too aggressively, which can lead to an oscillatory behavior.

To better understand this effect, in Fig. 3 we depict the position of the robot over time (planned in orange, actual/performed in blue) in the $x - y$ plane considering two extreme configurations: $\tau_a = 0$ s (left) and $\tau_a = 2$ s (right). For $\tau_a = 0$ s, the controller makes decisions based solely on the current (desired) position of the robot. Therefore, it is unable to anticipate and properly compensate for future robot dynamics, leading to local distortions in the trajectory waveform. Notably, the robot systematically lags behind the desired trajectory, mainly due to the non-negligible network delay associated with the transmission and execution of control commands. For $\tau_a = 2$ s, the tracking error increases. This is because the controller predicts too far along the planned trajectory, and generates control commands that may not be consistent with the actual dynamic response of the robot. Although the resulting trajectory waveform appears less distorted, the phase mismatch between the actual and planned position of the robot is more pronounced. These results confirm that the impact of $\tau_a$ is non-negligible, though there exists an optimal value $\tau_a^*$ that balances the trade-off between insufficient and excessive anticipation.

Therefore, we use $\tau_a = \tau_a^* = 0.6$ s in the rest of the experiments, corresponding to the minimum $\varepsilon$. This configuration represents the best achievable tracking performance in




(a) The AGV generates constant uplink traffic at $R_{\mathrm{AGV}} = 10$ Mbps.




(b) The AGV generates constant uplink traffic at $R_{\mathrm{AGV}} = 20$ Mbps.

Fig. 4: Mean tracking error $\varepsilon$ as a function of the 5G E2E latency, for various values of $R_{\mathrm{AGV}}$ (top and bottom) and $R_{\mathrm{UE}}$ (in the legend).

the absence of network effects.

### B. Numerical Results

Now, we evaluate the degradation of the tracking error introduced by the 5G communication network, considering the full experimental setup described in Sec. II. In Fig. 4 we plot the mean tracking error $\varepsilon$ between the planned and actual trajectory of the AGV, as a function of the source rate of the Secondary UE $R_{\mathrm{UE}}$ (in the legend) and the resulting E2E latency of the 5G network.[1] In Fig. 4a (Fig. 4b) the AGV generates uplink traffic at $R_{\mathrm{AGV}} = 10$ (20) Mbps. For $R_{\mathrm{AGV}} = 10$ Mbps (Fig. 4a), $\varepsilon$ remains relatively stable as $R_{\mathrm{UE}}$ increases, meaning that the controller can effectively compensate for the latency (in the range between 140 and 175 ms). This suggests that the network is not congested, and operates far from its capacity limits. For $R_{\mathrm{AGV}} = 20$ Mbps (Fig. 4b), the robot generates more traffic which, combined with the traffic generated by the Secondary UE, may cause additional delays. In this regime, $\varepsilon$ gradually increases as the latency increases up to 700 ms where the network becomes more congested. Notably, control commands from the gNB become progressively delayed and outdated, preventing the robot from driving properly along the planned trajectory. This is the typical behavior of a Networked Control System (NCS) [18]: performance degradation is not necessarily gradual, but shows a non-linear transition as the network latency approaches the stability margin of the controller.

[1]The measured E2E latency is relatively high, since the OAIBOX platform introduces significant computational delays, including protocol stack processing, buffering, and hardware interfacing.

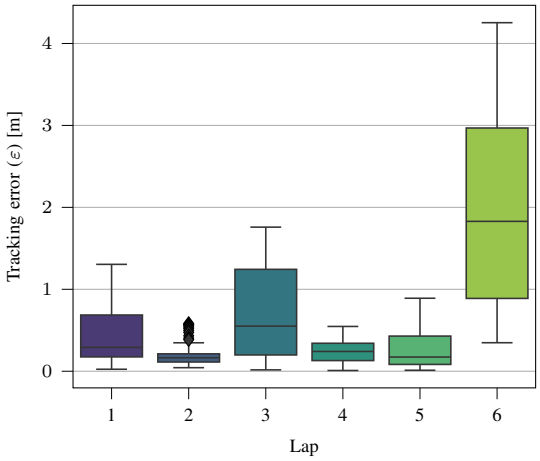



Fig. 5: Tracking error $\varepsilon$ in different laps for $R_{\mathrm{AGV}} = 10$ Mbps. The Secondary UE generates no traffic for the first 5 laps, and then transmits at $R_{\mathrm{UE}} = 10$ Mbps in Lap 6. Packets are discarded if they are not successfully received within a delay threshold of 175 ms.

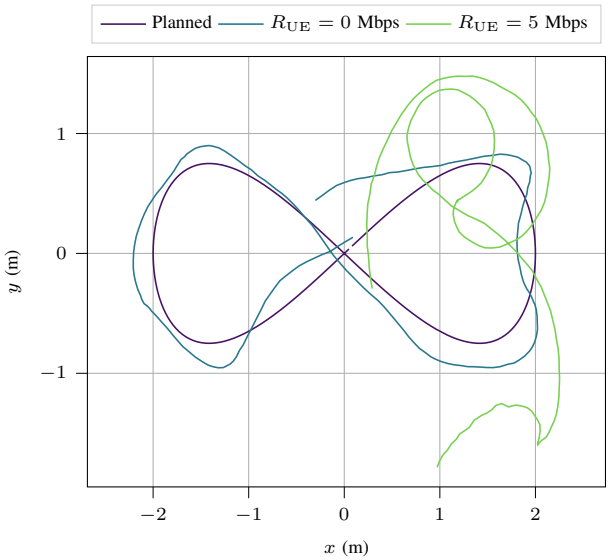



Fig. 6: AGV trajectory (planned in purple, performed in green), with and without the Secondary UE, for $R_{\mathrm{AGV}} = 10$ Mbps. Packets are discarded if they are not successfully received within a delay threshold of 175 ms.

Next, we ran additional tests in a more constrained scenario where control commands must be received within a predefined latency threshold, set experimentally to 175 ms, otherwise packets are discarded. This application regime is consistent with typical 5G NR Layer-2 measurement specifications, where packet losses also account for packets successfully received but delayed more than a given threshold [19]. In fact, under critical realistic conditions, delayed data may be worse than receiving no data at all, as the controller may compute incorrect control actions that may even destabilize the entire system. In Fig. 5 we report the boxplots of the tracking error computed for each lap of the AGV along the trajectory, for $R_{\mathrm{AGV}} = 10$ Mbps. The Secondary UE generates no traffic in the first 5 laps, and then transmits data at a source rate $R_{\mathrm{UE}} = 10$ Mbps in Lap 6. We observe that $\varepsilon$ increases a lot

compared to the previous results with no packet dropping, even with no Secondary UE, and eventually explodes when it starts generating traffic. Moreover, $\varepsilon$ varies a lot, indicating large variance and jitter, which may render the control infeasible as the network latency exceeds the physical response capability of the ROSbot XL motors.

To further illustrate this trend, in Fig. 6 we compare the optimal/planned trajectory (in purple) with the actual/performed trajectory (in blue or green) for $R_{\mathrm{UE}} = 0$ and $5$ Mbps, in the $x - y$ plane. It is evident that the AGV fails to follow the reference path, deviating significantly from the desired trajectory. For $R_{\mathrm{UE}} = 5$ Mbps, the AGV even hits the walls of the room, and is unable to reach the desired destination. These results demonstrate that packet loss is the dominant component of the control error, which is in line with the conclusions of our simulation work in [12].

## V. Conclusions and Future Work

In this paper we presented an experimental testbed to evaluate the joint impact of communication and control in a representative IIoT robotic system. The testbed consists of a Husarion ROSbot XL AGV sending LiDAR and odometry data to a private 5G gNB (implemented using an OAIBOX SDR), which in turn transmits control commands generated by a ROS 2 controller to drive the AGV along a predefined trajectory. Experimental results show that the system can tolerate moderate communication delays below a critical threshold by appropriately tuning the controller parameters, with no catastrophic degradation of control performance. However, as the traffic load increases, the system becomes progressively more sensitive to congestion, and the control performance deteriorates more rapidly. We also investigated a scenario in which packets with excessive delay are discarded. In this case, the control error may explode, with the robot deviating significantly from the reference trajectory.

Finally, it is important to acknowledge the limitations of the proposed testbed. First, the results are influenced by the characteristics of the OAI platform, which introduces severe processing and computational delays. Moreover, the considered controller, while representative, is relatively simple. As part of our future work, we plan to investigate more advanced control strategies, and extend the experimental campaign to larger-scale scenarios with additional simulated UEs. We will also evaluate advanced network protocols capable of prioritizing data transmissions based on the underlying control error, and consider grant-free [20] (rather than grant-based) schedulers.

## Acknowledgment

This work was carried out in the framework of the CNIT National Laboratory WiLab and the WiLab-Huawei Joint Innovation Center.

## References

[1] Z. Yang, J. Casas, A. Villamil, and G. P. Fettweis, "Joint resource allocation and controller selection for communication and control co-design," in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2025.

[2] Z. Yan, H. Zhou, H. Tabassum, and X. Liu, "Hybrid LLM-DDQN-Based Joint Optimization of V2I Communication and Autonomous Driving," *IEEE Wireless Communications Letters*, vol. 14, no. 4, pp. 1214–1218, Apr. 2025.

[3] G. Liu, J. Hu, Z. Ma, P. Fan, and F. R. Yu, "Joint Optimization of Communication Latency and Platoon Control Based on Uplink RSMA for Future V2X Networks," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 9, pp. 13 458–13 470, Sep. 2025.

[4] S. Wang, S. Zhu, C. Chen, F. Shen, W. Zhang, and X. Guan, "Communication and Control Co-Design for Heterogeneous Industrial IoT: A Logic-Based Stochastic Switched System Approach," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 10, pp. 3379–3393, Oct. 2025.

[5] X. Fang, C. Lei, W. Feng, Y. Chen, M. Xiao, N. Ge, and C.-X. Wang, "Sensing-Communication-Computing-Control Closed-Loop Optimization for 6G Digital Twin-Empowered Robotic Systems," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 10, pp. 3330–3346, Oct. 2025.

[6] W. Chen, X. Yu, F. Zhu, X. Chen, K. Zhang, C. Yu, H. Zhao, and A. V. Vasilakos, "Joint Communication and Control Optimization of a Multi-Vehicle Platooning System," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 11, pp. 19 017–19 027, Nov. 2025.

[7] G. Cuozzo, S. Cavallero, F. Pase, M. Giordani, J. Eichinger, C. Buratti, R. Verdone, and M. Zorzi, "Enabling URLLC in 5G NR IIoT Networks: A Full-Stack End-to-End Analysis," in *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022.

[8] D. K. Sah, M. Vahabi, and H. Fotouhi, "A Comprehensive Review on 5G IIoT Test-Beds," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 4139–4163, May 2025.

[9] B. S. Khan, S. Jangsher, A. Ahmed, and A. Al-Dweik, "URLLC and eMBB in 5G Industrial IoT: A Survey," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1134–1163, Jul. 2022.

[10] A. Narayanan, A. S. D. Sena, D. Gutierrez-Rojas, D. C. Melgarejo, H. M. Hussain, M. Ullah, S. Bayhan, and P. H. J. Nardelli, "Key Advances in Pervasive Edge Computing for Industrial Internet of Things in 5G and Beyond," *IEEE Access*, vol. 8, pp. 206 734–206 754, Nov. 2020.

[11] Y. Xu, H. Zhou, and Y. Deng, "Task-Oriented Semantics-Aware Communication for Wireless UAV Control and Command Transmission," *IEEE Communications Letters*, vol. 27, no. 8, pp. 2232–2236, Aug. 2023.

[12] F. Bragato, T. Fontana, M. Giordani, M. Schellmann, J. Eichinger, and M. Zorzi, "Network-Aware Control of AGVs in an Industrial Scenario: A Simulation Study Based on ROS 2 and Gazebo," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2025.

[13] M. Groshev, L. Zanzi, C. Delgado, X. Li, A. Oliva, and X. Costa-Pérez, "Energy-Aware Joint Orchestration of 5G and Robots: Experimental Testbed and Field Validation," *IEEE Transactions on Network and Service Management*, vol. 22, no. 4, pp. 3046–3059, Aug. 2025.

[14] G. Caso, O. Alay, A. Brunstrom, H. Koumaras, A. Díaz Zayas, and V. Frascolla, "Experimentation in 5G and beyond Networks: State of the Art and the Way Forward," *Sensors*, vol. 23, no. 24, Jan. 2023.

[15] S. Carbonara, D. Pugliese, E. Boffetti, F. Greco, B. Didonna, G. Grieco, A. Fascista, and L. A. Grieco, "An Indoor Experimental Testbed for 5G-Based UAV Control and Communication," in *21th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2025.

[16] G. Yoon and B. Hong, "Scalable and Robust Mobile Activity Fingerprinting via Over-the-Air Control Channel in 5G Networks," 2024. [Online]. Available: https://arxiv.org/abs/2409.12572

[17] T. Zhang, C. Xue, J. Wang, Z. Yun, N. Lin, and S. Han, "A Survey on Industrial Internet of Things (IIoT) Testbeds for Connectivity Research," 2024. [Online]. Available: https://arxiv.org/abs/2404.17485

[18] Y. Wang, S. Wu, C. Lei, J. Jiao, and Q. Zhang, "A Review on Wireless Networked Control System: The Communication Perspective," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 7499–7524, Mar. 2024.

[19] 3GPP, "NR; Layer 2 measurements (Release 16) ," *TS 38.314*, 2015.

[20] F. Pase, M. Giordani, S. Cavallero, M. Schellmann, J. Eichinger, R. Verdone, and M. Zorzi, "A Distributed Neural Linear Thompson Sampling Framework to Achieve URLLC in Industrial IoT," *IEEE Transactions on Wireless Communications*, vol. 25, pp. 12 996–13 010, Feb. 2026.