Locomotion-Based UAV Control Toward the Internet of Senses

Nassim Sehad, *Graduate Student Member, IEEE*, Bilel Cherif, Ibrahim Khadraoui, Wassim Hamidouche, Faouzi Bader, Riku Jäntti, *Senior Member, IEEE*, and Mérouane Debbah

Abstract—The emergence of 5th-generation networks and the introduction of the ultra-low latency Internet, namely, tactile Internet, by the International Telecommunication Union (ITU), has opened up a wide range of applications. Extended Reality (XR), holoportation, and remote control of machines are among the ones that would revolutionize the future of factories, smart cities, and digital healthcare. Virtual Reality (VR) technology provides users with highly realistic visual and auditory experiences, enabling a high sense of immersion and embodiment in virtual environments. In the real world, however, we use more senses than only vision and hearing to perceive our surroundings. Particularly, tactile sensation is the only bidirectional modality that enables us to perceive and interact with the objects and surfaces around us. This brief introduces a real-life testbed Unmanned Aerial Vehicle (UAV)-based system that leverages different technologies, including VR, 360° video streaming over 4G/5G, and edge computing to enable 6 Degrees of Freedom (6DoF) view with haptic feedback for a maximized immersion. The demonstration videos of the testbed are made publicly available on the following links: Link1 and Link2.

Index Terms—Virtual reality, tactile sensation, 5G and beyond, edge computing.

I. INTRODUCTION

RECENTLY, the paradigm of the Internet of Senses (IoS) has been attracting much attention from researchers and telecom vendors [1], [2], [3]. Researchers in various fields, such as human-computer interaction, artificial intelligence, and sensor technology, are exploring new ways to integrate sensors and technologies to enhance human-computer interactions. The development of the IoS is driven by the desire to create more natural and intuitive ways for people to interact with technology and create new opportunities for businesses and industries. By 2030, it is expected to integrate IoS into 6G networks, allowing for even more immersive and context-aware interactions. Additionally, this integration

can improve network performance by providing detailed information about the environment, users, and connected devices. This information can be used to optimize network configurations and resource allocation in real-time, reducing congestion and increasing efficiency [4]. Furthermore, as Unmanned Aerial Vehicles (UAVs) become more widely adopted across various industries, it is essential to evaluate the user experience when interacting with these robots, improving the overall user experience and how UAVs are used in different fields. Understanding the user's experience when interacting with UAVs makes it possible to design and optimize these devices to meet users' needs better. For example, UAVs equipped with cameras and computer vision algorithms can create an augmented reality experience, allowing users to view and interact with digital content in the context of the real world [5]. Furthermore, UAVs equipped with microphones and speech recognition technology can create voice-controlled interfaces for controlling the UAV or other applications.

Sainidis et al. [6] present a system that improves the efficiency and safety of using UAVs for first responders using single-hand gestures and augmented reality visualization. In [7], joystick control for UAVs is compared to a custom human pose control method using arm movements and human pose estimation. The latter has promising potential, and the authors plan to improve it. However, prolonged use of hand gesture interfaces can cause fatigue and discomfort, which is why [8] presents a hands-free Virtual Reality (VR) flying interface through head movements. These studies lack consideration for intuitive user movements, synchronization of multiple media, haptic feedback, and consideration to network delay between visual and added senses to improve the presence experience [9]. These shortcomings raise the need for further development, and improvement opportunities in this field [10].

In this brief, we propose a novel architecture for controlling UAVs that incorporates multiple sensory modalities including haptic feedback, which is synchronized with the UAV dynamics and End-to-End (E2E) delay. Our solution addresses the limitations of existing UAV control systems and offers improved control and user experience. To provide a more immersive and interactive experience for the user, we present three types of haptic feedback, including wearable suit vibrations, vibrating gloves, and vibrations from a treadmill's base. Furthermore, the proposed system involves the digital transmission of information for all senses through
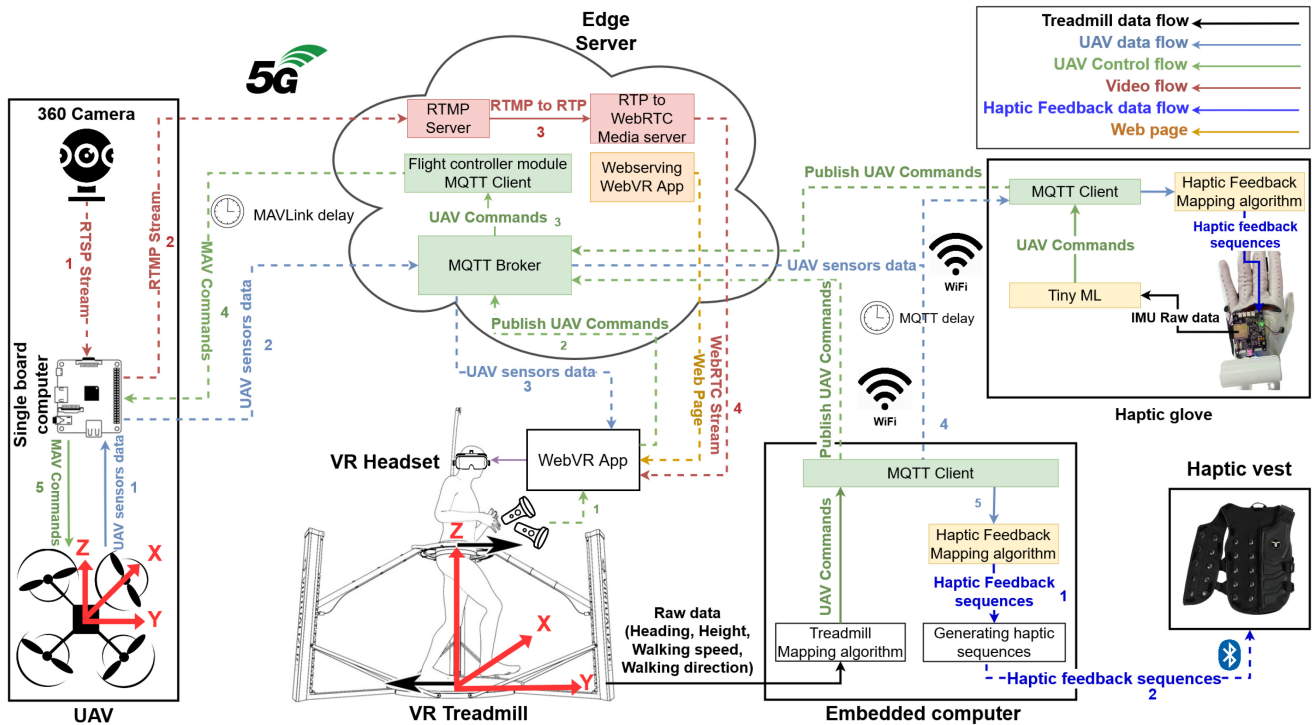
Fig. 1.   6DoF View based on UAV control using body movements on locomotion platform with haptic feedback.

streaming, eliminating the distinction between physical and virtual reality and enabling telepresence communication. Our system can be deployed in various applications, such as virtual tourism and entertainment, healthcare, disaster response, and industry.

The healthcare field requires technology for teleoperation of medical assistive robots with haptic and immersive video feedback. Although there have been studies on shared control methods and rehabilitation exercises [11], [12], few have explored the synchronization of multiple haptic feedback while considering network delays. Our proposed system can address these gaps and has the potential to be applied in the healthcare field, for example, to perform rehabilitation exercises. Additionally, the system can support disaster response efforts by combining movement and manipulation abilities. Moreover, mobile manipulation robots can support disaster response efforts, as highlighted by the 2011 Fukushima disaster. However, current robot technology has limitations in such scenarios, which require both movement and manipulation abilities [13]. Furthermore, Integrating new human-machine interaction interfaces through VR in the industry can boost productivity and safety while improving training and simulation efficiency. Our solution aligns with Industry 5.0 goals of enhancing operator safety and ergonomics through whole-body robot control and haptic feedback [14]. One other promising application for this technology is virtual tourism, where the locomotion platform provides intuitive user movements, and haptic feedback can augment the wind feeling.

The rest of this brief is organized as follows. Section II describes the proposed system for UAV control with 360° video streaming and haptic feedback. Next, the synchronization algorithm of user control and haptic feedback is presented in Section III. Section IV provides and analyzes experimental results. Finally, Section V concludes this brief.

II. System Description

Our architecture for real-life implementation features microservices-based components at the edge server that are containerized for portability and scalability. The system enables the end-users to control remote UAV using 360° video streams and sensory data from a remote location and to immerse themselves in the remote environment through natural body movements such as walking, crouching, and stretching. The remote UAV is equipped with a 360° camera that streams live footage to the user's Head-Mounted Display (HMD). The user can control the UAV through body movements once the camera stream is received. The system components are depicted in Figure 1 and explained below in further detail.

1) *Edge Server:* The edge server helps minimize latency and ease the processing burden on the UAV and HMD by offloading demanding tasks near the end user. The server includes i) a Web server that is the interface for the VR video, ii) an Message Queuing Telemetry Transport (MQTT) broker that serves as a message queue for UAV commands, haptic feedback, and network stats, ii) an Application Programming Interface (API) for the UAV's flight control that constantly listens for incoming command messages and updates its status.

2) *Embedded Computer:* A Single-Board Computer (SBC) integrated with the VR Treadmill converts height, movement speed, and heading variables from the VR treadmill into UAV instructions (speed, heading, and altitude) and sends them to the edge server through an MQTT broker. A Python script

running as an MQTT client on a separate computer communicates with the haptic suit by subscribing to topics on the edge-based MQTT broker.

3) Unmanned Aerial Vehicle: The 550mm Quad-Copter UAV is equipped with a CubeOrange flight controller running ArduPilot, an SBC, a 5G modem, and a 360° camera. The CubeOrange is connected to the SBC via USB to gather sensor data, while the SBC serves as a WiFi access point for the camera to transmit video to the SBC before being sent to the edge media server. The SBC is also connected to a 5G modem for Internet access. Payload distribution is meticulously arranged to ensure the center of gravity remains in the middle, critical for maintaining stability and achieving a balanced rotation regime. The 360 camera is fixed on the UAV's frame with a short M6 bolt to avoid affecting the moment of inertia and causing instability during flight.

4) VR User: The VR user accesses a Web page where they enter their credentials and view a map of available UAVs. After selecting a UAV, a new Web page opens with a 360° video feed from the UAV's camera. The VR user can control the camera's Field of View (FoV) using a joystick with fixed or head-following FoV angles. Users can adjust FoV angles with the joystick to control the UAV's surroundings better. Additionally, text overlays on the 360° video show the UAV's sensory data, and its position is displayed on a 2D map as an overlay. This information helps control the UAV and avoid collisions with surrounding objects, providing a more immersive and engaging VR experience.

5) Haptic Suit: The TactSuit X40, a wireless haptic vest from bHaptics, has 40 individually programmable vibrotactile motors. bHaptics offers a tool called the bHaptics designer to create custom haptic feedback patterns and export them as tact files. These tact files can be sent to the vest via a Software Development Kit (SDK) to play the haptic patterns. A Python script on a computer operates as an MQTT client to receive data from the edge broker. An algorithm processes this data to determine the UAV's status (e.g., taking off, landing, hovering) and play the corresponding haptic sequence.

6) Haptic Gloves: The built haptic gloves are used to control UAVs for takeoff and landing by converting hand movements into control commands. A neural network analyzes sensor data to send the correct command to the UAVs via a SBC. This technology offers a more intuitive and engaging experience for flying UAVs.

7) Locomotion Platform: A VR Treadmill tracks the user's movements (speed, heading, height) and converts them into UAV commands through an algorithm that processes sensor data from the treadmill platform. The algorithm also implements haptic feedback by using the UAV's sensors to translate its movements into vibrating frequencies the user feels. The haptic feedback maps each vibration frequency to a unique frequency and amplitude to give the user different sensations for different UAV maneuvers. The resulting torques at the copter motors level are reflected in different vibration amplitudes at the treadmill level. The treadmill can provide the user with a more intuitive way to explore the 360° view by enabling 6DoF and allowing them to see things up close by moving towards them.

III. User Controls and Haptic Feedback Synchronization

The overall communication architecture that links the UAV to the Extended Reality (XR) application could be seen as two sub-networks with different natures. The network link that serves the XR application (MQTT broker, media player) to the user is considered a reliable network. In contrast, the link connecting the UAV to the edge host is a non-reliable network link (5G radio). The field tests have shown that the 5G network link quality presents many unpredictable degradations when flying at high altitudes, caused by the downward tilt of the eNodeB, resulting in the UAV only receiving side lobes. Under some particular conditions, it is hard to guarantee an acceptable video streaming quality due to the limited bandwidth and high latency. Haptic feedback could be used as a second channel to inform the system user about the UAV states once the video streaming quality is unpredictable. The user could rely on haptic feedback to ensure the UAV's safety and an immersive experience. In order to achieve synchronization between the equipment that provides the haptic feedback (Haptic suite and locomotion) and the UAV's state, an algorithm was implemented to map the user's movements into UAV's movements, while prioritizing the different actions set that gets executed by the UAV, and estimates the UAV's position with respect to the user's actions based on the onboard sensors feedback. Simplifying the complex movements into a set of sequential commands allows us to accurately monitor the UAV's feedback sensors and deduce the UAV's state based on the measured variations.

Synchronizing the user's movements to the UAV's actions and providing accurate state-centric haptic feedback is a challenging task in the context of UAVs [15]. The haptic patterns have to be mapped into 3D movements of the UAV, while the haptic vibration needs to be mapped into the actual UAV's state (speed). Furthermore, the heterogeneity of the network introduces unpredictable latency and bandwidth variations. To address the issue, we identified a combination of actions that can be used to access any point in the 3D space. The chosen actions are Yaw angle control, speed control, and altitude control. The implemented strategy relies on publishing every lightweight message that describes the basic set of commands and the UAV's sensors measurement into a set of independent channels (MQTT topics). The different scripts running on the system's equipment subscribe to specific channels to deduce the up-to-date states of the UAV and the XR app user.

A. User Movements Mapping and Optimization

The locomotion platform provides mainly three sensors: navigation speed, heading, and height. The variations of the user's position on this platform could be seen as a succession of cylindrical coordinates, where the origin is fixed to the UAV's frame. Once the user is in immersive mode, the cylinder's origin is merged with the UAV's frame origin in relative positioning. The user can perform any movement using a relative angle (user's heading) to that origin, an altitude in the range of the allowed values, and a speed that gets translated into a distance. The performed movements on that

local coordinates system get translated into a global Cartesian coordinates system by Equation (1).

$$x = r \cos \theta, \ y = r \sin \theta, \ z = z, \tag{1}$$

where θ is the angle measured between the user's last position and the actual position at instant t, and r is the integral of the locomotion user's speed during the sampling time $\delta r = \int v dt$. z is the user's height at the instant t multiplied by an amplification factor $z = \alpha H$. Once the new coordinates are computed through the user's heading, speed, and height, the new position is mapped into a succession of prioritized commands that gets sent to the UAV to achieve the targeted point in space. The sent commands ensure that the UAV moves to some succession of points in space mapped to the global positioning in Cartesian coordinates. The succession of commands is determined using some fixed priorities attributed to the different types of commands to ensure that those movements will provide safe and pleasant movements in the immersive mode. Algorithm 1 gives a pseudo code of the algorithm that determines the different translation vectors based on the sensor's values that track the locomotion platform user movements. Once the translation vector is computed, the algorithm determines the necessary movements to reach the new location in space. The *map()* function is called to translate and prioritize the new spatial position into a succession of commands, mainly yaw, altitude, and speed. The *map()* function filters angle sensor measurements to reduce fluctuations caused by user movements on the platform. It also includes a limiter to adjust the UAV's altitude within allowed limits based on the user's position along the z-axis. Finally, after the filtering stage an amplification component can increase the UAV's speed to move faster than the measured walking speed.

B. Haptic Feedback Mapping and Optimization

The UAV continuously transmits data from its sensors, which are then processed to determine the UAV's status (armed, climbing, landing, etc.). Depending on the UAV's status, a predefined vibration sequence is played. The implemented architecture was designed to ensure a constant delay time for the haptic feedback, even in degraded network conditions. Furthermore, the size of the different messages was optimized to enhance the reliability of the communication. This reliable implementation relies on haptic feedback as a secondary information channel, even when video streaming of acceptable quality is impossible. The experimental results show that Mavlink messages are optimized, with average sizes ranging from 12 to 280 bytes. To communicate essential sensor measurements for the haptic feedback, messages of 88 bytes were used. Mavlink messages were designed to ensure a stable delay when transmitted over the unreliable part of the network (5G network). The received Mavlink messages are published on specified topics to an MQTT broker. After numerous tests, the MQTT messages were also optimized. Table I illustrates the impact of the Quality of Service (QoS) level and topic size on the overall message size. The decision was made to use concise topics and the QoS level 0 for the different required sensor measurements.

TABLE I
QoS LEVEL ACCORDING TO THE MQTT MESSAGE SIZE

Topic size	Small topic			Large topic		
QoS level	0	1	2	0	1	2
Packet size (byte)	76	114	171	555	774	1079

Algorithm 1 Actions Mapping Algorithm

Require: $\theta(t), z(t), r(t)$
Ensure: $|\theta_1| - |\theta_0| > 0, \theta_0 \times \theta_1 \neq 0, \theta \in [-180, 180]$
  $\theta_1 \leftarrow \theta(t), z_1 \leftarrow z(t), r_1 \leftarrow r(t)$
  $\theta_0 \leftarrow \theta(t-1), z_0 \leftarrow z(t-1), r_0 \leftarrow r(t-1)$
  // Compute the translation vector direction
  if $sign(\theta_0) = sign(\theta_1)$ **then**
    $\theta \leftarrow ||\theta_0| - |\theta_1||$
    if $\theta_1 \leq \theta_0$ **then**
      $\theta \leftarrow -\theta$
    end if
  else
    if $\theta_1 \geq 0$ **then**
      if $\theta_0 \geq -90$ & $\theta_1 \leq 90$ **then**
        $\theta \leftarrow |\theta_0| + |\theta_1|$
      else
        if $\theta_1 \geq 90$ **then**
          $\theta \leftarrow -(|180 - |\theta_0|| + |180 - |\theta_1||)$
        else
          $\theta \leftarrow -(|\theta_0| + |180 - |\theta_1||)$
        end if
      end if
    else
      if $\theta_1 \leq -90$ & $\theta_0 \geq 90$ **then**
        $\theta \leftarrow |180 - |\theta_0|| + 180 - |\theta_1|$
      else
        if $\theta_1 > 90$ **then**
          $\theta \leftarrow |180 - |\theta_0|| + |180 - |\theta_1|| - 360$
        else
          $\theta \leftarrow -(|\theta_0| + |\theta_1|)$
        end if
      end if
    end if
  end if
  // Map the translation into a succession of commands
  $cmd[] \leftarrow map(\theta, |z_1 - z_0|, |r_1 - r_0|)$

IV. RESULTS

A. Experimental Setup

The deployed measurement setup consists of multiple timers that measure the delay between different components. A timer is used to measure the delay between a published message from the Dronekit message listener of the XR application and its delivery to the subscribers through the MQTT broker service. The timer starts counting once the message is transmitted from the Dronekit message listener and stops counting once the client receives the published message. This setup measures the latency on the reliable part of the network. Another timer measures the latency on the unreliable network segment. The edge server and UAV communicate using the Mavlink protocol. The timer is deployed on the edge server side and starts counting when a special payload message, called the TIMESYNC message, is received. The TIMESYNC message contains a time value transmitted from the client to the server at a regular frequency. The timer measures the delay between
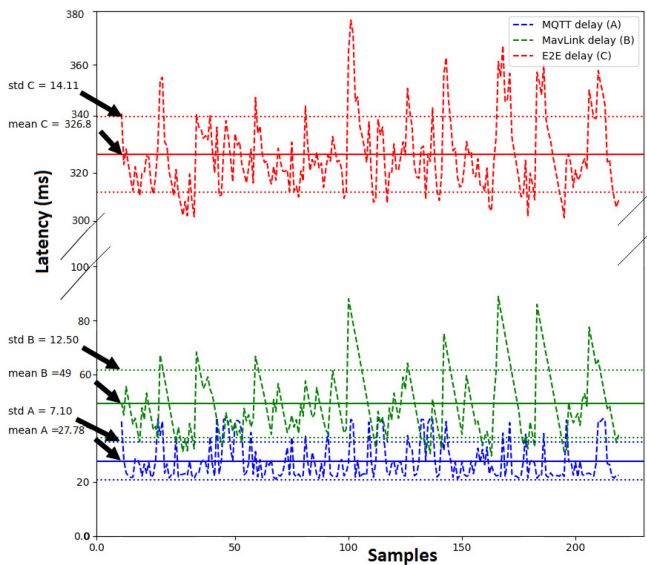
Fig. 2. Latency of the Mavlink, MQTT and E2E messages.

the successively received TIMESYNC messages. The last part of the measurement setup involves evaluating the sensor's sampling latency. Using a network packet sniffer (Wireshark), the traffic emitted by the onboard computer to the Edge cloud server is recorded during the experiment. The recorded packets are filtered using a payload identifier to only include the set of sensors used in the haptic feedback algorithm. The time difference between every two successive messages is calculated to evaluate the sensors' data availability delay. The sum of these measurements gives the E2E latency. The haptic feedback devices' response latency is excluded from the calculation as it heavily depends on the different vendors' software stacks and the used actuators.

B. Results and Analysis

Figure 2 illustrates the latency measured during the measurement of the different data paths. Plot (A) presents the reliable part of the network, MQTT, and exhibits a relatively stable and low latency with a mean of 27.78 ms and a standard deviation (std) of 7.10 ms, indicating that the data transmitted through this protocol with the chosen formatting is suitable for the application. Plot (B) shows the network latency for the critical Mavlink message, which provides the speed measurements the algorithm needs. The Mavlink network delay is low, with a mean of 49 ms and a std of 132.5 ms, showing that mobile network variations do not significantly impact the packet. The final plot (C) depicts the E2E delay, which is higher than the sum of the previous two delays, as the mean delay is 326.8 ms with a std of 14.11 ms. Analysis showed that this delay is likely related to the MEMS IMU sensor used by the flight controller, which is subject to micromechanical variations at high frequencies, affecting the accuracy of measurements. Availability delay of sensor measurements

is the bottleneck of the overall system, as the data frequency appears to be available every 200 ms, although the delay remains reasonable for our application.

V. DISCUSSION AND CONCLUSION

In this brief, we have presented an architecture of a platform that integrates multiple sensory inputs to enable the teleoperation of UAVs through a novel interaction interface, creating a more immersive user experience. The average E2E delay for generating sensory data and receipt by haptic feedback devices was 326 ms, significantly lower than the transmission delay of 4k resolution 360° video. As a result, haptic feedback provides users with better reactions to different situations, and the E2E delay remains almost constant, enabling the possibility of applying algorithms that require predictability. The sensory data can also adjust the haptic feedback intensity and mitigate high frame loss in degraded network conditions.

REFERENCES

[1] "Ericsson." Accessed: Oct. 10, 2022. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/consumerlab/reports/10-hot-consumer-trends-2030

[2] "ZTE." Accessed: Oct. 10, 2022. [Online]. Available: https://www.zte.com.cn/global/about/news/20200323e2.html

[3] R. Joda et al., "The Internet of Senses: Building on semantic communications and edge intelligence," *IEEE Netw.*, early access, Dec. 26, 2022, doi: 10.1109/MNET.107.2100627.

[4] G. Pasolini, A. Guerra, F. Guidi, N. Decarli, and D. Dardari, "Crowd-based cognitive perception of the physical world: Towards the Internet of Senses," *Sensors*, vol. 20, no. 9, p. 2437, 2020.

[5] R. Ribeiro et al., "Web AR solution for UAV pilot training and usability testing," *Sensors*, vol. 21, no. 4, p. 1456, 2021.

[6] D. Sainidis, D. Tsiakmakis, K. Konstantoudakis, G. Albanis, A. Dimou, and P. Daras, "Single-handed gesture UAV control and video feed AR visualization for first responders," in *Proc. Int. Conf. Inf. Syst. Crisis Response Manag. (ISCRAM)*, Blacksburg, VA, USA, 2021, pp. 23–26.

[7] F. Zorić, G. Vasiljević, M. Orsag, and Z. Kovačić, "Towards intuitive HMI for UAV control," in *Proc. Int. Conf. Smart Syst. Technol. (SST)*, 2022, pp. 325–332.

[8] A. M. Hashemian, M. Lotfaliei, A. Adhikari, E. Kruijff, and B. E. Riecke, "Headjoystick: Improving flying in VR using a novel leaning-based interface," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 4, pp. 1792–1809, Apr. 2022.

[9] D. Narciso, M. Melo, J. Vasconcelos-Raposo, and M. Bessa, "The impact of olfactory and wind stimuli on 360 videos using head-mounted displays," *ACM Trans. Appl. Percept.*, vol. 17, no. 1, pp. 1–13, 2020.

[10] Y. Oh, R. Parasuraman, T. McGraw, and B.-C. Min, "360 VR based robot teleoperation interface for virtual tour," in *Proc. 1st Int. Workshop Virtual Augmented Mixed Reality HRI (VAM-HRI)*, 2018, pp. 1–5.

[11] H. Lv, G. Yang, H. Zhou, X. Huang, H. Yang, and Z. Pang, "Teleoperation of collaborative robot for remote dementia care in home environments," *IEEE J. Transl. Eng. Health Med.*, vol. 8, pp. 1–10, 2020.

[12] H. Lv et al., "GuLiM: A hybrid motion mapping technique for teleoperation of medical assistive robot in combating the COVID-19 pandemic," *IEEE Trans. Med. Robot. Bionics*, vol. 4, no. 1, pp. 106–117, Feb. 2022.

[13] T. Klamt et al., "Flexible disaster response of tomorrow: Final presentation and evaluation of the CENTAURO system," *IEEE Robot. Autom. Mag.*, vol. 26, no. 4, pp. 59–72, Dec. 2019.

[14] S. Proia, R. Carli, G. Cavone, and M. Dotoli, "Control techniques for safe, ergonomic, and efficient human–robot collaboration in the digital industry: A survey," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1798–1819, Jul. 2021.

[15] M. Sabet, M. Orand, and D. W. McDonald, "Designing telepresence drones to support synchronous, mid-air remote collaboration: An exploratory study," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2021, pp. 1–17.