Girletti, L., Groshev, M.,Guimarães, C., Bernardos, C.J. y Oliva, A.de la (2020). An Intelligent Edge-based Digital Twin for Robotics. In *Proceedings of 2020 IEEE Globecom Workshops (GC Workshops).*

DOI: https://doi.org/10.1109/GCWkshps50303.2020.9367549

An Intelligent Edge-based Digital Twin for Robotics

Luigi Girletti*, Milan Groshev†, Carlos Guimarães‡, Carlos J. Bernardos¶, Antonio de la Oliva§
Universidad Carlos III de Madrid, Spain
Email: {lgirlett*, mgroshev†, cmagalha‡}@pa.uc3m.es, {cjbc¶, aoliva§}@it.uc3m.es

***Abstract*—Digital Twin is one of the use cases targeted by the fourth industrial revolution (Industry 4.0), which, through the digitalization of the robotic systems, will enable enhanced automation and remote controlling capabilities. Building upon this concept, this work proposes a solution for an Edge-based Digital Twin for robotic systems, which leverages on the cloud-to-things continuum to offload computation and intelligence from the robots to the network. This imposes stringent requirements over the communication technologies which are fulfilled in the proposed solution by relying on 5G. This solution is implemented in an E2E scenario and validated through a set of experimental evaluations. Results show that offloading the robot's functions to the edge is feasible when supported by the 5G connectivity. Moreover, the benefits of introducing intelligence and automation are also assessed.**

***Index Terms*—Digital Twin, Edge and Fog, 5G, Intelligence**

I. Introduction

The concept of Digital Twin, which has evolved towards a multitude of definitions in recent years [1], is identified as one of the main Industry 4.0 (I4.0) use cases. In I4.0, most of the processes are run by machines (e.g., robots) and, thus, a Digital Twin allows their remote control either by a human or intelligent agent. A Digital Twin consists of mapping the physical and virtual worlds (e.g., a physical machine with its virtual replica) supported by a continuous exchange of data that ties the two worlds together. It comprises *computation models* (e.g., physics and motion) that define its behavior and, driven by the advent of I4.0, also *analytics models* (e.g., learning and prediction) that define intelligent and automated behaviors. It may also include *visualization models* (e.g., 3D modeling and augmented reality) that mirror the physical entity as a visual representation. Finally, the *connected data* comprises the data exchange between the different models.

The high computation required by the *models* of a Digital Twin prompted the use of cloud computing. However, as cloud resides many networking hops away from the physical entity, the Digital Twin is highly dependent on the performance of the network(s) between the physical entity and its digital replica (in terms of e.g. packet-loss, jitter and latency). Consequently, offloading time-sensitive operations to the cloud may not be feasible.

Edge and Fog Computing [2] emerged as the natural candidate to fill this gap by placing computational resources closer to the user. In this sense, while ETSI provides a framework for Multi-access Edge Computing (MEC) over static substrates (e.g., data centers or servers) deployed at the edge, the industry-led Industrial Internet Consortium (IIC) extends this definition with less powerful and mobile computational substrates including the end-user devices. Offloading the abovementioned Digital Twin models to an infrastructure closer to the physical entity not only mitigates the problems associated with a cloud-based deployment, but also paves the way for further enhancements. Namely, (*i*) offloading of time-sensitive tasks; (*ii*) context information can be leveraged for automation and intelligence; (*iii*) computational and storage distribution; and (*iv*) enhanced privacy and security of the data.

Nevertheless, independently of the computing infrastructure used to implement the Digital Twin, mirroring the physical entity through a digital replica, especially in near real-time, imposes a new set of networking requirements, including low-latency, and high reliability and availability. The emerging 5G network can fulfill those requirements by bringing the performances of wired-like connections to the wireless domain, while keeping the benefits of a wireless connection (e.g., mobility and high connection density). Moreover, it enables differentiated services to coexist.

This work proposes an Intelligent Edge-based Digital Twin for Robotics, where the computation and analytics models of the robots are mainly offloaded to an intelligent Edge and Fog, while still envisioning their extension towards the cloud (cloud-to-things continuum). The proposed solution is supported by 5G connectivity, as the alternative to wired technologies able to meet the stringent networking requirements, and by a set of intelligence capabilities (e.g., task learning, prediction and optimization), as facilitators for enhanced automation and control. Its initial assessment is performed over a physical testbed, which not only validates the proposed solution but also showcases the introduced benefits.

The remainder of this paper is organized as follows. Section II overviews the exiting state of the art. In Section III, the solution for an Intelligent Edge-based Digital Twin for Robotics is presented, which is validated and assessed in a set of experiments over a 5G and 4G-enabled testbed in Section IV. Finally, conclusions and future research are presented in Section V.

II. Related Work

Cloud computing has been initially leveraged to cope with the computational-intensive tasks of a Digital Twin. The concept of a cloud-based Digital Twin platform is presented in [3], identifying entities, actors and the interactions among them. In [4], the authors proposed a cloud-based Digital twin architecture for cyber-physical systems. It focuses on the communication between the physical devices (mainly identified as sources of data), the digital devices in the cloud (providing non-time-sensitive analysis) and the digital and physical devices (control feedback from the digital to the physical world). Moreover, solutions that exploit the co-existence of fog, edge and cloud computing have been proposed. [5] proposes the applicability of fog, edge and cloud computing mapping them to each level of a Digital Twin, namely unit (e.g., devices), system (e.g., factories) and system-of-systems (e.g., interconnected factories), respectively. Following a similar approach, [6] proposes and implements a novel Digital Twin based data management framework for metal additive manufacturing. It uses the edge for life-cycle processes needing local real-time computation and the cloud to store historical data and uses analytics models to enhance the production efficiency.

5G is regarded as a key technology of the I4.0, and so of the Digital Twin. A comparison between the remote control of a Digital Twin under LTE and a proof-of-concept of 5G has been performed in [7]. However, it does not consider the use of edge and fog computing.

Finally, AI/ML can be applied to the huge and heterogeneous amount of data gathered by a Digital Twin. AI/ML algorithms can be used: (*i*) to improve the efficiency of fault diagnosis in manufacturing [8], both training the model in the virtual world (when no historical data is available) and doing predictive maintenance in the real world; (*ii*) to train an agent to infer the required actions to perform a task (e.g., humanoid robot lifting an object of unknown weight [9]); and (*iii*) to optimize network performances (e.g., using a Digital Twin of a network to train a deep learning algorithm optimizing the normalized energy consumption of users [10]).

Proponents of previous work already address, although individually, the different key technologies in the scope of Digital Twin. However, their implementation and validation in an environment comprising the cloud-to-things continuum (i.e., fog, edge and cloud computing), 5G connectivity and intelligence capabilities can provide a full assessment of the potential of the Digital Twin applications in I4.0 environments. In the following sections, this work aims to fill this gap.

III. Intelligent Edge-based Digital Twin for Robotics

In this section, the proposed solution for an Intelligent Edge-based Digital Twin for Robotics is presented, which not only offers a digital mirror of the physical robotic system but also enables its real-time control with zero perceived latency from a remote location. For that, the proposed solution relies on an Intelligent 5G Edge and Fog infrastructure comprising the following strands: (*i*) cloud-to-things continuum; (*ii*) 5G connectivity; and (*iii*) intelligence.

A. Intelligent 5G Edge and Fog Infrastructure

As depicted in Fig. 1, this work envisions a physical infrastructure where computing, networking and intelligence resources operate in a jointly and distributed fashion along the cloud-to-things continuum.

At the *computing* strand, it inherently assumes the availability of cloud resources that enables a ubiquitous, convenient and on-demand access to a shared pool of static computing resources, as well as its extension towards more distributed but less powerful resources available at the network edge to enable computational capabilities at the proximity of the data sources. Moreover, these concepts are extended further down towards the multitude of mobile and volatile terminals, which, due to their heterogeneity, have different capabilities in terms of storage, computing, communication, and power resources. The decision to offload and allocate virtual function across each computing substrates must consider latency, cost in terms of bandwidth and energy, and privacy concerns.

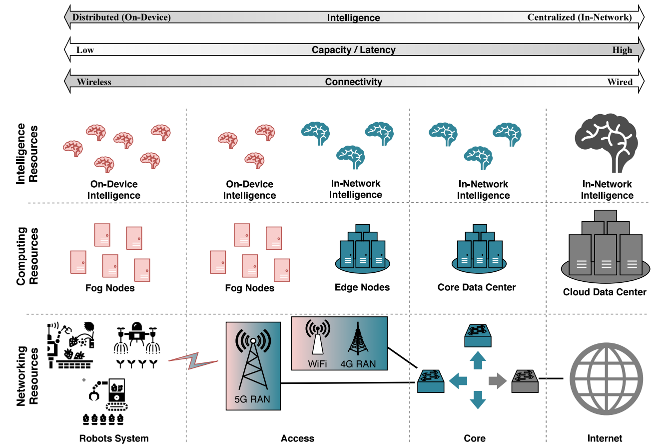

Fig. 1. Intelligent 5G Edge and Fog Infrastructure

At the *networking* strand, on one hand, wired technologies provide the best values for requirements such as low-latency, high-bandwidth and reliability, but they are limited in terms of flexibility, mobility and high-density connectivity, all of them required at the edge of the network. On the other hand, wireless technologies can meet the latter requirements, while the former can only be guaranteed when operating in interference-free or controlled environments (e.g., when operating in licensed bands, like cellular networks). Still, current cellular networks (i.e., 3G and 4G) cannot fulfill every requirement set by digital twinning. In this sense, 5G has positioned as the candidate solution to fulfill all the aforementioned requirements, not only introducing a new powerful radio solution but also network slicing, which evolves the current *one-size-fits-all* to a *custom-fit* paradigm where different virtual and logically isolated networks (i.e., network slices) are tailored to distinct requirements. Nevertheless, at the transport-level, connections mainly rely on wired technologies.

At the *intelligence* strand, intelligent and automated mechanisms (e.g., leveraging on AI/ML) are unleashed across different elements of the fog, edge and cloud computing substrates. In particular, from a distributed *on-device* intelligence that processes data where it is generated, up to an increasingly centralized and more powerful *in-network* intelligence as it moves towards the edge and the cloud. The envisioned intelligence capabilities must, therefore, be placed according to the level of privacy of the data and outcomes (e.g., federated

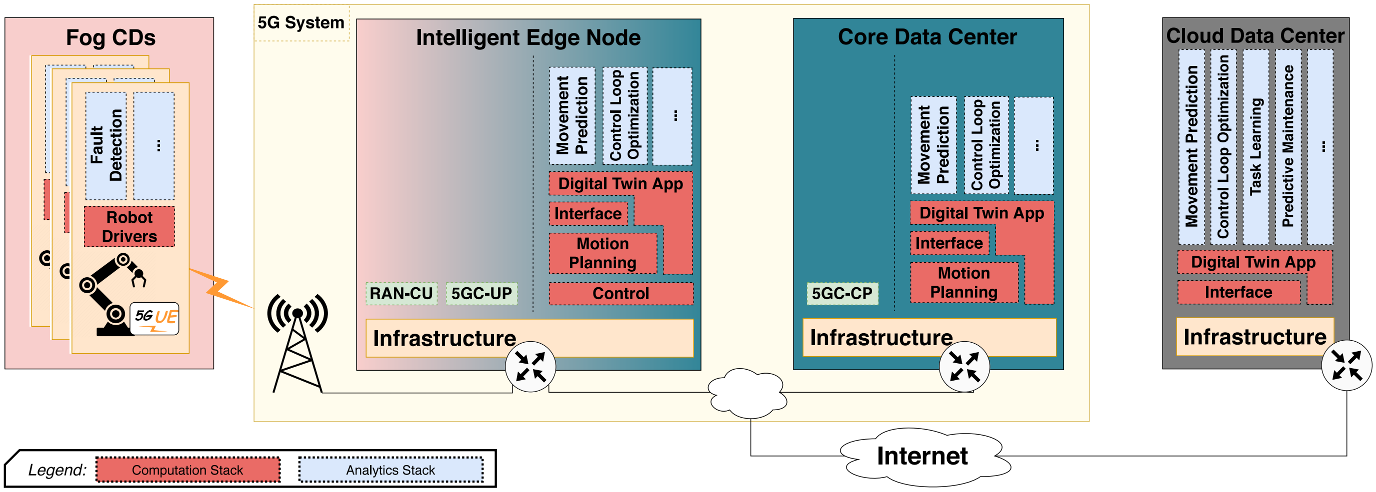

Fig. 2. Intelligent ROS-based Edge-based Digital Twin

learning [11]), the computational resources and communication requirements, and the level of shared functionalities.

B. ROS-based Edge Digital Twin

Robot Operating System (ROS) is a robotics middleware that provides common robotic functionality (e.g., read sensor data, navigation, planning, etc.) over general hardware abstraction using low-level device control. Such an approach not only makes available a common interface to interact with a different set of robots but also avoids vendor lock-in situations. Targeting for an open and generic solution that can be integrated with a variety of robotic systems, the proposed Intelligent Edge-based Digital Twin relies on ROS to interface with the underlying robots.

The proposed solution (as depicted in Fig. 2) is composed by the computation and analytics stacks, which modules are offloaded towards the edge and fog (and also towards the cloud), together with the connected data. By offloading the computation and analytics stacks from the robot to computing resources in the network, several benefits are envisioned: (*i*) simpler and cheaper robots; (*ii*) better cooperation and automation; (*iii*) scalability and reliability; and (*iv*) safer failover mechanisms and easier maintenance.

The modules that compose the computation and analytics stacks of the proposed Digital Twin are presented next:

*1) **Computation Stack**:* The computation stack defines the behavior of the robot, its operational states and the control flexibility. It is envisioned as various ROS modules that are distributed across the robot and the edge, core or cloud data center:

- The *Robots Driver* consists of the lowest-level module that directly interacts with the robot hardware and is responsible for: (*i*) making available sensor data and operational states to the other modules, and (*ii*) executing instructions or navigation commands received from the control module. Due to the tight relation with the robot's hardware, this module must be deployed in the robot.
- The *Control* module defines an abstraction that allows robot manipulation. It receives a navigation command and runs a control loop, following a given frequency, towards the *Robots Driver*. The loop is closed by the robot continuously sending-back the operational state. This module can only be offloaded to a close edge node, due to latency constraints.
- The *Motion Planning* module is responsible for finding inverse kinematics and building a path for the robot. The created path consists of a series of navigation commands that are sent to the *Control* module. This module can be placed in both edge node or core data center, in a balance between computing time and the network latency.
- The *Interface* module represents a high-level abstraction for the core motion planning functionalities. This module can be seen as the gateway between the operator and the robot manipulator. It can be offloaded to any computational resource from the edge up to the cloud.
- The *Digital Twin App* gives a human understanding of the connected data, by implementing the 3D models and control mechanisms. While the former provides a visual representation of the variations of the physical robot, the latter enables remote control and maintenance. Finally, it has the flexibility to be deployed in the user device (e.g., laptop, tablet), edge node, core or cloud data center.

It comes naturally that while moving up in the computation stack, the control loop increases. Consequently, hardware-specific optimizations can be achieved but the robot loses precision and reaction time. Computation modules that are more tolerant to latency can be placed in a third-party cloud, however, latency-sensitive modules must be placed in the edge or core data center.

*2) **Analytics Stack**:* After the data is collected, aggregated and transformed through the *Interface* module, the analytics stack comprises techniques (such as statistics, AI/ML and simulations) to monitor, simulate and predict the behavior of the physical robot. Subsequently, by interfacing with the *Digital*

Twin App, the modules of the analytics stack can suggest actions to control the physical robot or provide predictions.

- The *Movement Prediction* module focuses on navigation commands predictions to cope with the variability of the wireless medium (e.g., increased jitter or packet loss).
- The *Task Learning* module aims at teaching an AI-based agent how to dynamically execute tasks (e.g., sorting objects) without a human in the loop.
- The *Predictive Maintenance* module aims at predicting hardware failures of the robot components using maintenance data and suggesting when maintenance is required.
- The *Control Loop Optimization* module analyzes information from both the robot and network, along with other contextual information, to identify the best configuration parameters that provide the highest performance.

The previous modules can be distributed between the edge, core and cloud data center (i.e., in-network intelligence) since they operate on historical data and are resource-demanding with respect to training and inference tasks.

However, the proposed solution is flexible to also accommodate analytics modules on the robot itself (i.e., on-device intelligence).

- The *Fault Detection* module targets the detection of faults in the hardware components of the robot by correlating raw information exposed by the *Robot Drivers*.

Three type of interactions among the modules of the abovementioned stacks are envisioned (Fig. 3): (*i*) get state information from the physical robot; (*ii*) send actions to the physical robot; and (*iii*) analytics.

The *Control* module is the bridge component between the *Robot Drivers* and the remaining of the computation stack. All the modules of the computation stack use the *Control* module to get state information about the robot and/or send actions to the robot. However, the *Digital Twin App* can choose also to send actions to the robot via the *Motion planning* or *Interface* modules. The difference resides on the levels of abstraction provided by each module, as higher-level modules offer higher abstractions of the underlying robotic system. Lastly, the analytics interaction gets the state information, stores it for analysis and recommends an action/prediction that the *Digital Twin App* executes.

In particular, the *Interface* module offers a custom-made interface (e.g., Python API or REST-based), while the interaction with the *Robot Drivers*, *Control* and *Motion Planning* modules is done using the ROS communication protocols. ROS provides a publish-subscribe platform to exchange topic information between different nodes which, once registered, communicate via configurable topics in a peer-to-peer fashion. Lastly, the *Digital Twin App* makes use of custom-made interfaces to interact with the analytics stack modules.

IV. Validation Results

This section provides an assessment of the proposed Intelligent Edge-based Digital Twin for Robotics, through its implementation and instantiation targeting a ROS-based robotic manipulator.

A. Experimentation Setup

The experimentation setup (depicted in Fig. 4) used for this evaluation is built over the 5TONIC Laboratory, located in the IMDEA Networks Institute (Madrid, Spain). The robotic system is composed of both physical and simulated instances (running on a MiniPC) of the Niryo One robotic manipulator, a 6-axis robotic arm powered by ROS. Due to the lack of 5G and 4G interfaces on these devices, they are connected via Ethernet to two CPEs (HUAWEI 5G CPE Pro Baloong 5000 and HUAWEI B315s-22, respectively) that provide connectivity towards the 5G and 4G networks available at 5TONIC. 5TONIC provides an implementation of 5G NSA (BB630 baseband and Advance Antenna System AIR 6488) and 4G networks (BBU5216 baseband and RRU 2203 with integrated antenna) provided by Ericsson. Finally, an edge data center and a more distant cloud data center (emulated through a virtual machine that introduces an artificial delay of 7ms) host the required virtual entities.

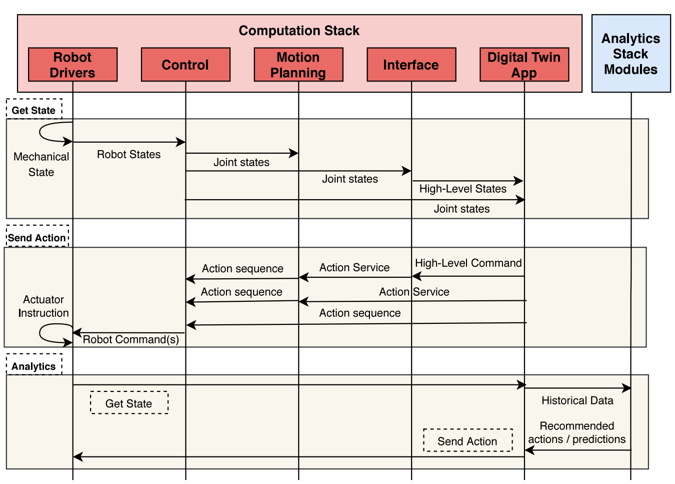

Fig. 3. ROS-based Digital Twin Workflow

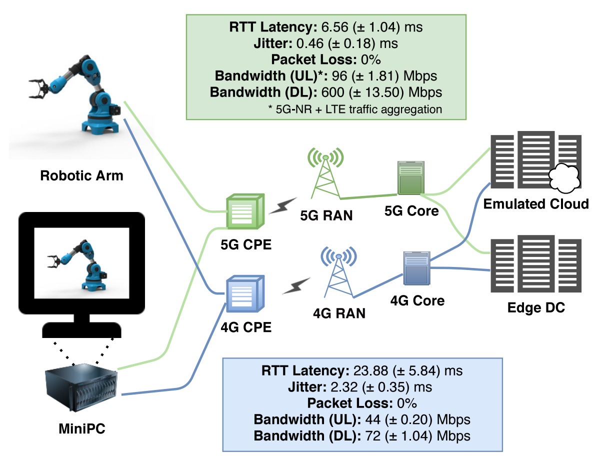

Fig. 4. Experimentation Setup at 5TONIC Laboratory

B. Experiment 1: Digital and Physical Synchronization

This experiment aims to evaluate the synchronization between the physical and digital instances, by measuring their distance offset over time. The physical Niryo One robotic manipulator is controlled by a remote *Digital Twin App* that is continuously sending instructions (in a loop) to the robot and receiving its pose on each instant. Such experiment is performed for a different combination of the following configurations: (*i*) 5G vs 4G (Net); (*ii*) *Digital Twin App* located on the edge vs cloud (DT); and (*iii*) computation stack located on the Robot vs offloaded to the edge (Comput). In doing so, results were measured for five different configurations.

Fig. 5 shows the results for the five configurations considered (expressed as a 3-tuple *Net-DT-Comput*), namely the position of both robotic arm and its digital twin replica over time (bottom) and the synchronization accuracy (top).

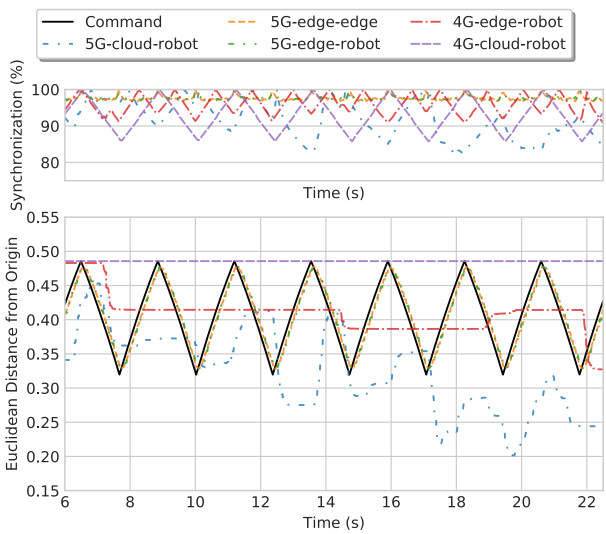

Fig. 5. Synchronization between the Digital Twin and the Physical Robot

Considering as the baseline configuration (i.e., state of the art solution) having the full computation stack in the robot, results show that the *Digital Twin App* is able to achieve a synchronization of 97.72±0.65% when leveraging on 5G connectivity (i.e., *5G-edge-robot*). An error on the synchronization is witnessed because the Niryo One Robot has a control loop of 20 ms, meaning that any update on the pose is only sent to the *Digital Twin App* on the next control loop after receiving the instruction. When relying on 4G connectivity (i.e., *4G-edge-robot*), the synchronization decreases down to 95.40±2.88%, making the robot to move erratically. Since the latency is bigger than the robot's control loop, instructions which lifespan is expired are discarded by the robot.

In fact, when using 4G, the computation stack cannot be offloaded to the edge due to the latency witnessed in the link. When 5G is considered, it is possible to offload the computation stack to the edge (i.e., *5G-edge-edge*), while maintaining the same performance levels between the physical and the digital replica (i.e., synchronization of 97.66±0.67%).

Finally, when considering the *Digital Twin App* in the cloud (i.e., *5G-cloud-robot* and *4G-cloud-robot*), the high latency caused unpredictable movements to be executed by the robot (when using 5G) or to not move at all (when using 4G), as expired instructions are discarded by the robot.

C. Experiment 2: System Load

This experiment aims to compare the loading limits for both 5G and 4G and to identify when the Digital Twin starts to misbehave. A simulated robot manipulator is used for this experiment, which receives continuous instructions from the *Digital Twin App*. At the same time, different traffic loads (i.e., 5Mbps, 10Mbps, 25Mbps, 50Mbps, 75Mbps and 100Mbps) are injected at the uplink of the wireless link. Additionally, to overcome the misbehavior of the Digital Twin when using 4G, the control loop of the robot was increased to 30ms and only the *Digital Twin App* is running on the edge.

Fig. 6 depicts the synchronization accuracy between the robotic arm and its digital replica for different system loads.

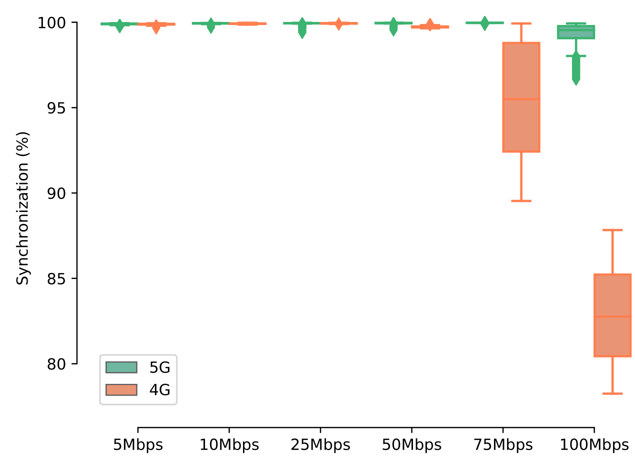

Fig. 6. System Load with Background Traffic

When the system load is below the saturation point of the link, the *Digital Twin App* does not suffer any disruption. However, as the system load gets closer to the saturation point, an impact on the synchronization between the physical and digital robot can be witnessed (50Mbps and 100Mbps for 4G and 5G, respectively). With its current implementation, 5G already provides an important enhancement in terms of bandwidth with respect to the 4G. With the implementation of 5G SA, together with the slicing of the radio access network (RAN) [12], it will be possible to define advanced QoS policies. In this way, the digital twin delay-sensitive traffic (URLLC type) can be prioritized over the background traffic (eMMB type), ensuring the correct operation of Digital Twin even under link saturation.

D. Experiment 3: Adaptive Control Loop Configuration

This experiment aims to demonstrate the capability of autonomously adapt the configuration of the control loop, so that the Quality of Experience (QoE) when using the Digital

Twin is not impacted by changes in the network conditions. In this experiment, the *Digital Twin App* is continuously issuing instructions towards the robot while introducing an additional delay in the link between both end-to-end entities. A *Control-Loop Optimization* module, as part of the analytics stack, is implemented which continuously monitors the RTT latency and, based on the measured values, re-configures the control loop of the robot (to avoid instructions to be discarded as explained in Experiment 1).

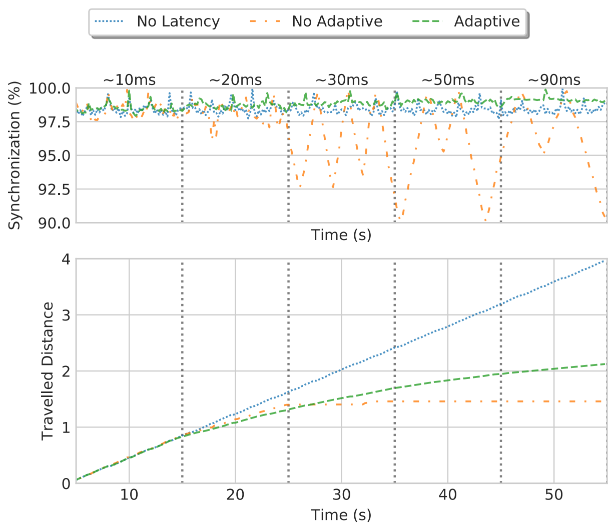

Fig. 7. Adaptive Control Loop

Fig. 7 presents the comparison between the normal operation of the proposed Digital Twin (i.e., with a fixed control loop of 20ms) and its operation when implementing the control loop optimization mechanism while increasing the RTT latency. Moreover, it presents the behavior if no latency is introduced. When using the optimization mechanism, the *Digital Twin App* maintains its synchronization with the physical robot, even when the latency is increased up to 90ms. However, at the cost of having slower movements executed by the robot (green line). In opposite, without the proposed mechanism, the robot moves slightly faster but with a lower synchronization (i.e., latency of 20ms), stopping the execution of any command when the latency is above 30ms (orange line).

Although the current implementation of the *Control-Loop Optimization* module is using a threshold-based mechanism, it can be enhanced with more advanced AI-based agents and considering a combination of different parameters for decisions (e.g., latency, packet loss and jitter). As the complexity increases, so do the required resources to compute the decisions. Thus, these modules can benefit from the computational resources available at the edge (and beyond).

V. Conclusion and Future Work

This paper presents a solution for an Intelligent Edge-based Digital Twin for Robotics, converging the cloud-to-things continuum, 5G and intelligence technologies with the Digital Twin paradigm. The proposed solution is then implemented and validated in an E2E scenario through a set of experiments aiming to assess the benefits introduced by the aforementioned technologies. Results validate the proposed solution, highlighting the feasibility of offloading the robot's modules to the edge and the importance of the 5G to fulfil the networking requirements. Moreover, it is showcased that intelligence can play a key role to ease automation and optimization (e.g., by reconfiguring the robot's parameters) in order to guarantee the best performance at each instant.

For future research, the considered modules of the analytic stack are going to be implemented and evaluated. Also, the solution is going to be integrated into a Technology Readiness Level (TRL) 5/6 environment, where multiple robotic systems and other types of services need to coexist simultaneously.

Acknowledgments

This work has been (partially) funded by H2020 EU/TW 5G-DIVE (Grant 859881) and H2020 5Growth (Grant 856709). It has been also funded by the Spanish State Research Agency (TRUE5G project, PID2019-108713RB-C52PID2019-108713RB-C52 / AEI / 10.13039/501100011033)

References

[1] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[2] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, 2019.

[3] K. Borodulin, G. Radchenko, A. Shestakov, L. Sokolinsky, A. Tchernykh, and R. Prodan, "Towards Digital Twins Cloud Platform: Microservices and Computational Workflows to Rule a Smart Factory," in *Proceedings of The10th International Conference on Utility and Cloud Computing*, ser. UCC '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 209–210.

[4] K. M. Alam and A. El Saddik, "C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[5] *Modeling of Cyber-Physical Systems and Digital Twin Based on Edge Computing, Fog Computing and Cloud Computing Towards Smart Manufacturing*, ser. International Manufacturing Science and Engineering Conference, vol. Volume 1: Additive Manufacturing; Bio and Sustainable Manufacturing, 06 2018.

[6] C. Liu, L. Le Roux, C. Körner, O. Tabaste, F. Lacan, and S. Bigot, "Digital Twin-enabled Collaborative Data Management for Metal Additive Manufacturing Systems," *Journal of Manufacturing Systems*, 2020.

[7] P. Isto, T. Heikkilä, A. Mämmelä, M. Uitto, T. Seppälä, and J. M. Ahola, "5G Based Machine Remote Operation Development Utilizing Digital Twin," *Open Engineering*, vol. 10, no. 1, pp. 265 – 272, 2020.

[8] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning," *IEEE Access*, vol. 7, pp. 19 990–19 999, 2019.

[9] I. Verner, D. Cuperman, A. Fang, M. Reitman, T. Romm, and G. Balikin, "Robot Online Learning Through Digital Twin Experiments: A Weightlifting Project," in *Online Engineering & Internet of Things*, M. E. Auer and D. G. Zutin, Eds. Cham: Springer International Publishing, 2018, pp. 307–314.

[10] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn from a Digital Twin," 2019.

[11] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019.

[12] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN Slicing for Verticals: Enablers and Challenges," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 28–34, 2019.