RESEARCH ARTICLE

Toward 5G Edge Computing for Enabling Autonomous Aerial Vehicles

GERASIMOS DAMIGOS[1], **TORE LINDGREN**[1], **AND GEORGE NIKOLAKOPOULOS**[2], **(Member, IEEE)**

[1]Ericsson Research, 977 53 Luleå, Sweden
[2]Robotics and AI Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden

Corresponding author: Gerasimos Damigos (gerasimos.damigos@ericsson.com)

ABSTRACT Offloading processes responsible for a robot's control operation to external computational resources has been in the spotlight for many years. The vision of having access to a full cloud cluster for any autonomous robot has fueled many scientific fields. Such implementations rely strongly on a robust communication link between the robot and the cloud and have been tested over numerous network architectures. However, various limitations have been highlighted upon the realization of such platforms. For small-scale local deployments, technologies such as Wi-Fi, Zigbee, and blacktooth are inexpensive and easy to use but suffer from low transmit power and outdoor coverage limitations. In this study, the offloading time-critical control operations for an unmanned aerial vehicle (UAV) using cellular network technologies were evaluated and demonstrated experimentally, focusing on the 5G technology. The control process was hosted on an edge server that served as a ground control station (GCS). The server performs all the computations required for the autonomous operation of the UAV and sends the action commands back to the UAV over the 5G interface. This research focuses on analyzing the low-latency needs of a closed-loop control system that is put to the test on a real 5G network. Furthermore, practical limitations, integration challenges, the intended cellular architecture, and the corresponding Key Performance Indicators (KPIs) that correlate to the real-life behavior of the UAV are rigorously studied.

INDEX TERMS 5G, edge computing, robotics, UAV.
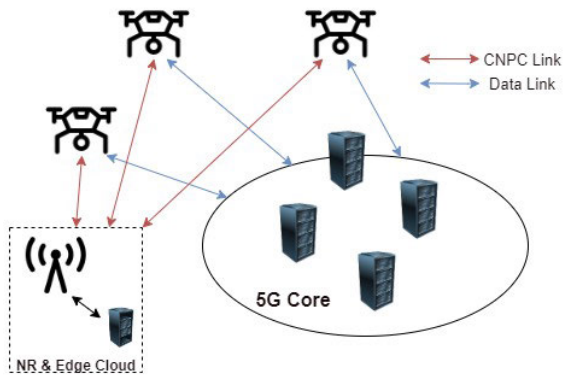
I. INTRODUCTION

A. BACKGROUND & MOTIVATION

For a long time, there has been an increasing interest in unmanned aerial vehicles (UAVs) in both research and industry communities. UAVs offer an extensive number of capabilities, such as easy deployment, high maneuverability and range, as well as the ability to carry various types of payloads for different needs. In many cases, a reliable real-time control channel for the remote operation of a UAV or large computational resources on board the UAV is required. Researchers and organizations are looking into the manifestation of these capabilities, either by improving communication technologies or by focusing on optimizing

The associate editor coordinating the review of this manuscript and approving it for publication was Di Zhang.

algorithms that consider the available onboard resources. In recent years, a consideration for tackling both problems in a common architecture has emerged.

A possible solution would be the combination of a large external computational resource in synergy with a reliable communication system to enable the real-time transmission of sensor data. Numerous organizations have considered the possibility of using application servers close to the sensing-platform or an agent for offloading computational-demanding tasks. Such servers are typically called *Edge servers* and this term usually expresses the characteristic of those servers to be as close as possible to the corresponding agent. This practice primarily serves the latency requirements of the corresponding applications [1]. Having an edge server performing computationally demanding tasks enables modest platforms to perform more complex tasks, and

FIGURE 1. Baseline network architecture of cellular UAV communication. Payload (data link) vs. non-payload (CNPC) communication.

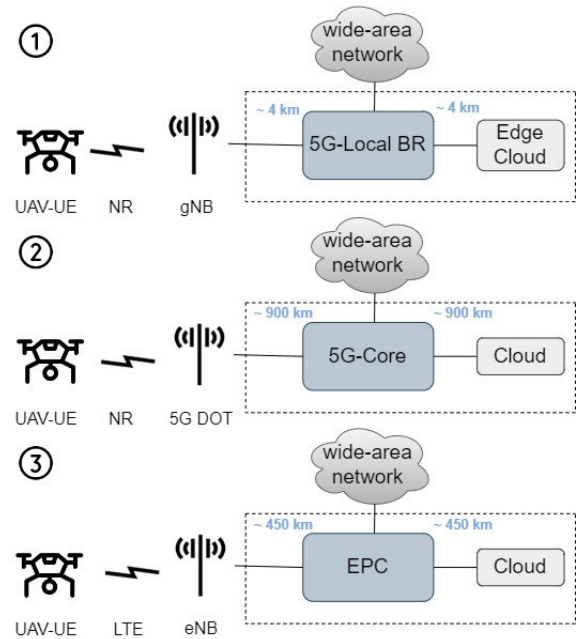**FIGURE 2.** Three different network topologies for the used experimental setup.

offers the possibility of exploring other cloud technologies. For example, distributed processing, containers, container orchestrators, etc. However, having external computational resources for serving real-time applications requires a reliable communication system. In this study, the offloading of a time-critical control operation for an unmanned aerial vehicle (UAV) using cellular network technologies was evaluated and demonstrated experimentally.

Using a cellular network, and in particular a 5G network, for communication with UAVs introduces a number of new possibilities and features to the field. Higher data rates, lower latency, increased coverage, and the potential use of the 5G Quality of Service (QoS) features are some of the most notable [2]. Furthermore, 5G networks enable the use of remote technology in time critical applications. Remote medicine applications [3], [4], the Industry 4.0 paradigm [5], [6], 5G connected vehicles [7] and 5G connected Internet of Things [8] are some noteworthy topics that are expected to significantly benefit from 5G networks. 5G connected UAVs and unmanned aerial systems (UAS) are also examples of such applications [2].

Another notable benefit when considering serving UAVs over cellular networks would undoubtedly be the utilization of the installed infrastructure and the coverage benefits that wide area cellular technologies offer compared with local area wireless network solutions. Applications such as UAV-assisted power line inspection, harbor inspections, and other long-range demanding applications are expected to benefit from the utilization of cellular networks.

Usually, in conventional cellular networks, the application servers are located in the cloud; thus, communication with a UAV suffers from latency effects [9]. With the edge server option cellular network providers have the ability to collocate the edge server with a base station (BS), thereby avoiding wide area network routing and significantly reducing the latency of the system.

UAV communications are generally categorized into two groups: control and non-payload communications (CNPC) and payload communications. The baseline architecture is shown in Fig. 1. The CNPC links are responsible for

the control and operation of UAVs. In order to maintain safe and reliable flight operations, they must satisfy a stringent set of requirements. Many studies have indicated rough requirements regarding the CNPC and the payload communication links [2], [10]. Besides the CNPC link, many studies consider UAVs as a sensing platform and design their application around the payload communication link. During that definition, other vital indicators revolve around the throughput and the Age of Information (AoI) metrics and address various complex problems. These indicators are often utilized together; a sample of the exciting and promising problems in optimizing communication schemes would include path planning scenarios, where the AoI, throughput, round trip packet latencies, and energy consumption are dominant factors in solving these problems [11], [12].

In this paper, the autonomous operation of a 5G enabled UAV is presented and operated by utilizing an edge server solution. The focus is on comparing the performance with that of similar architectures. The architecture of each scenario is illustrated in Fig. 2. We name the three architectures as 1) *Local 5G & Edge Cloud*, 2) *5G with remote core & Cloud* and 3) *Public LTE & Cloud*.

B. RELATED WORK & CONTRIBUTION

Using external computational resources, while utilizing a variety of communication systems to connect a robotic agent to a cloud server, has been thoroughly examined in the past. Zeng et al. [2] presented a comprehensive article describing both 5G enabled UAVs, as well as UAVs serving as mobile network nodes, and discussed related aspects of essential requirements, they also described the radio aspect of such

systems. Voigtländer et al. [13] implemented one of the first 5G enabled closed loop robotic systems and stated the possibility of using such systems in real-life applications.

Further, Wu et al. [14] presented a cellular connected UAV experimentally comparing two edge architectures, while also considering and modeling the latency aspect of the system, which was subsequently used in the analysis of the phenomenon for the control of the system. Nevertheless, the presented study did not capture the correlation of the presented delays in the behavior of the UAV.

In [15], Kumaras et al. presented a similar solution to ours, placing an autonomous UAV's command and control component in the edge server. As a result, the authors achieved one of the very first successful autonomous flights of a UAV utilizing the 5G network and an edge server. The proposed solution utilized a compact 5G network, where the core of the network was deployed on a local computer, and the radio interface was deployed using a Software Defined Network (SDN). However, an in-depth analysis of the proposed architecture was not captured (such as the operational frequency of the offboard controller). More specifically, the uplink and downlink latencies, the jitter KPI and the correlation of the presented KPIs with the acquired behavior of the UAV was not captured.

Similar to [15], Bekkouch et al. [16] presented an architecture that places the offboard controller on the edge server of a 5G network. The authors captured the round-trip latency for the control and command operation of an emulated UAV. Additionally, they correlated the round-trip latency to the simulated behavior of the UAV. However, they assumed that considering an edge server would decrease the latency to approximately zero. Whereas in an uncongested real-life 5G network with a 5G-enabled UAV, the latency presented in the round trip time between the edge server and the considered UAV is a major factor in the behavior of the UAV.

Recently, Taleb et al. [17] proposed a complete framework for enabling immersive services in an enhanced teleoperation UAV case, also considering a real-life 5G network. This work examined various KPIs describing the system's performance and excellent in-depth analysis. However, since the study utilized a teleoperation scheme and placed the UAV controller onboard the UAV, an autonomous scenario was not considered, and the correlation with the flying behavior of the UAV was not captured.

Finally, Markopoulos et al. [18] presented a field test study where a 5G-enabled UAV is controlled over a 5G network. Similar to [15], the authors utilized an edge server to generate the path that the UAV should follow and send it to the UAV over the 5G interface. The study captures different capabilities in the uplink and downlink direction of the 5G interface in correlation with the distance from the base station and the corresponding signaling conditions. In addition, a complimentary latency measurement is provided with an average value of $\sim 30\ msec$. However, this article did not use a UAV controller offboard the UAV and did not capture other additional KPIs, like the jitter measurements, downlink and

uplink latencies, and the system's behavior in high network load or cell load conditions. Hence, a low-level architecture for the offloading of excessive computationally demanding time-critical controllers (like optimization-based controllers) over a 5G interface still needs to be added to the literature. Also, the real-life capabilities and limitations of a corresponding 5G-enabled UAV should be defined. However, the available literature lacks a quantitative real-life experimental analysis of a *5G enabled UAV - Edge server system*, an in-depth described architecture and the corresponding modeling and correlation of the presented delays to the behavior of the 5G-enabled UAV.

In this study a novel architecture for a 5G enabled UAV is proposed and evaluated experimentally and as such the contributions of this study can be summarized as follows:

- We propose a novel and reliable architecture for the integration of an UAV autonomous operation with 5G mobile networks considering that it is fully embedded and implemented in the Robotic Operating System (ROS).
- We prove resilience of the proposed novel architecture through extensive experimentation and evaluation in terms of the closed-loop latency requirements and the reliability of the autonomous system, including all the intelligence of the 5G enabled UAV architecture that it is implemented in an edge server.
- A list of unexplored potential performance improvements that can be applied to a 5G enabled UAV-Edge system considering the findings of this research is proposed. The real-life results and measurements are also displayed with the intention of serving as a baseline for future quantitative reasoning on the design of autonomous UAV missions over a 5G - Edge architecture.
- We introduce KPIs and evaluate the proposed architecture in comparison with other available solutions and different 5G networks, while we consider for the first time ever the use case of varying obstacles, which makes the overall mission very time critical.

The remainder of this article unfolds with the following sections. In Section II, the system design and the proposed architecture are presented. In Section II-C, the expected behavior of a closed control system when running on multiple machines over a real cellular network is presented, while Section III, presents the experimental findings of this study. Finally, in Section IV, further improvements for a 5G enabled autonomous UAV are discussed.

Abbreviations: The abbreviations used throughout the paper are summarized in Table 1.

II. SYSTEM DESIGN & ARCHITECTURE

A system consisting of a mobile robot (UAV) and a major computational unit, the edge server, which is connected to the UAV via 5G, was developed. For this test, a time-critical autonomous robotic application was evaluated in terms of its latency requirements, while considering alternative

UAV	Unmanned Aerial Vehicle
5G	5th Generation
4G LTE	4th Generation Long Term Evolution
CNPC	control and non-payload Communication
GCS	Ground Control Station
QoS	Quality of Service
UAS	Unmanned Aerial Systems
BS	Base Station
ROS	Robotic Operating System
MPC	Model Predictive Control
NMPC	Non-linear Model Predictive Control
CC	Control Computer
NR	New Radio
SA	Stand Alone
VM	Virtual Machine
KPI	Key Performance Indicator
EPC	Evolve Packet Core
ISM	Industrial, Scientific, and Medical
APN	Access Point Name
URLLC	Ultra Reliable Low-Latency Communication
IP	Internet Protocol
UE	User Equipment
NAT	Network Address Translation
VPN	Virtual Private Network
RTT	Round Trip Time
3D	3 dimensional
CA	Collision Avoidance
CCDF	Complementary Cumulative Density Function
ROS 2	Robotic Operating System, version 2
AoI	Age of Information
UDP	User Datagram Protocol
TCP	Transport Protocol Layer

architectures (i.e., changing the communication interface and the overall architecture from a local 5G with a local core breakout, to 5G with a remote 5G core, to a public LTE). The experiment consisted of a UAV executing a circular trajectory at a constant height and evaluating its behavior using a controller operating at a frequency of $40\,Hz$. On the UAV side of the application, the state representation of the UAV is sensed via a VICON motion capture system [19]. The position, orientation, and velocity of the agent are then sent to the edge server. The control commands are generated in the edge server and indicate the trajectory of the UAV, while they are sent back over the 5G setup. Various architectures and experiments have been performed to test the low-latency requirements and reliability of this UAV-Edge proposed closed-loop control link.

A. SYSTEM DESIGN & ARCHITECTURE
To design the proposed architecture, multiple topologies were studied, considered, and evaluated for the execution of the considered application. To make a comparison possible between similar architectures, a baseline system that was later modified to constitute the manifestation of different architectures was designed. A UAV with limited computational resources takes-off from a set point, executes an autonomous mission, and performs a safe landing operation. This autonomous mission consist of the UAV executing a circular trajectory at a constant height. The

entire autonomy of the UAV takes place at a remote server where a Model Predictive Control (MPC) controller [20] is implemented and is fully responsible for the operation of the UAV.

In Fig. 3, the baseline architecture of the proposed system is illustrated. Consider the block that is abbreviated as *UAV abstraction* to be one UAV. The reason for this abstraction is that the UAV in use is a Crazyflie 2.0 [21], which possesses limited computational resources and lacks the ability to carry payloads exceeding $15\,g$. The rationale behind the use of a small-scale UAV is to showcase the ability to utilize the 5G network and the external computational resources of the edge server to fully support UAVs that cannot carry extensive computational resources onboard. This practice is meant to enable small UAVs with the ability to use the full potential of the edge cloud. Additionally, when excluding large computational resources from the UAVs, the opportunity arises to equip them with additional sensors that can enable fully autonomous operations, even in GNSS-denied environments. The UAV communicates with a radio connection, Radio 1, using a control computer that simulates the actual onboard computer of a UAV. This control computer (CC) hosts the 5G router, where communication with the 5G Ericsson Radio DOT is established. The latter radio communication is illustrated with the 5G New Radio (NR) interface. From this DOT base station, data traffic is sent to the local breakout node of this 5G stand alone (SA) network. Note here that in most cellular network architectures, in order for data traffic to reach the final destination it has to route from the core of the network, or similarly from an entity that imitates the core of the network, for example, the local core breakout. Here, the local breakout imitates a local version of a 5G core, and is responsible for routing packets of data to their destination and implementing other essential functionalities. The use of a local breakout is essential in latency-sensitive applications, as it mitigates extended data routing. The data destination is a Linux virtual machine (VM) hosted at the edge cloud application server of this 5G innovation network in the premises of the Luleå University of Technology. This Linux VM hosts the entire MPC operation in all three scenarios considered in this study. Subsequently, after each cycle of the MPC controller, the computed control commands follow the reverse path to the *UAV abstraction* node responsible for the operation of the UAV dynamics. The whole application was implemented in ROS. The ROS framework supports either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) based communication. After initial pilot experiments, the authors tested the framework rigorously, and the results between TCP and UDP were almost identical. Therefore, even though UDP is universally accepted as the desired protocol to transmit data in specific use cases, e.g., command and control data in teleoperation scenarios, instead, TCP was selected as the transmission protocol. The authors selected TCP for the reason that it exhibited identical performance to UDP. Further, TCP allows the authors to monitor the
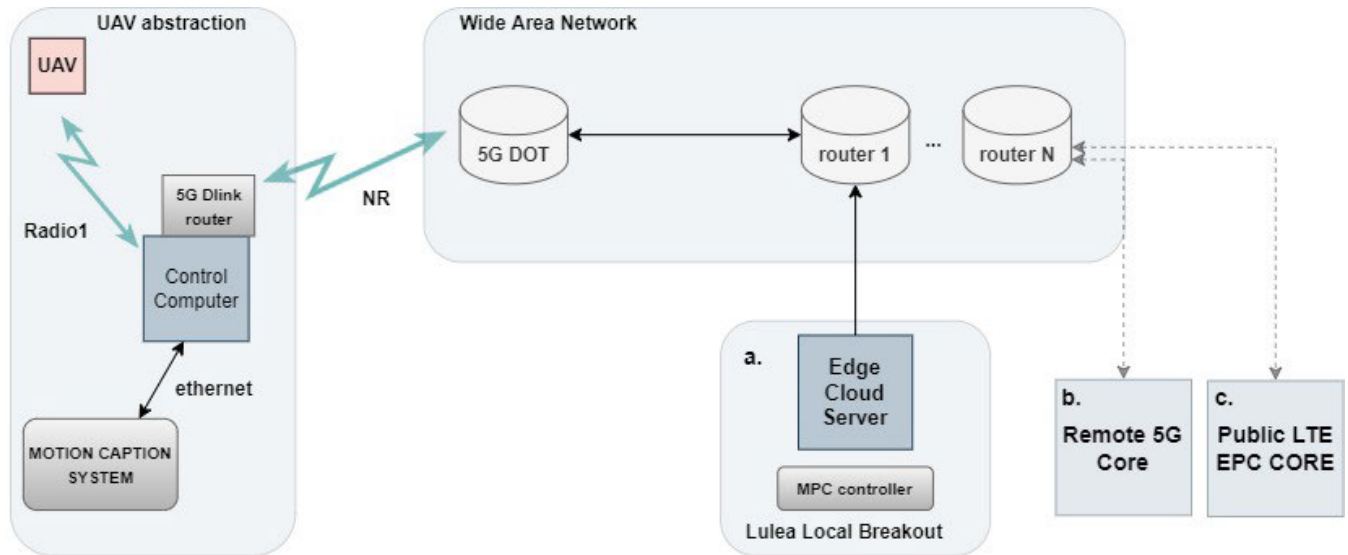
FIGURE 3. System architecture for the UAV communication over 5G to the edge server. For the *a.* scenario, the *Local 5G & Edge Cloud* is used and the MPC controller is hosted on the edge cloud server. For the *b.* scenario, the *5G with remote core & Cloud*, all the control loop data has to pass through the remote 5G core. For the *c.* scenario, the *Public LTE & Cloud*, all the control loop data has to pass through the public LTE core.

congestion behavior of the network. More specifically, the advantage of TCP in the considered framework is that when congestion conditions occur, the TCP congestion control mechanism is triggered, and that is directly captured in the latency measurements of the system. There is no congestion control in UDP. Additionally, the use of UDP or TCP is commonly known in the UAV community and the MAVLink protocol [15], [16], [17], so the proposed solution remains consistent with the alternatives in the transport network layer.

In this study, the challenges of the presented architecture were examined and important Key Performance Indicators (KPI) that make autonomous operations possible, reliable, and robust for a 5G enabled UAV are evaluated. To conclude this subsection, alternative architectures that are considered for comparison purposes are presented. As described above and illustrated in Fig. 2, to examine different scenarios we consider the modification of three main components. The first component is the NR radio interface, the second is the edge server, and the third is the local breakout. The second architecture paradigm swaps the local breakout with a remote 5G core and the edge server with a cloud server. The fundamental difference in this change is that the 5G core is located $\sim 900\,km$ away from the DOT base station. This corresponds to a significant increase of in-between devices or router hops and is expected to severely affect the latency performance of the architecture. The *third* option is similar to this solution, where the NR radio interface was swapped with LTE, the edge server with a cloud server and the local breakout with the Evolved Packet Core (EPC) of the system. Here, the EPC is estimated to be located approximately $\sim 450\,km$ from the public BS and is responsible for routing all sensed data to their destination. For this implementation,
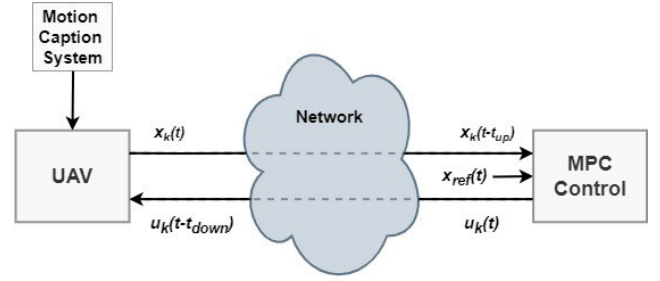
the load of the public network is expected to be a dominant factor of the system's performance.

Finally, in Fig. 4, a map of Sweden and the corresponding locations are shown in order to illustrate the spatial separation aspect of the core and cloud of the networks for each described scenario. An example of the data traffic for the local 5G case is shown. The path of each data packet is illustrated in the enlarged portion of the figure. In most cellular networks, the data has to pass through the core of the network. The core of the network points to the final destination of each packet. Thus, for the considered scenario, the path for each corresponding round trip time (RTT) measurement is described as follows: 1) a sensor data packet departs from the UAV abstraction and arrives at the *acting core* of the network, which is a *5G local core breakout*. 2) It is routed to the destination of the packet that is, the edge cloud. The edge cloud is located in the same premises as the experiment site. 3) After all the processing that takes place in the edge cloud, the corresponding command for the considered packet is produced and then sent back to the UAV abstraction. The command packet departs from the edge cloud and reaches back to the core of the network. 4) The core of the network indicates the final destination of the control command packet which is the UAV abstraction. It is worth noting that for the remaining two scenarios, the traffic has to go through the other corresponding cores of each network. For the *Public LTE & Cloud* scenario data has to go through the *EPC core*, and for the *5G with remote Core & Cloud* scenario, data has to go through the *remote 5G core*. The computing destination is the same for all three of the considered scenarios. It is the Linux VM located in the premises of Luleå University of Technology. The physical distance of the cloud server in the latter two described scenarios constitutes this computational
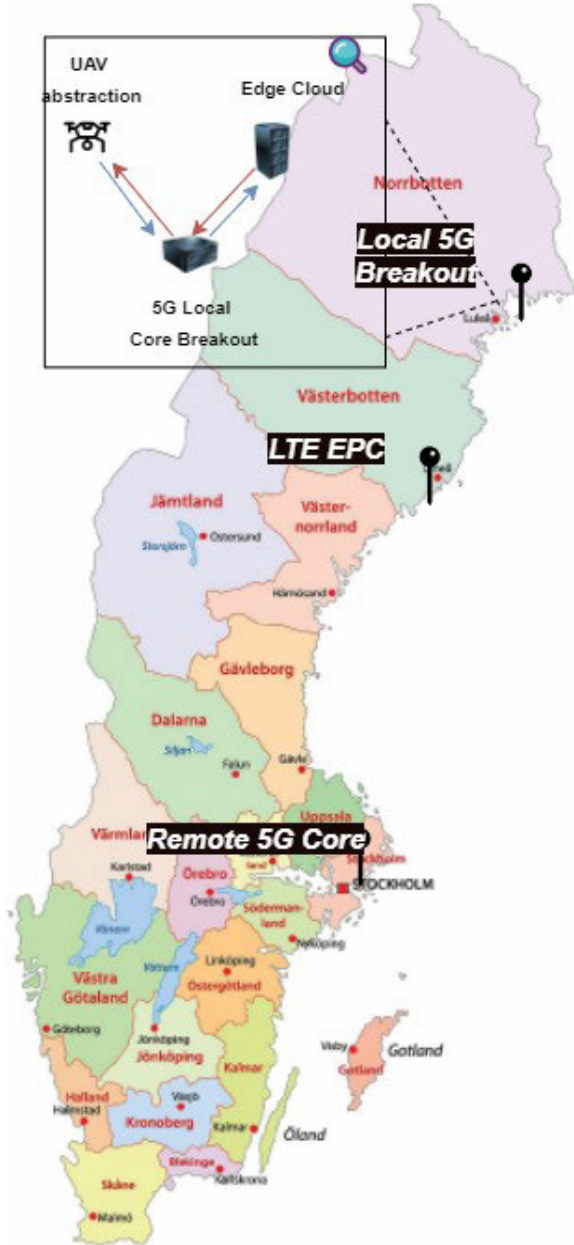
FIGURE 4. Spatial separation of the remote 5G Core at Stockholm, Kista, the Luleå Local Breakout and the 4G LTE EPC core on Sweden's map. Note here that in most cellular communication architectures data traffic has to go through the core of the network. An example is illustrated for the *Local 5G & Edge Cloud* architecture but the same reasoning holds for the remaining two, herein illustrating the benefits of an edge based solution. The three pins on the map illustrate the location of the corresponding core of the network.

FIGURE 5. Edge server oriented proposed closed loop control architecture.

B. DEVICES & HARDWARE

As briefly described above, the UAV is a Crazyflie 2.0, and more information can be found in [21]. This UAV has limited computational resources and communicates with the CC utilizing a Crazyradio PA antenna [22], which operates under low latency and at $2.4\,GHz$ Industrial, Scientific, and Medical (ISM) band radio [22]. The 5G router that was used for communicating over the 5G innovation network is a D-Link DWR-2101 5G [23]. The entire 5G network runs over $3.7\,GHz$, and supports a 5G stand alone version, which is used in this study. This 5G innovation network offers two options for connecting to the 5G core. The selection between the two is established through different Access Point Names (APN), either to the local breakout server at Luleå center ($\sim 4\,km$ away from the experiment site), or to the primary remote 5G core in Kista, Stockholm ($\sim 900\,km$ away). Considering the travel distance to Kista, as well as traditional network delays in internet protocol (IP) based networks, the data packet delay for traveling to Kista and back is significant for time-critical applications. The aforementioned observation will be further discussed in sections II-D and II-E.

C. CLOSED LOOP CONTROL ARCHITECTURE

For the closed loop control of the UAV a Model Predictive Control approach was adopted, as shown in Fig. 5. MPC is a popular choice for researchers working on UAVs. Many articles in the literature use MPC controllers for a UAV platform. Therefore, the selected MPC is briefly described, and the reader is referred to the full article [20]. Here the UAV is presented as a robot with six degrees of freedom and a fixed body frame.

The body frame and the global frame are denoted as $\mathbb{W}$ and $\mathbb{B}$ respectively, whereas the kinematics of the UAV are shown in (1).

unit to be considered as a **cloud server**, while for the *Local 5G & Edge Cloud*, the computational unit can be considered as an **edge cloud**, even if it is not located in the same premises as the *5G local core breakout*. An additional improvement for every described architecture would be the placement of the computational unit to the same premises of the core of each network. Hence the physical distance and the extra additional traffic hops are reduced.

$$\dot{p}(t) = v_z(t)$$

$$\dot{v}(t) = R_{x,y}(\theta,\phi)\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} u(t)$$

$$\dot{\phi}(t) = \frac{1}{\tau_\phi}(K_\phi \phi_{ref}(t) - \phi(t))$$

$$\dot{\theta}(t) = \frac{1}{\tau_\theta}(K_\theta \theta_{ref}(t) - \theta(t)) \qquad (1)$$

The position of the UAV is denoted as $p = [p_x, p_y, p_z]^T$ and the linear velocity is denoted as $v = [v_x, v_y, v_z]^T$. A rotation matrix that describes the attitude of the UAV in Euler form is denoted by $R(\theta(t), \phi(t)) \in SO(3)$, where $SO(3)$ is the $3D$ rotation group. Additionally, the roll and pitch angles are respectively defined as $\phi \in [-\pi, \pi]$ and $\theta \in [-\pi, \pi]$. $\phi_{ref} \in R$ represents the reference input value in roll, $\theta_{ref} \in R$ represents the reference input value in pitch and $T \geq 0$ represents the reference input value for the total thrust. The only parameters that affect the acceleration are the magnitude and the angle of the thrust vector produced by the motors, the linear damping terms $A_x, A_y, A_z \in R$ and the gravity of earth g. This can be derived from (1). A first order system is used to model the relationship between the attitude (roll/pitch), and the referenced terms ϕ_{ref} and $\theta_{ref} \in R$, with gains K_ϕ and $K_\theta \in R$ and time constants τ_ϕ and $\tau_\theta \in R$. Additionally, a lower-level attitude controller takes as input the thrust, roll and pitch commands and generates the motor commands for the UAV. Note here that the position and the linear velocity in the described setup are obtained through the VICON motion capture system.

The MPC controller that was used for the operation of the UAV during this study, experiences two latency effects. Initially all the data $X_k(t)$ acquired by the sensing system is delayed by a time interval of t_{up}, while every thrust, roll and pitch command ($u_k(t)$, also shown in Fig. 5) sent by the MPC to the lower-level attitude controller is delayed by a time interval of t_{down}. Where t_{up} is the time required for the sensing data to reach the edge server (where the MPC controller is hosted), and t_{down} is the required time for the transmission of the commands from the edge server to the 5G enabled UAV. An overview of the expected behavior of the UAV is presented in Section II-E, and the expected outcome is observed and discussed in Section III. Although there are many articles addressing delayed closed loop control systems in the literature [24], during this study in order to evaluate the system's reliability we chose to operate within the limits of the network and thus, we purposely selected a closed loop control method that does not take into account latency effects.

In this work, the focus is on the evaluation of the behavior of the MPC over a multiple machine architecture by examining the behavior of the system in correlation with the network latency. To succeed that, multiple experiments were designed in order to identify when the system was most affected by the latency effect, as well as when the system was more reliable to network latency.

D. ROBOT - EDGE SERVER COMMUNICATION

Compared to what is considered state-of-the-art today, 5G provides various improvements over 4G LTE and also over other local area wireless networks. Although maybe not yet implemented in public networks, some of the most notable features would be availability of higher data rates, and URLLC (ultra reliable low-latency communication) to provide improved reliability and latency of the CNPC link.

This indicates that 5G is a promising technology for the evolution of UAV communications [2], [10], [25], [26]. Another striking feature is the Quality of Service (QoS) capabilities that a specific user equipment (UE) can request from the network. In addition, multi-user transmission with centralized scheduling enables the cellular network to make more efficient use of the spectrum [2]. For further information about communication comparisons, we encourage the reader to have a look at [27].

One vital aspect of the proposed framework is the integration of the ROS software with the 5G network. The problem here relies on the fact that ROS is not designed to function over public IP based and wide area networks. One of the most pronounced causes of that would be the Network Address Translation (NAT) method of translating local private IP addresses to public IPs. Since ROS requires a direct IP visibility in-between nodes a workaround is essential. Here, Virtual Private Network (VPN) was chosen in order to integrate the 5G innovation network with ROS and also utilize the encryption layer that will likely be necessary when operating autonomous missions over non-private networks. The choice of using a VPN on one hand naturally adds computational and transmission overheads but on the other hand reduces the risk of spoofing and can be considered one of the most robust solutions for running robotics applications over non-private networks. Note that our suggestion is to consider only peer-to-peer VPN solutions in order to avoid directing the packet traffic over a third node VPN server, which inevitably increases the latency of the system.

The ROS topology used in this study is briefly described. In Fig. 6, a two machine-node system is visualized. The first node, called *UAV abstraction* runs on the *UAV abstraction* component and is responsible for the UAV dynamics and the operation of the VICON system. The second node runs on the edge server and hosts both the ROS master, and the MPC controller. The VPN interface is created and evaluated over different connectivity scenarios, i.e. two scenarios for 5G (*Local 5G & Edge Cloud* and *5G with remote core & Cloud*) and one scenario for *public 4G LTE & Cloud*, thereby allowing us to directly compare the performance.

E. IMPACT OF LATENCIES IN THE CLOSE LOOP SYSTEM

The fact that the network latency affects time-critical applications has been well studied and known in the literature [28], [29]. Of interest in this study is the effect of the latency on the autonomous mission of the UAV. The circular trajectory chosen for this experiment serves exactly that purpose. Considering the design of this system, the drone executes each control command continuously until it receives a new command. From this fact, it can be expected that the trajectory of the drone might vary from its desired circular trajectory, drifting out from the circle and then rapidly trying to correct this with the future commands. An illustrative figure of this behavior is shown in Fig. 7, where an example
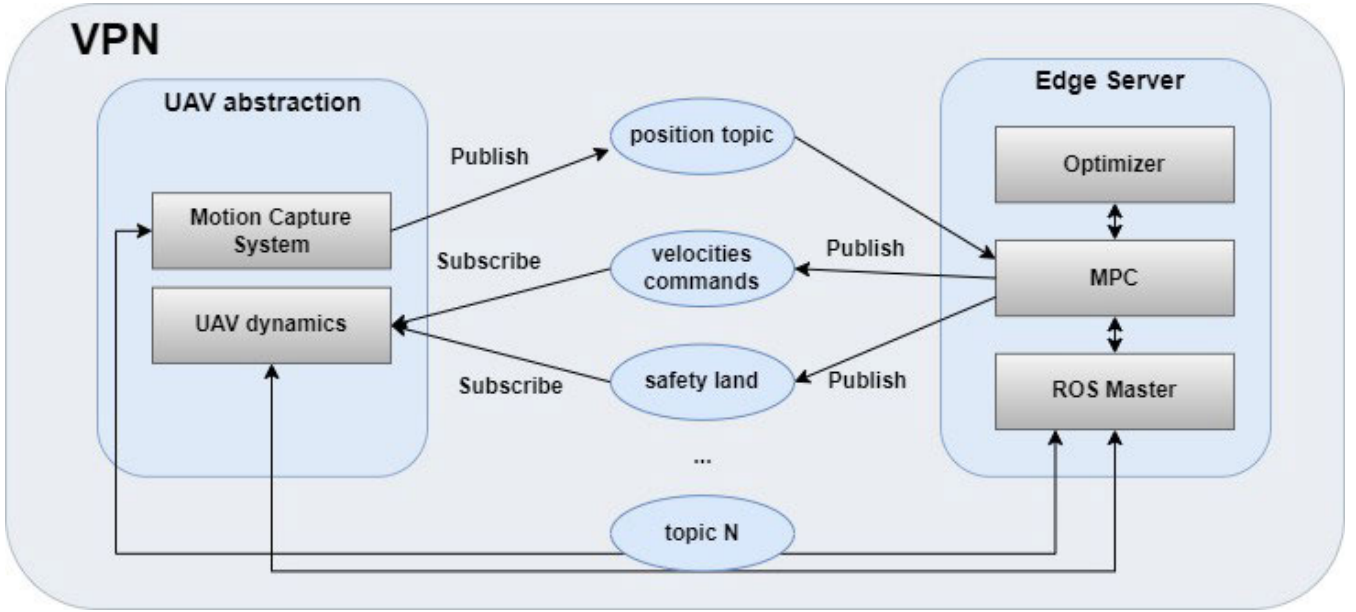
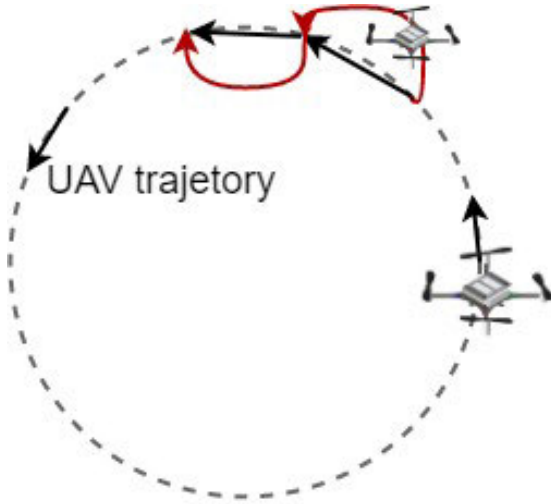**FIGURE 6.** ROS multiple machine node topology over wide area networks.

FIGURE 7. Latency behavior expected at the UAV trajectory.

of the trajectory variation is shown, as the delay in the next command occurs and in Fig. 8, the behavior of the control commands delivered at varying time intervals is shown.

Therefore, to ensure reliable and robust execution of the autonomous mission, the time intervals between the commands should be as constant as possible. A threshold L_{th} is used to denote the maximum tolerance between two subsequent control commands where,

$$\forall_N, \ T_N - T_{N-1} = \Delta T_N < L_{th} \qquad (2)$$

and T_N, T_{N-1} are the times of arrival of the two subsequent control commands. When the time interval ΔT_N between the two commands exceeds the L_{th} (latency threshold) value then

the system experiences latency. The system is declared to be robust in terms of latency when $\forall_N, \ \Delta T_N - \Delta T_{N-1} < \epsilon$ and $\epsilon \to 0$. The closed loop control round trip time is $cl_{RTT} = t_{up} + t_p + t_{down}$, where t_{up} is the time require for the sensor data to be available at the edge server, t_p is the execution time of the MPC, and t_{down} is the time required for the velocity commands to be available at the UAV abstraction node. Ideally, $\Delta T_N \to t_p$ should be preserved, as shown in Fig. 8. During section III these terms will be used to evaluate the system.

It was observed that delays in the data transmission occurring at the radio communication phase, that is, either in the *Radio1* interface or in the *5G NR* interface are negligible compared to delays owing to routing of the signal, etc. It was observed that the significant factor that affects the latency (or delay) measurements is the stochastic and to a certain amount unpredictable behavior of wide area networks. Such delays can be modeled with a combination of propagation delays, serial delays, and routing delays, but also with additional methods, such as queuing theory or Poisson distributions which can be employed to better describe delays of the system [29].

F. COLLISION AVOIDANCE USE-CASE

This subsection examines a more demanding scenario in order to evaluate the reliability of a UAV's time-critical autonomous mission when being served over a 5G interface. Here a collision avoidance scenario is considered and implemented over the proposed architecture. Consider the non-linear model predictive controller (NMPC) proposed at [30] and the UAV dynamics that are modeled by (1). The UAV performs a circular trajectory at a certain height,

FIGURE 8. Sensor data and command velocities time sequence affected by the corresponding latency.

similar to the one that section II-C described. The primary distinction is that the proposed architecture is application agnostic and the MPC controller described in section II-C is swapped for an NMPC controller that considers collisions. The NMPC controller takes the collision avoidance constraint into account and performs reactive navigation (i.e., diverging from the desired circular trajectory) to avoid the obstacle. More specifically, the constraints of this model are derived from [31] and [32]. The essence of this design is that an obstacle is represented by a sphere centered on its position. A projectile motion model determines the velocity of the obstacle. The collision avoidance condition is satisfied if the UAV lies outside the sphere. The problem, as with the use-case presented in section II-C, is solved by optimization. For more information, the reader is referred to the complete study. For this use-case, the emphasis is to capture the reliability in a real-time-critical operation, such as the collision avoidance scenario, and demonstrate that the 5G enabled UAV can reliably offload computational intensive optimization methods, such as the NMPC, to the edge cloud. That said, the purpose of this study is not to evaluate the corresponding controller, i.e., the MPC described in section II-C or the NMPC described here, but to assess whether the proposed architecture is robust enough to support any controller.

To further elaborate on the considered use-case, the following is presented. Foremost, the UAV continues to execute its circular trajectory, where the circular trajectory is a baseline trajectory that simulates many real-life scenarios. Simultaneously at each control cycle, the offloaded NMPC solves the optimization problem considering the potential

FIGURE 9. Complete data flow regarding the collision avoidance use-case.

collision with a dynamic obstacle. The obstacle in this scenario is a ball whose actual position in the 3D space is captured again by the Vicon motion capture system. The NMPC hosted on the edge server operates at the frequency of $40\,Hz$. Thus, to perform a successful maneuver and achieve the collision avoidance action, the action must be computed in the remote edge server, and the control command that describes the action must be sent back to the UAV. Fig. 9 illustrates a complete data cycle, and the separate steps are explained below:

1) The state of the robot and the position of the ball (i.e., the obstacle) are captured.
2) The data are sent from the UAV abstraction and over the 5G NR to the BS. The destination is the edge server.
3) The BS forwards the data to the Local 5G core.
4) The data arrives at the local 5G core breakout, where that instance of the 5G core takes care of routing the packets. The destination remains to be the edge server. Note that in most cellular networks data has to route through the core of the network.
5) Data are sent from the local 5G core breakout to the edge server.
6) Data reaches the edge server where the NMPC is operating. An action is decided, either to continue the circular trajectory or to perform a collision avoidance maneuver. The control command is sent back to the UAV.
7) The control command data are sent from the edge server to the UAV. First they have to route over the local 5G core breakout.
8) The control command data arrive at the local 5G core breakout, where that instance of the 5G core takes care of routing the packets. The destination remains to be the UAV abstraction.
9) The control commands are routed from the local 5G core breakout to the serving BS.
10) The serving BS sent the control command data to the UAV abstraction.
11) The control commands arrive at the UAV abstraction and are fed to the onboard attitude controller. The action occurs.

The results for the corresponding experiment are presented in section III-D. Finally, it's important to highlight that optimization techniques are computationally intensive [33], [34]. Hence, additional constraints make the real-time solution of the system increasingly demanding. Nevertheless, this is a representative example of what the proposed architecture can offer. Many UAVs with limited onboard resources would struggle to solve optimization problems in real-time. Another crucial addition would be that connected robot agents with centralized processing can easily exchange environment data, such as potential obstacles, etc.

III. EXPERIMENTAL RESULTS

The experiments that took place are divided in three categories. The performance of the autonomous mission of the UAV's is tested while offloading the main control operation of the agent to an edge server as described before. The access to the edge server is provided by the Research Institutes of Sweden [35]. Throughout the realization of all the experiments the same topological architecture shown in Fig. 3, is used. The experiments are divided as follows: 1) using a local 5G interface and routing to the edge server using the local breakout, we name this *Local 5G & Edge Cloud*, 2)

using the 5G interface but routing through the remote 5G core (located $\sim 900\ km$ away) and offloading the UAV control to a cloud resource, and 3) operating through a public 4G LTE provider and offloading the UAV control to a cloud resource. Note here that the most widely used definition of an edge server is exploited. The computational unit that was used is located in the edge cloud server of the 5G innovation network. The server used can be considered as an edge only in the first scenario, whereas for the remaining architecture scenarios, the corresponding solution of the cloud offloading topologies is considered. The different scenarios are illustrated in Fig. 2.

A. SYSTEM LATENCY EVALUATION
The reliability and consistency of the system were tested in the first batch of the experiments. For this purpose complementary cumulative distribution functions (CCDFs) of the measured uplink and downlink latencies are presented. Here, each sample of the uplink latency is defined as t_{up} and each sample of the downlink latency is defined as t_{down}, as show in Fig. 8. This essentially excludes the processing time from the equation. The results for the uplink and downlink distributions are shown in Fig. 11, and in Fig. 12, respectively. Note that each CCDF curve in both the uplink and the downlink figures demonstrates a different scenario. The curves were plotted on a $log - log$ scale to simplify the comparisons between different scenarios. In Fig. 11, it can be seen that the uplink case of the *Local 5G & Edge Cloud* option demonstrates constant behavior across different experiments. The curve appeared almost vertical without any visible significant variations or outliers. Thus, the expected behavior of the system can be considered stable, whereas the maximum latency of the application remains at an acceptable level. When it comes to the *5G with remote core & Cloud* and the *LTE & Cloud* scenarios, both of them are significantly shifted to the right and over the $100\ msec$ mark. Additionally both of them exhibit a significant elbow effect in their corresponding CCDF curves. For the *LTE & Cloud* case this point is correlated to the load of the network, as it corresponds to the different experiments conducted at rush-hours while the overall drift of the curve to the right is more radical comparing to the *5G with remote core & Cloud* scenario. Both of the latter scenarios present strong variations in the uplink and maintain a high mean latency value. Thus, these options are considered to be less reliable for the operation of a CNPC link utilized for the execution of time-critical autonomous missions.

Considering the downlink characteristics of the system, the reader is referred to Fig. 12. In the *Local 5G & Edge Cloud* scenario, the system's behavior is clearly very stable and the corresponding CCDF curve appears almost vertical, while the overall latency here is lower than the latency of the uplink. In the *LTE & Cloud* and *5G with remote core & Cloud* scenarios, the behavior is relatively stable but again present a relatively high latency when considering the operational requirements of a CNPC link. It should be noted that the latency characteristics of the experiments are mostly related
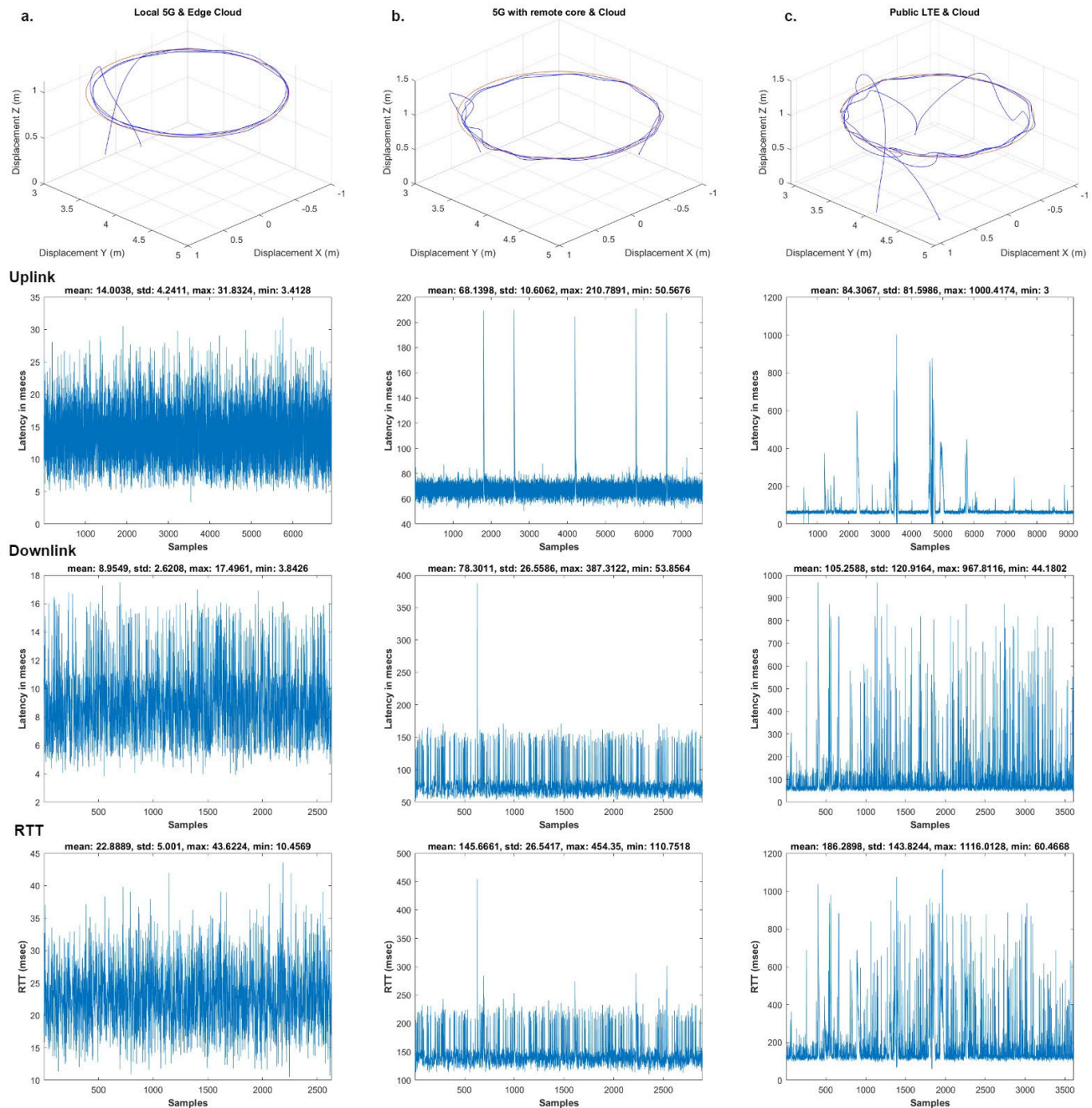
FIGURE 10. The first row shows the UAV trajectories. The second row shows the latency measurements for the Uplink interface. The third row shows the latency measurements for the Downlink interface and the forth row shows the round trip time. Each column corresponds to a different scenario for a different experiment. The first column represents the *Local 5G & Edge Cloud*, the second column the *5G with remote core & Cloud*, and the third column the *Public LTE & Cloud*. Note here that the publishing frequency of the UAV's state is at 100 *Hz*, while the MPC controller is operated at 40 *Hz*.

to the core network routing and are not strongly correlated to the radio interface of the different scenarios.

Finally, the 95th percentile of the uplink and donwlink interfaces is presented for comprehensive and comparison purposes of each architecture. The corresponding values are listed in Table 2.

To conclude this section, the round trip mean latency and the mean jitter of the system for our three different scenarios

TABLE 2. 95th percentile values of latencies in msecs. Distinguishing between different scenarios and Uplink vs. Downlink.

Scenarios	Uplink (msec)	Downlink (msec)
Local 5G & Edge Cloud	28.29	13.38
5G with remote core & Cloud	78.48	152.88
Public LTE & Cloud	150.74	135.88

is analyzed. The mean round trip latency and the jitter of the network is calculated across the entire set of the experiments
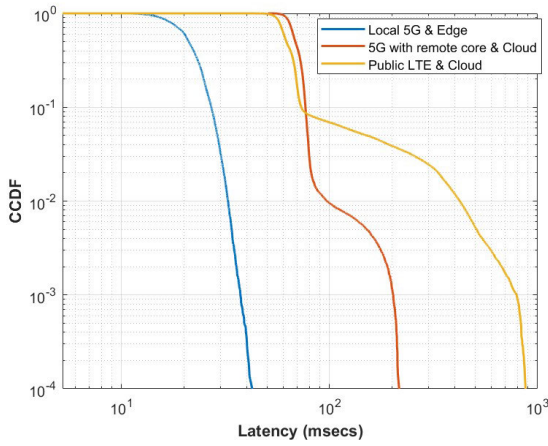
FIGURE 11. Uplink CCDF plot for each scenario. The horizontal and the vertical axes are shown in *log − log* scale. This figure is produced after an aggregation of multiple experiments. The knee behavior for the *Public LTE & Cloud* scenario corresponds to different loads of the network across multiple experiments.
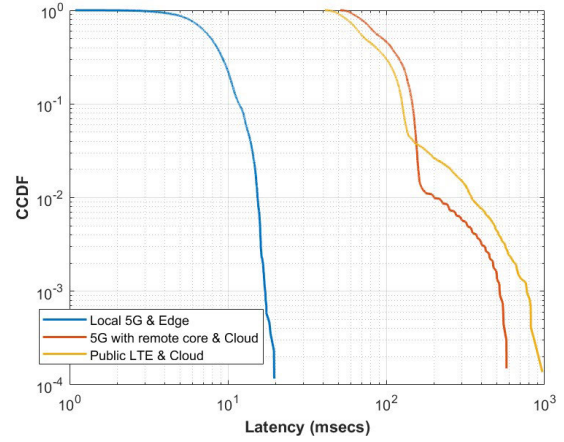
FIGURE 12. Downlink CCDF plots for each scenario. The horizontal and the vertical axes are shown in *log − log* scale. This figure is produced after an aggregation of multiple experiments. The knee behavior for the *Public LTE & Cloud* scenario corresponds to different loads of the network across multiple experiments.

TABLE 3. Mean Round Trip Time and Mean Jitter.

Scenarios	Round Trip (msec)	Jitter (msec)
Local 5G & Edge Cloud	29.33	5.34
5G with remote core & Cloud	171.87	34.62
Public LTE & Cloud	165.74	38.83

for every scenario. The jitter of the network is calculated by subtracting subsequent round trip measurements and constitutes a measurement that is usually used to express the reliability of a network. The corresponding values are listed in Table 3. Here it can be observed that for the proposed architecture (*Local 5G & Edge Cloud*) the mean RTT measurements are significantly better than the alternative examined solutions. A notable factor is the jitter measurements which constitutes a very important KPI in the robotics control field. Here the measurements indicated that the deviation in subsequent latency RTT measurements is relatively low without using any of the latency enhancement features, (such as the URLLC) proposed in the *5G* networks. Finally, the *5G with remote core & Cloud* architectures and the *Public LTE & Cloud* architecture suffer from the extended number of hops present in these networks as well as the load of the network for the *Public LTE & Cloud* implementation.

B. TRAJECTORY EVALUATION

In this subsection correlations between the latency of the system and the expected trajectory of the drone (as described in subsection II-E) is analyzed. The purpose is to provide an overall evaluation of the reliability of the system in terms of latency. The results are shown in Fig. 10. The first row corresponds to the UAV's trajectory in the $3d − space$, the second row corresponds to the uplink latency of each experiment, the third row corresponds to the downlink latency, and the 4th row to the RTT measurements. The *b. scenario* illustrates a baseline paradigm of the described

phenomenon. The UAV followed the scheduled circular trajectory, while experiencing constant latency with a mean value of $\Delta T_N = \sim 146\,msec$. This is derived by the fact that each command, generated by the MPC controller, is produced at $40\,Hz$ or every $25\,msec$, thus, as described in Section III-A, for this experiment $\Delta T_N > L_{th}$. The effect of the constant latency is not visible throughout the entire observed trajectory because the UAV flew at a low speed. More specifically, the UAV flies with a mean velocity of $\sim 0.25\,m/sec$, thus at a time interval of $\sim 146\,msec$ travels a distance of $3.65\,cm$. Now consider one of the big 4 spikes in the uplink latency graph. Here, we experienced a round-trip time of $\sim 320\,msec$ and the UAV traveled a distance of $8.0\,cm$. As expected, the four big spikes presented in the uplink latency were also visible at the trajectory of the UAV. For the *scenario a.* it can be seen that the mean round-trip latency is not significant enough to affect the UAVs operation, thus resulting in a steady and expected behavior throughout the autonomous mission. Further, the case of the public LTE network during rush hours *scenario c.* was examined. In this case the mean latency was significantly larger, hence resulting in poor performance of the UAV. This result is considered unreliable and thus the system is classified as unstable. When performing the same experiment for the *scenario c.* during non-rush-hours the result was an acceptable behavior where the effect of latency was not as visible in the trajectory. This observation indicates that the network is significantly affected by the current load. This uncertainty must be taken into account when operating CNPC links and time-critical applications, for example by utilizing QoS features of cellular networks.

Here it is important to note that compared to the existing solutions in the literature, it is essential to meticulously define and describe all the parameters beforehand according to the examined use case. For example, in [16], the simulated examined solution included an autonomous UAV executing
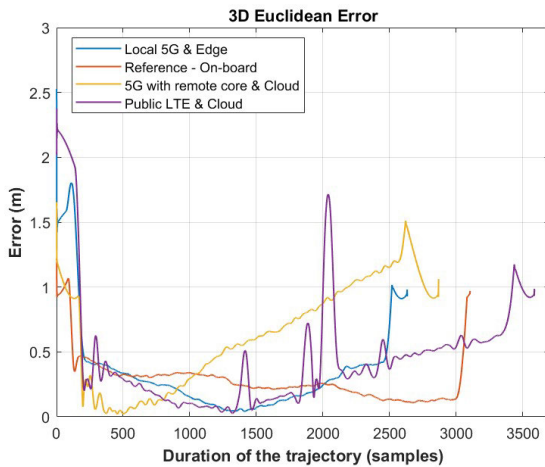
FIGURE 13. 3D euclidean error between the desired (i.e., reference) trajectory and the accomplished trajectory.

circles at the desired altitude. However, the acquired trajectory is planned to be a circle with a radius of $200\,m$, and the operational frequency of the controller is set to operate at $100\,Hz$; thus, with the presented latency of $\sim 15\,msec$, the average crossing distance is approximately 1.5 m. Therefore, the magnitude of the error is significantly higher compared to the scenario presented in the current article and would deem the solution unusable (additional information is presented in section III-D). Finally, in the considered scenario, the authors captured the dropped package rate to be approximately zero and thus did not present any correlation to the observed trajectory of the UAV. Therefore, it is critical to know that the used network interface is designed to fit well within the limits of the uplink network interface. More specifically, the uplink capacity is calculated to be $94\,Mbps$, while the measured uplink data (the captured state of the UAV) is approximately $592\,Kbps$, with an extra VPN overhead of 7%. In that case, if the UAV were about to transmit more data than the uplink interface can support, then a significant delay in the RTT time of the control packets would be observed. Further, as a result, an increased dropped packet rate would be observed at the buffers on the UE side.

C. ERROR EVALUATION

In order to further evaluate the proposed architecture and compare it with the alternatives, one additional KPI is introduced, and a reference scenario that is considered the ideal outcome for the supposed MPC controller and platform. In the reference scenario, the same MPC controller is utilized, whereas all the conducted calculations for the operation of the UAV are performed onboard the platform. Referring to Fig. 3, the onboard configuration considers only the UAV abstraction block. Delays in the sensor data packets for the uplink interface and the control command packets for the downlink interface are considered negligible since there is no external communication. In order to compare the considered architectures, the 3D euclidean distance error in time is utilized between the reference, i.e., desired trajectory

compared to the accomplished trajectory. Fig. 13 illustrates the results for the four considered scenarios. It is visible that the behavior of the system for the *Local 5G & Edge* scenario and the reference scenario is comparable since the error in time does not differ significantly. However, when comparing with the *Public LTE & Cloud* scenario and the *5G with remote core & Cloud*, it is evident that the error increments throughout the manifestation of the mission and also presents significant spikes for the *Public LTE & Cloud scenario*. Those spikes are the outcome of the large spikes visible on the delay analysis of the corresponding experiments, which are illustrated in Fig. 10. Furthermore, it is evident that the error is not bounded as it is in the *Local 5G & Edge* and the reference scenarios, which yields an unstable system. Finally, note that the baseline error that occurs throughout the manifestation of the trajectory while utilizing the reference architecture can be assigned to the fact that the UAV model does not capture the UAV dynamics completely. The small form factor of the used platform, failed experiments and crushes, and slight imbalances in the UAV's structure yield the observed baseline error.
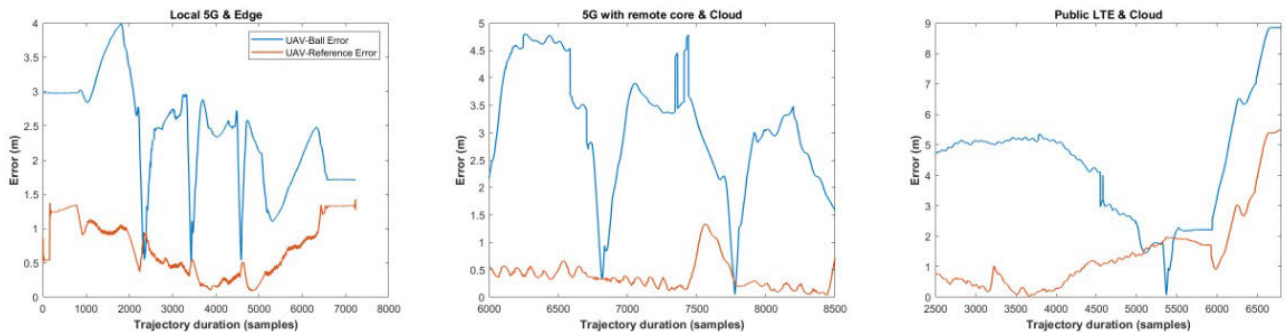
D. COLLISION AVOIDANCE EVALUATION

During these experiments, the UAV offloaded a collision avoidance (CA) task to the edge cloud while utilizing a 5G network. A dynamic ball was thrown at the UAV to collide. The ball was thrown from different initial positions and with different velocities. The robot's state and the ball's state are processed in the edge server. A potential collision is identified; if so, reactive navigation occurs, and a new path is sent back to the UAV. Fig. 14 depicts the obstacle's position, the UAV's actual position, and the UAV's desired position in the 3D space. Fig. 15 depicts the 3D euclidean error between the ball and the UAV as well as the UAV with its reference trajectory. For a collision to occur, the obstacle and the exact position of the UAV must overlay on each axis. The Euclidean distance between the ball and the UAV captures potential collisions and collisions, with a measurement error of approximately $5\,cm$. Furthermore, the Euclidean distance between the UAV and its reference trajectory indicates the reactive navigation aspect of the NMPC. For example, when a collision is about to happen, and the NMPC correctly predicts the occurrence of a possible collision, the reactive path of the UAV must diverge from its reference trajectory in order for a CA avoidance maneuver to have success.

Regarding the *Local 5G & Edge* architecture, the CA task was consistently successful. Fig. 15 illustrates three attempts to collide with the UAV and three successful deviations from the reference trajectory in order to avoid the collision. Fig. 14 shows one of the corresponding CA maneuvers. Note here that the system operates under bounded latency and jitter that this study has discussed the statistical values in the previous subsection III-A. Maintaining the expected values for such use-cases is essential, as unexpected delays would lead to the UAV's crash. On the contrary, the CA task was unsuccessful in several experiments for the *Public LTE & Cloud* and the *5G*
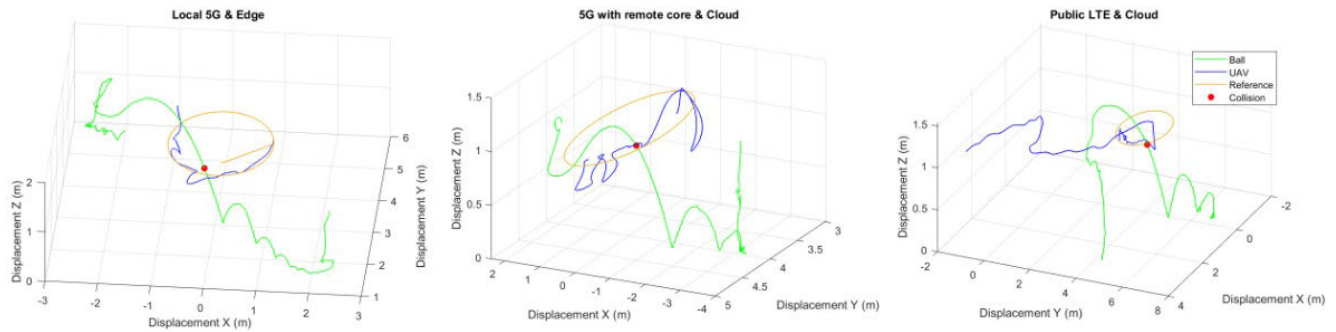
FIGURE 14. 3D trajectories for the UAV, the reference trajectory and the obstacle trajectory. In the first row the figure depicts the *Local 5G & Edge* implementation with a successful CA, in the second and third rows, the *5G with remote core & Cloud*, and the *Public LTE & Cloud* scenarios respectively. For the latter two, a collision is illustrated.

FIGURE 15. Collision avoidance Euclidean errors. The first row depicts the *Local 5G & Edge* scenario, the second row depicts the *5G with remote core & Cloud* scenario, and the third row depicts the *Public LTE & Cloud* scenario.

with remote core & Cloud architectures. The corresponding results regarding the latter two architectures are also depicted in Fig. 14 and Fig. 15, respectively. The latency aspect of an offloaded CA avoidance task constitutes an essential factor in the succession of the mission. Falanga et al. at [36] comprehensively described the effect of sensing latency in collision avoidance systems. However, two main points are derived throughout these experiments: 1) the fact that the *Public LTE & Cloud* and the 5G with remote core & Cloud architectures present significant latency compared to the controller's operational frequency, thus decreasing the permitted speed and distance ratio that the controller could initially handle, 2) the jitter aspect of the transmitted packets, makes the system less flexible when considering adjustments for delayed systems.

IV. CONCLUSION AND FUTURE WORK
A real world 5G enabled UAV with its control operation hosted on an edge server was implemented and the performance was evaluated. The proposed novel architecture was shown to work well for the operation of an autonomous mission. It was validated that the latency of the system for the *Local 5G & Edge* architecture remained well below the acceptable maximum latency level for the examined autonomous mission across different experiments. Considering the *5G with remote core & Cloud* architecture, although exhibiting relatively larger latency compared to *Local 5G*

& Edge, this architecture also maintained an acceptable behavior for the examined mission. Additional improvements can be made by hosting the cloud resources in the remote 5G core premises. When examining the public LTE network, while utilizing a cloud server as the computational unit, it was observed that the system is heavily affected by both the load of the network and the relative distance of the Evolved Packet Core of the LTE network (mostly due to the number of the in-between hops and in-between devices rather than the physical distance itself). Even though the three architectures cannot be considered as a one-to-one comparison, it was demonstrated that the feasibility of the systems depends on the network load and the in-between devices. Thus, the *Public LTE & Cloud* system for the high load case is categorized as moderately unpredictable for the considered application. On the contrary, in low load cases the *Public LTE & Cloud* architecture presented an acceptable behavior for the examined mission. Another noteworthy proposal would be the use of 4G LTE QoS features (e.g. priority scheduling) to achieve a better performance in the high load cases. Finally, it should be stated that a similar behavior was observed between the reference architecture and the proposed architecture. The reference architecture is defined as the architecture which all computations regarding the UAV autonomous mission are performed onboard.

During this article, the stochastic feature of the user's data arrival was considered, i.e., the UAV was considered

to be a user a user of the cellular network in all the examined scenarios. Nevertheless, the observed statistics on the user data, such as data packet latencies and jitter, were not utilized in the closed-loop section of this article. More specifically, an estimation "block" after the arrival of the data at the remote computing unit could be included to be a part of the overall framework. However, this article aimed to identify the raw behavior of real-life systems equivalent to that and investigate the practical limits and challenges. Additionally, to further enhance the closed-loop control performance, someone could utilize the presented results for baseline comparison or in estimation blocks during closed-loop control.

The true benefit of cellular enabled UAVs is the additional functionality and features that cellular communications can offer. Such features include wide area coverage, the use of existing infrastructure, and a CNPC link for reliable operation of the UAV and others. More specifically for the 5G case: network slicing, 5G QoS features, 5G enabled cellular positioning as well as increased bandwidth would be some notable ones. Another item to bear in mind would be the fact that ROS was used over a VPN solution, which inevitably added some processing time latency to the system. On the other hand it is essential to have a protection layer that supports the considered application when communicating over wide area networks. Considering these aspects, it can be assumed that a private 5G network solution would yield a reliable communication by lowering the latency and making the entire system more predictable. Additionally, the use of ROS without the utilization of a VPN would be possible, considering the fact that a direct IP visibility would be viable and the exposure of the application would not be a concern because of the use of a private network. Further improvement to the system by utilizing the ROS 2 QoS features [37] can be also considered. Finally, cloud technologies can be implemented in the edge server with an effort to also enable all the benefits offered by distributed applications.

This work aims to demonstrate that a real life, time critical UAV mission can be executed utilizing a 5G cellular network and an edge server. However, the proposed architecture can be applied to many similar paradigms and urges the reader to utilize and expand on this direction. Another important topic is the correlation of the radio interface with the mobile nature of the UAV. This can significantly affect the reliability of any application similar to the one that was presented in this study. Further applications that might significantly benefit from the use of 5G networks would be the potential integration with multiple agents, real-time communication and more specifically, another suggestion would be the centralized processing of UAV swarms and other UAS.

REFERENCES

[1] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, and K. W. Wen, "MEC in 5G networks," ETSI, Sophia Antipolis, France, White Paper 28, 2018, pp. 1–28.

[2] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *Proc. IEEE*, vol. 107, no. 12, pp. 2327–2375, Dec. 2019.

[3] K. Karako, P. Song, Y. Chen, and W. Tang, "Realizing 5G- and AI-based doctor-to-doctor remote diagnosis: Opportunities, challenges, and prospects," *BioScience Trends*, vol. 14, no. 5, pp. 314–317, 2020.

[4] Y. Zhang, G. Chen, H. Du, X. Yuan, M. Kadoch, and M. Cheriet, "Real-time remote health monitoring system driven by 5G MEC-IoT," *Electronics*, vol. 9, no. 11, p. 1753, Oct. 2020.

[5] S. K. Rao and R. Prasad, "Impact of 5G technologies on industry 4.0," *Wireless Pers. Commun.*, vol. 100, no. 1, pp. 145–159, May 2018.

[6] M. Gundall, Gundall, M. Strufe, H. D. Schotten, P. Rost, C. Markwart, R. Blunk, A. Neumann, J. Grießbach, M. Aleksy, and D. Wübben, "Introduction of a 5G-enabled architecture for the realization of industry 4.0 use cases," *IEEE Access*, vol. 9, pp. 25508–25521, 2021.

[7] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned Internet of Connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5213–5222, Aug. 2021.

[8] S. Li, L. Da Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018.

[9] Ericsson. *Critical IoT Connectivity: Ideal for Time-Critical Communications*. Accessed: Apr. 25, 2022. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/critical-iot-connectivity

[10] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3417–3442, 4th Quart., 2019.

[11] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of information in energy harvesting aided massive multiple access networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1441–1456, May 2022.

[12] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization-aided energy-efficient information collection in Internet of Underwater things networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1775–1789, Feb. 2022.

[13] F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz, D. Pensky, and A. Knoll, "5G for robotics: Ultra-low latency control of distributed robotic systems," in *Proc. Int. Symp. Comput. Sci. Intell. Controls (ISCSIC)*, Oct. 2017, pp. 69–72.

[14] Y.-H. Wu, C.-Y. Li, Y.-B. Lin, K. Wang, and M.-S. Wu, "Modeling control delays for edge-enabled UAVs in cellular networks," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16222–16233, Sep. 2022.

[15] H. Koumaras, G. Makropoulos, M. Batistatos, S. Kolometsos, A. Gogos, G. Xilouris, A. Sarlas, and M.-A. Kourtis, "5G-enabled UAVs with command and control software component at the edge for supporting energy efficient opportunistic networks," *Energies*, vol. 14, no. 5, p. 1480, Mar. 2021.

[16] O. Bekkouche, T. Taleb, and M. Bagaa, "UAVs traffic control based on multi-access edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[17] T. Taleb, N. Sehad, Z. Nadir, and J. Song, "VR-based immersive service management in B5G mobile systems: A UAV command and control use case," *IEEE Internet Things J.*, early access, Nov. 15, 2022, doi: 10.1109/JIOT.2022.3222282.

[18] G. Makropoulos, H. Koumaras, S. Kolometsos, A. Gogos, T. Sarlas, T. Jarvet, G. Srinivasan, and F. Setaki, "Field trial of uav flight with communication and control through 5G cellular network," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Sep. 2021, pp. 330–335.

[19] Vicon. *Vicon Motion Capture System*. Accessed: Apr. 25, 2022. [Online]. Available: https://www.vicon.com/software/captureu/

[20] B. Lindqvist, S. S. Mansouri, and G. Nikolakopoulos, "Non-linear MPC based navigation for micro aerial vehicles in constrained environments," in *Proc. Eur. Control Conf. (ECC)*, May 2020, pp. 837–842.

[21] Bitcraze. *Crazyflie*. Accessed: Apr. 25, 2022. [Online]. Available: https://www.bitcraze.io/products/old-products/crazyflie-2-0/

[22] Bitcraze. *Crazyradio PA*. Accessed: Feb. 25, 2022. [Online]. Available: https://www.bitcraze.io/products/crazyradio-pa/

[23] *DLink, 5G Router*. Accessed: Feb. 25, 2022. [Online]. Available: https://eu.dlink.com/uk/en/products/dwr-2101-5g-wifi-6-mobile-hotspot

[24] G. P. Liu, J. X. Mu, D. Rees, and S. C. Chai, "Design and stability analysis of networked control systems with random communication time delay using the modified MPC," *Int. J. Control*, vol. 79, no. 4, pp. 288–297, 2006.

[25] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[26] D. Mishra and E. Natalizio, "A survey on cellular-connected UAVs: Design challenges, enabling 5G/B5G innovations, and experimental advancements," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107451.

[27] D. Hui, S. Sandberg, Y. Blankenship, M. Andersson, and L. Grosjean, "Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE," *IEEE Veh. Technol. Mag.*, vol. 13, no. 4, pp. 60–69, Dec. 2018.

[28] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, 2nd Quart., 2018.

[29] F. Zhang, Y. Sun, L. Cheng, X. Li, J. H. Chow, and W. Zhao, "Measurement and modeling of delays in wide-area closed-loop control systems," *IEEE Trans. Power Syst.*, vol. 30, no. 5, pp. 2426–2433, Sep. 2015.

[30] B. Lindqvist, S. S. Mansouri, A.-A. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6001–6008, Oct. 2020.

[31] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3556–3563.

[32] P. Sopasakis, E. Fresk, and P. Patrinos, "OpEn: Code generation for embedded nonconvex optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6548–6554, 2020.

[33] Y. Song and D. Scaramuzza, "Policy search for model predictive control with application to agile drone flight," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2114–2130, Aug. 2022.

[34] S. S. Mansouri, C. Kanellakis, E. Fresk, B. Lindqvist, D. Kominiak, A. Koval, P. Sopasakis, and G. Nikolakopoulos, "Subterranean MAV navigation based on nonlinear MPC with collision avoidance constraints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9650–9657, 2020.

[35] *Research Institutes of Sweden*. Accessed: May 9, 2022. [Online]. Available: https://www.ri.se/en/ice-data-center/our-offer/ice-data-center-edge-and-5g

[36] D. Falanga, S. Kim, and D. Scaramuzza, "How fast is too fast? The role of perception latency in high-speed sense and avoid," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1884–1891, Apr. 2019.

[37] ROS 2. *Unique Network Flows ROS 2*. Accessed: May 17, 2022. [Online]. Available: http://design.ros2.org/articles/unique_network_flows.html

GERASIMOS DAMIGOS received the M.Sc. degree in electrical and computer engineering from the University of Patras, Greece. He is currently pursuing the Ph.D. degree with the Luleå University of Technology. He is also working with Ericsson Research. He is also a part of the European Horizon 2020 AERO-TRAIN Project. His current research interests include robotics, unmanned aerial vehicles (UAVs), and mobile networks with the main focus on 5G networks.

TORE LINDGREN received the Ph.D. degree from the Luleå University of Technology, in 2009. He is currently a Senior Researcher with Ericsson Research. His research interests include radio communication for UAVs and radio-based positioning.

GEORGE NIKOLAKOPOULOS (Member, IEEE) was affiliated with the NASA Jet Propulsion Laboratory (JPL), Pasadena, CA, USA, for contacting collaborative research on aerial planetary exploration. His team participated in the DARPA Grand Challenge on Sub-T Exploration with COSTAR1 Team of NASA, where we won the Second Stage of the Competition, in February 2020. He currently works as a Chair Professor of robotics and artificial intelligence (RAI), while heading the Robotics Team with the Division of Systems and Interaction, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Sweden. He is a member of the Board of Directors at euRobotics, a member of the Scientific Council of ARTEMIS in the field of robotics and AI, and a member of IFAC TC on robotics. He is an Elected Expert of the Permanent Working Group (PWG) of A.SPIRE with a focus on process optimization and ultra carbon coal, and also an Elected Member on the Aeneas-XECS in embedded control systems.

● ● ●