Exocentric control scheme for robot applications: An immersive virtual reality approach

Julio Betancourt, Baptiste Wojtkowski, Pedro Castillo Garcia, Indira Thouvenin

Exocentric control scheme for robot applications: An immersive virtual reality approach

J. Betancourt, B. Wojtkowski, P. Castillo and I. Thouvenin

***Abstract*—Unmanned Aerial Vehicles (UAVs) exhibit great agility but usually require an experienced pilot to operate them in certain applications such as inspection for disaster scenarios or buildings. The reduction of cognitive overload when driving this kind of aerial robot becomes a challenge and several solutions can be found in the literature. A new virtual control scheme for reducing this cognitive overload when controlling an aerial robot is proposed in this paper. The architecture is based on a novel interaction Drone Exocentric Advanced Metaphor (DrEAM) located in a Cave Automated Virtual Environment (CAVE) and a real robot containing an embedded controller based on quaternion formulation. The testing room, where real robots are evolving, is located away from the CAVE and they are connected via UDP in a ground station. The user controls manually a virtual drone through the DrEAM interaction metaphor, and the real robot imitates autonomously in real time the trajectory imposed by the user in the virtual environment. Experimental results illustrate the easy implementation and feasibility of the proposed scheme in two different scenarios. Results from these tests show that the mental effort when controlling a drone using the proposed virtual control scheme is lower than when controlling it in direct view. Moreover, the easy maneuverability and controllability of the real drone is also demonstrated in real time experiments.**

***Index Terms*—Virtual robotics, Virtual reality, UAVs, Teleoperation, Automatic control, Robotics.**

I. Introduction

In extreme or unsafe environments (space exploration, telesurgery, mine excavation, high pressure, high temperature, biological contamination and so on), traditional methods inevitably expose operators to danger on-site. Teleoperation systems are promising solutions when the missions are physically dangerous or impossible to do for an operator on-site or in his neighborhood. In other words, they allow the human ability to do some tasks from a remote distance and introduce several applications that recently have been developed, see [1]–[8].

Despite that the concept of teleoperation was proposed decades ago [9], it has not been widely implemented in real applications. One of the major impediments is associated with the user's limited situational awareness [10], that is defined as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" [11].

New remote interaction technics such as voice or video conferencing have reached a high level of sophistication and widespread use as an aid for teleoperation. While the feeling of being present in a remote environment is clearly possible with these systems, a complete immersion cannot be realized without the possibility of natural interaction with the remote environment.

Haptic feedback, as a complementary modality of visual and auditory modalities refer to both, kinesthetic and tactile information and include position, velocity, force, torque, vibration, etc. Using a teleoperation system with haptic feedback, the users can thus truly immerse themselves into a distant environment, i.e., modify it, and execute tasks without physically being present but with the feeling of being there. Several possible teleoperation schemes using haptic feedback to steer single or multiple robots have been proposed in the last decade [12]–[18].

Within the robotics field, there has been great enthusiasm for multi-rotor vehicles due to their characteristics: hovering capability, maneuverability and small size. Most of the applications designed for these vehicles relate to the inspection and monitoring of places inaccessible to humans. The possibility of letting aerial robots interact with the environment opens an additional wide set of potential applications such as maintenance, construction, cooperative grasping and transportation. Most of these applications require robots physical interaction with the surrounding objects and with uncertain and unknown environments which could result in an unreliable control of the vehicle. Similar problems, in the field of situational awareness, have been scrutinized in remote systems and navigation interface. Though these fields are different, their common interest is to efficiently manage actual or virtual cameras to obtain necessary situational awareness in remote worlds or virtual environments [19]–[21].

The problems of remote robot control interfaces are well-documented but are still an open field to be improved. The lack of situational awareness by the human operators and the cognitive overload of having reduced sources of information for teleoperating robots, negatively impact human performance. We propose in this paper a new exocentric metaphor in a virtual environment, with a virtual drone piloting the real drone, in order to answer the issue of situational awareness. The main contributions of this work can be stated as follows:

- a new interaction metaphor DrEAM (Drone Exocentric Advanced Metaphor) displayed in an immersive environment, and piloting the real environment.
- a new framework connecting the CAVE (Cave Automated Virtual Environment) and the drone arena with a real

This work has been sponsored by the French government research program Robotex Equipment of Excellence (ANR-10-EQPX-44).

J. Betancourt, B. Wojtkowski, P. Castillo and I. Thouvenin are with the Université de Technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex, France. (gubetanc, bwojtkow, castillo, ithouvenin)@hds.utc.fr

programmable drone prototype.

- a user experience methodology to evaluate the metaphor from the point of view of the user's cognitive overload.

Moreover from a robotics point of view, the practical applications of this system are: a) a powerful testbed for analyzing and testing robot's performance (not only aerial vehicles) when it is being controlled remotely in virtual environments, b) a system for training novice users in robots applications carried out remotely, while enhancing their confidence in this kind of task, and finally c) a system for academic purposes. From a virtual or mixed reality (VR or XR) point of view, this interface is a new system taking into account the real system uncertainty while interacting.

Related work

For teleoperation and navigation missions many UIs have been proposed for transmitting and representing the environment where the robot is evolving. For example, in [22]–[25] a bilateral teleoperation for Vertical Take off and Landing (VTOL) vehicles, based on impedance controllers with haptic feedback and considering time-varying delays was carried out. Main results consisted in providing the stability of the teleoperation loop with respect to bounded external forces (i.e. operator and environment forces).

One important feature in a teleoperation system is the visual feedback of the remote environment. It comes often from one or more monocular cameras mounted on the robot, which leads to several problems such as the limited field of view (FOV), lack of visualization and poor visibility. New different aerial drone teleoperation has been developed based on graphic interfaces and considering in a different manner the visual feedback [26]–[35]. For instance, several simulations and experimental tests for the inspection of buildings using interfaces based on head positions and gestures, and a wearable exoskeleton interface, were presented in [26], [36]. The main strength of the proposed interaction methods is the ability to perform multi-modal interactions. Soon, and with the emergence of new technology, an approach based on Non-Invasive BCI device with expressive manner on face in remote presence using quadcopter control with the Emotive EPOC headset was implemented in [37]. Electroencephalogram signals were used in experimental tests to tele-operate the vehicle. A wearable drone teleoperation is presented in [38] where the data glove allows the user to control the drone's trajectory by hand motion, and a haptic system is used to augment their awareness of the environment surrounding the robot. Similarly, in [39] a remote control of a quadrotor vehicle is proposed using electromyographic and inertial sensors. The user controlled the aerial robot in position and orientation moving his hand. Experimental tests testified the good performance of the UI. A disadvantage for all these interfaces is that the recognition algorithms must to be precise and the user needs to be close to the drone's camera in order to recognize gestures or visual markers. In other words, it is required to have the vehicle in the line of sight (LoS).

Recent advances in the field of VR and Augmented Reality (AR) have allowed the use of this technology, improving teleoperation systems [40], [41]. For example, in [42] an immersive AR environment for conduction remote maintenance via a robot has been proposed. However, neither simulation nor experimental tests were presented. Following same ideas, in [43] a prototypical system for exploring and controlling a drone indirectly from an exocentric view point was validated in simulations and experimental tests. The EXO users experienced both motion blur and slight artifacts due to the limited resolution of the head-mounted display. In [44] a ground control station for drones based in an immersive virtual environment was developed. The environment simultaneously informs the operator about the position and condition of the vehicles. Hence, only simulation results were developed. In [45] an interface based in a mixed reality environment and natural language for controlling UAVs was developed. Real experimental tests were carried out using a set of known virtual landmarks. The system is limited since the speech API cannot capture every sentence from participants accurately.

Furthermore in [32] the authors proposed a teleoperation interface for mobile robots with a freely configurable viewpoint using photorealistic textures of the physical world obtained by a omnidirectional and depth cameras. The viewpoints were acquired by a head tracker equipped on a head mounted display. Experimental tests were carried out to validate the proposed approach, however some limitations from the assumption of flat floor and the feedback of the camera depth were found. Following the same ideas, a prototype realizing the Flying Frustum concept was developed in [46], [47]. The work is based on visualization super-imposed on a 3D printout of a terrain while using either a handheld or a headset augmented reality interface. Although, some preliminaries were presented, the approach still requires formal evaluation and validation due its limitations. One limitation is the current state of augmented reality technology, and specifically the questionable usability of see-through headsets primarily due to the limited field of view. Latter on, in [35] the authors exposed a novel framework for enabling catching and visible 360 content transmission for wireless VR networks, as well reducing the traffic over the backhaul, and therefore, enabling VR users achieve their delay requirements. Simulation results have shown that the proposed approach yields significant performance gains.

More recently, in [33], [48] the authors presented an interface for providing adaptive views to enhance drone teleoperation. These adaptive views are conceived using the position and orientation of the robot while using information from a 3D point cloud. Nevertheless, real world factors such as reconstruction (pointcloud) noise and latency may likely influence results because the lack of the course preview beforehand can change participant behavior.

In [34] a third-person interface for piloting a drone was developed. The main idea in this work is to enhance the situational awareness using a spatially coupled second drone for beyond visual line-of-sight drone operation. The authors presented a proof-of-concept prototype using commercially available drones, nevertheless, some cognitive or complexity issues arose from two drone controls.

Despite that many solutions can be found in literature for solving or improving teleoperation systems using VR, new solutions are emerging and are sometimes relatives with the technological advance. In our solution, we propose that the user immersed in a virtual environment (VE) for controlling real robots in different environments. In this VE the operator perceives the real world and controls the robot (in our case an aerial vehicle) through interactions (virtual drone) within the virtual workspace. For example, the operator can move around the virtual scenario and drive the virtual aerial vehicle while the real vehicle tracks in real time the desired movements described by the virtual drone. Natural and intuitive interaction with the system is achieved through a PS Move joystick enabling the operator to define trajectories that the robot performs. In addition, one goal of this work is to evaluate also the impact of the new exocentric interaction for non-expert users when piloting an aerial drone in mixed reality, which is a new field of research combining the robotics and the virtual reality areas. Therefore, two scenarios representing different missions are proposed for practical evaluation.

The rest of the paper is organized as follow: the problem statement is addressed in section II. The virtual control scheme is presented in section III. In this section, all the components of the proposed architecture are described (testing rooms, control law and DrEAM's platform). Experimental fatigue tests results of the proposed virtual interface are carried out in section IV. Real-time experiments using a quadrotor vehicle, controlled by the DrEAM's platform are reported in section V. And finally, discussions and future work are introduced in section VI.

II. Problem Statement

Quadrotor aerial robots are a well established testbed for dynamic mobility and have been widely used for innumerable research studies in the recent years. The technological and scientific advances in designing, planning, estimating and controlling these dynamic systems, make these vehicles a versatile aerial robotic platform for many different tasks, including surveillance, bridge inspection, delivery and emergency rescue.

Piloting a drone is a difficult task because its unstable and, fast dynamics make it getting out of balance quickly with any external disturbance. Unlike a car that moves in a plane, aerial robots move in a 3-dimensional space. Besides, controlling a drone is not an intuitive task for novice pilots when they lose their spatial orientation. The clearest example is when the heading of the drone is directed towards the pilot while the joystick command is in the opposite direction.

This task of flying the quadrotor in direct view demands several cognitive overload in general for beginners, and in most cases the quadrotor ends up crashing during early flight experiments. Even if technological advances have improved them significantly by implementing inner controls for flying easily, when the vehicle leaves the operator's line-of-sight, this mental overload increases until the pilot loses the control of the robot, see Fig 1.

Reducing this cognitive overload when driving the drone is one of the main characteristics of the proposed virtual

Fig. 1: Left : stress when piloting a drone. Right : frustration after crashing the vehicle [49].

control scheme. This architecture is located in two arenas: one for the robot's testing room, where the real robots are evolving, and the second one named CAVE, where the virtual environment is placed. In this area, a World In Miniature (WIM) is modeled representing the real environment where the robot is acting. A new interaction metaphor DrEAM (Drone Exocentric Advanced Metaphor) is conceived for the virtual control scheme which represents in the virtual environment the real drone. Therefore, the goal of the architecture is proposing a solution for reducing the workload of the user when piloting remotely a real robot (in our case an aerial vehicle) from a virtual environment. This aim is conceived by an adaptive visual/sound feedback in the virtual environment, that can be used for improving the task.

III. Virtual Control Scheme

The proposed virtual control scheme can be seen as a structure composed by two blocks as depicted in Figure 2; one corresponds to the virtual environment that emulates the real scenario and the other the environment for missions of the real drone. For minimizing communications errors, the real world where and the CAVE platform communicate together via User Datagram Protocol (UDP) with their respective routers connected in the same network. Both platforms have their own VRPN (Virtual-Reality Peripheral Network) server for recovering the attitude and position of the real and virtual robot respectively. Each block is represented by their corresponding frames that in most cases are different. The following subsections explain these blocks and their components.

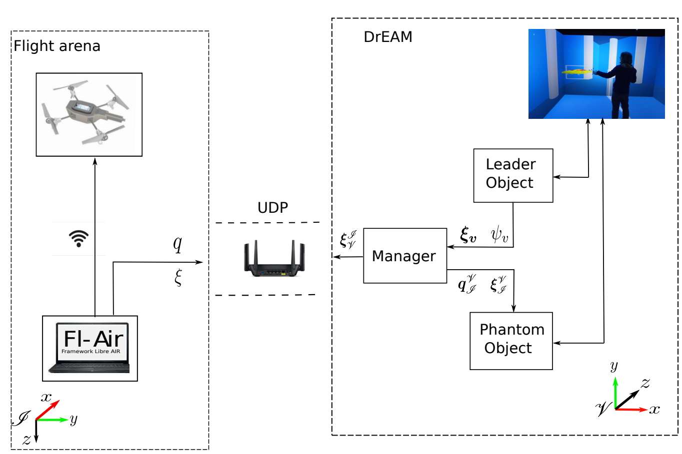


Fig. 2: Virtual control scheme

A. Real world environment

The real world environment (RWE) is where the robot is evolving. It can be seen as the scenario of the mission. This scenario will be represented as a WIM where the virtual drone will be piloted. For validating our proposal, the robot used for experimental purposes is a quadrotor vehicle and the RWE is a flight arena. Nevertheless, other real environments are possible if they can be represented as a WIM.

1) Flight arena : The flight arena is composed of an OptiTrack motion capture system (24 cameras, $1mm$ of precision) used to estimate the vehicle's position at 100Hz, and a monitoring room, separated from the test area for security reasons, where the ground station is placed. The size of the flight arena is determined by $10m \times 12m \times 6m$, see Figure 3. As previously explained, this flight arena could be also an outdoor flight arena, and instead of using the OptiTrack System a GPS RTK could provide the robot location.

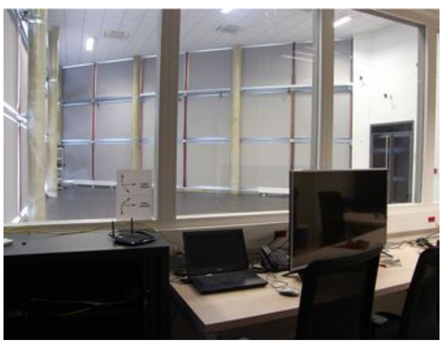


Fig. 3: Flight arena and monitoring room.

2) Quadrotor platform: The quadcopter prototype used in the flight tests is a Parrot AR Drone 2. Its firmware was modified to work under our software *Fl-AIR - Framework libre AIR* which is open source and runs a Linux-based operating system capable of implementing a wide range of control schemes, see [50]. The prototype has an internal Inertial Measurement Unit (IMU) for measuring its orientation and angular rates.

All the control algorithms are computed (each $5ms$) into the embedded system of the aerial vehicle and, each sampling period the drone communicates via Wi-Fi with the ground station (GS). This GS graphs the drone's states in the FLAIR simulator for analytical purposes. In addition, it communicates with the OptiTrack's software for collecting the position of the drone in order to send it via Wi-Fi as well among other values (desired references, gains, setup configuration, etc) as depicted in Figure 4.

The aerial drone in real-time flights is in autonomous mode using a controller that tracks the desired references coming from the CAVE. This controller is designed using the dynamic model of the robot and based on the quaternion formalism.

Quadrotor dynamic model : The translational and rotational dynamics of an aerial vehicle can be expressed as $\mathbf{x} := \begin{bmatrix} \xi & \dot{\xi} & \boldsymbol{q} & \Omega \end{bmatrix}^T$ where $\xi = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathrm{R}^3$

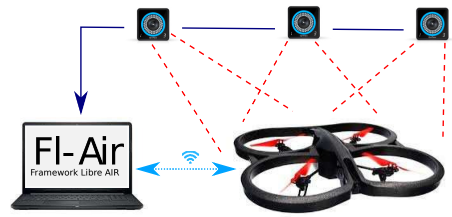


Fig. 4: Real world environment scheme.

symbolizes the position of the vehicle in the inertial frame, $\dot{\xi}$ its velocity, $\boldsymbol{q} = q_0 + \vec{q}$ with $\vec{q} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$ defines the vehicle orientation with respect to the inertial frame, represented as a unit quaternion and $\Omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ represents the rotational velocity in the body frame located on its center of mass. Let us consider the following frames: $\mathscr{I} = \{e_x, e_y, e_z\}$ which defines the fixed inertial coordinates and $\mathscr{B} = \{e_x^b, e_y^b, e_z^b\}$ representing the moving body frame as can be seen in Figure 5.

Therefore, following Newton's equations of motion, the mathematical representation of the quadrotor dynamics can be expressed as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\xi} \\ \mathbf{q} \otimes \frac{F_{th}}{m} \otimes \mathbf{q}^* + \bar{g} \\ \frac{1}{2}\mathbf{q} \otimes \Omega \\ J^{-1}(\tau - \Omega \times J\Omega) \end{bmatrix}, \qquad (1)$$

where the mass m is assumed to be constant, J defines the inertia matrix, F_{th} represents the main thrust of the vehicle in the body frame, $\bar{g} = (0, 0, -g)^T$ where g defines the gravitational force and τ denotes the external torques applied to the vehicle. The term $\mathbf{q} \otimes \frac{F_{th}}{m} \otimes \mathbf{q}^*$ represents the main thrust in the inertial frame and $\mathbf{q}^* = q_0 - \vec{q}$ defines the quaternion conjugate. This mathematical representation allows to use different sizes of prototypes and configurations of rotorcrafts.

The rotating blades generate the main thrust $F_{th} = \begin{bmatrix} 0 & 0 & \sum_{i=1}^4 f_i \end{bmatrix}^T$ where f_i represents the force in rotor i. The total torque of the vehicle is given by $\tau = \begin{bmatrix} \tau_\theta & \tau_\phi & \tau_\psi \end{bmatrix}^T$, for the pitch, roll and yaw movements.

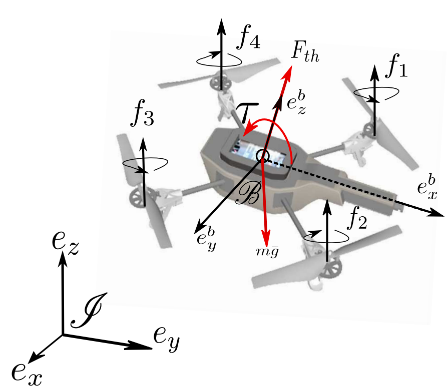


Fig. 5: Coordinate systems acting in the aerial vehicle. f_i denotes the force produced by each motor, F_{th} symbolizes the main thrust and τ represents the total torque.

Control scheme : The controller design for the real robot is in general a challenge, and several researchers have been working to improve performance and give robustness to the robot [51]–[55]. This task is carried out, generally, in a separate way before to remotely control the vehicle. In commercial robots, for example aerial drones are equipped with inner controllers for stabilizing their attitude (using Euler angles) and the translational movement is commanded manually by the pilot.

In this work, the control algorithm is proposed using the quaternion formalism. The mathematical underactuated representation of the vehicle is transformed into a full-actuated system by imposing a desired orientation with a unit quaternion, $\mathbf{q}_d$, that is related with the desired main thrust imposed by the controller, F_u. Therefore, when $\mathbf{q} \to \mathbf{q}_d$ implies that $F_{th} \to F_u$. These ideas have been developed in our previous works [56], [57], and they are extracted in the following main parts. As stated before, the goal of this paper is not to present a new controller for the quadrotor vehicle, instead of this, our challenge will be to control this vehicle remotely using a virtual environment, that falls within the scientific axis named virtual robotics.

The desired quaternion is proposed with the form

$$\boldsymbol{q}_d = \boldsymbol{q}_{f_d} \otimes \boldsymbol{q}_{\psi_d}, \tag{2}$$

with

$$\boldsymbol{q}_{f_d} = e^{\frac{\ln\left(\hat{F}_u \otimes \hat{F}^*_{th}\right)}{2}}; \quad \boldsymbol{q}_{\psi_d} = e^{\frac{\psi_d \hat{F}_{th}}{2}}; \quad \forall \psi_d \in \mathbb{R} \tag{3}$$

where ψ_d denotes the desired heading. In addition, $\hat{F}_u := \frac{\vec{F}_u}{\|\vec{F}_u\|}$.

Let us define the quaternion error as $\mathbf{q}_e = \mathbf{q}^*_d \otimes \mathbf{q}$, therefore, differentiating it with respect to time, it follows that

$$\begin{aligned} \dot{\boldsymbol{q}}_e = & = \tfrac{d}{dt}\left(\boldsymbol{q}^*_d \otimes \boldsymbol{q}\right) \\ \frac{1}{2}\boldsymbol{q}_e \otimes \Omega_e = & \ \frac{1}{2}\boldsymbol{q}^*_d \otimes \Omega^{\mathcal{I}} \otimes \boldsymbol{q} - \frac{1}{2}\boldsymbol{q}^*_d \otimes \Omega^{\mathcal{I}}_d \otimes \boldsymbol{q} \\ \Omega_e = & \ \boldsymbol{q}^* \otimes \left(\Omega^{\mathcal{I}} - \Omega^{\mathcal{I}}_d\right) \otimes \boldsymbol{q} = \Omega - \Omega_d \end{aligned} \tag{4}$$

where Ω_e represents the angular rate error in the body frame, $\vec{\Omega}^{\mathcal{I}}$ and $\vec{\Omega}^{\mathcal{I}}_d$ are the angular velocity and the desired angular velocity in the inertial frame and Ω_d denotes the desired angular velocity in the body frame.

For validating the virtual control scheme in real-time experiments, the following attitude control algorithm is proposed

$$\tau = J(-2K_{p_q}\ln(\mathbf{q}_e) - K_{d_q}\Omega_e) + \Omega \times J\Omega, \tag{5}$$

where $K_{p_q} = \mathrm{diag}\{[k_{p_\phi}, k_{p_\theta}, k_{p_\psi}]\} > 0$ and $K_{d_q} = \mathrm{diag}\{[k_{d_\phi}, k_{d_\theta}, k_{d_\psi}]\} > 0$ denote the control gains.

Note from (3) that the only desired value imposed in the controller is ψ_d. This value is related with the heading of the virtual drone and must be mapped from the virtual drone frame to the real drone frame before being used in the controller of the real robot. Observe that when introducing (5) into the rotational dynamics in (1) implies that these dynamics are stabilized. This means that $\mathbf{q}_e \to 0$ and then $F_{th} \to F_u$.

F_u can be seen as the desired behavior imposed to the aerial robot. In addition it is referred with respect to the error between the drone position and the desired translational motion (implying in some cases also their translational velocities), therefore in general words it is the proposed control input for the translational displacement.

Let us propose $F_u \in \mathbb{R}^3$ in the following form

$$\begin{aligned} F_u = & \ m\bar{g} - K_{p_t}(\xi - \mathbf{q}_{r\leftarrow v} \otimes \xi_v \otimes \mathbf{q}^*_{r\leftarrow v}) \\ & -K_{v_t}(\dot{\xi} - \mathbf{q}_{r\leftarrow v} \otimes \dot{\xi}_v \otimes \mathbf{q}^*_{r\leftarrow v}), \end{aligned} \tag{6}$$

where $K_{p_t} = \mathrm{diag}\{[k_{p_x}, k_{p_y}, k_{p_z}]\} > 0$ and $K_{v_t} = \mathrm{diag}\{[k_{v_x}, k_{v_y}, k_{v_z}]\} > 0$ are control gains. ξ_v describes the position of the virtual drone expressed in the virtual frame, the term $\mathbf{q}_{r\leftarrow v} \otimes \xi_v \otimes \mathbf{q}^*_{r\leftarrow v}$ represents the mapping of the data from the virtual environment to the inertial frame where the real drone is evolving.

Notice from (5) and (6), when $\mathbf{q}_e \to 0$, implies $\mathbf{q} \to \mathbf{q}_d$, and this means that $F_{th} \to F_u$ and therefore, $\xi \to \xi_v$. Observe that even if the controllers (5) and (6) have a simple form, they were obtained from an exact transformation from the underactuated system to a full actuated system, implying a desired quaternion for having a unique force representation. This transformation makes the system robust with respect to bounded external disturbances and nonlinear uncertainties in the model.

B. DrEAM in CAVE

1) Cave Automated Virtual Environment: A CAVE-like platform was used for validating DrEAM. This virtual room has a dimension of $7 \times 3.4\ m^2$ and is composed of four 3D projectors *Christie Mirage* 1920×1200, a workstation *HP* Z840 with two graphic cards *Nvidia* M5000, RF Active 3D Glasses, an OptiTrack motion capture system with 10 cameras, and a *PS Move* motion joystick. A Unity-plugin called *TransOne*, encapsulates data from VRPN into Unity Objects to simplify the data protocol between the motion capture system and Unity, and with the framework called Translife which creates the virtual environment.

2) DrEAM-Drone Exocentric Advanced Metaphor: The DrEAM is a new interaction metaphor conceived from the characteristics of the real robot and the scenario to create virtual robots and a WIM in the virtual environment where they can evolve. The goals of using DrEAM in robotics applications are; on one hand, training inexperienced users for developing real missions (inspection, surveillance, etc) using (semi) autonomous robots. On the other hand, to propose a friendly UI for reducing the cognitive overload when controlling remotely a real robot from a virtual environment. With DrEAM aside, the user is a spectator of the virtual environment (that is a 3D reproduction of the world), he can interact with it for moving, readjusting and rotating it using basic commands.

DrEAM naturally offers multiple points of view and multiple scales where the user can operate, all without requiring

Fig. 6: Left: Real world (flight arena). Right : DrEAM's virtual environment. Observe, in the right picture, that user is in DrEAM piloting the virtual drone for passing trough a virtual window, and note in left figure that the real drone is crossing also the 'real' window in autonomous mode without knowing the obstacle position.

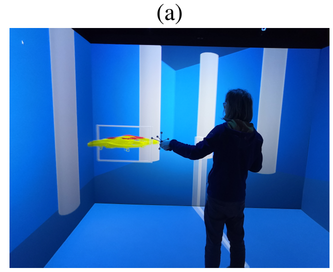
explicit modes or commands. In addition, it allows to control the real robot (via the virtual robot) in different views producing intuitively less physical and mental fatigue with respect to control it in direct view. User observation in the field of VR [58]–[61] indicates that the novice operators quickly adapt to the Worlds in Miniature, and that physical props are helpful when manipulating the WIM and other objects in the environment, see Figure 6.

The CAVE environment setup is mapped 1:5 to the real environment considering as statically scenario, using blender homemade model and unity basic shapes. This mapping is empirical and was determined during pre-tests of the experience in order to be manipulated in the cave. The virtual robot is a rigid body not affected by the law of physics that can be selected by the user using the *spherical ray-casting* function provided by Translife. This infers that its position will be the *PS-Move*'s position in the 3D-scene. The virtual frame of the 3D scene has the same origin as the inertial frame in the flight area, nevertheless, they were conceived with different rotations. In addition, target locations were placed and marked in the real environment using adhesive tape at a corresponding point on the floor and in the virtual environment with two spheres at the proper point.

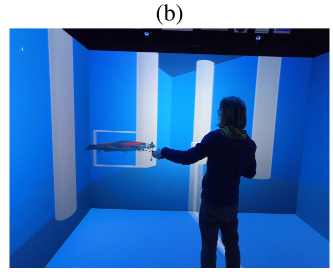
Once the real scenario is represented in WIM with a direct relationship between life-size objects, a virtual drone (VD) is placed at the same position where the real drone is. This VD is an object of the virtual world that the user can take/leave it by pressing/releasing a specific button on the *PS-Move*. While the user takes the VD, he can rotate and translate it with simple natural gestures (moving his hand), therefore the user has the impression that the drone is really in his hand.

In addition, user can readjust and reorient his virtual environment with respect to his position without changing, in any way, the scene (object positions), since the program only change user's position (in case of translation and rotation) and the field of view (in case of readjusting). As explained before, in DrEAM, the VD is manipulated by the user by only pressing the "Take" button on the index of the wand. The first step for manipulating the drone is that the PS Move must to be placed in the hitbox of the VD (which is the same size as the real robot). With this measure, a precise control of the VD is guaranteed. In the second step, the user can describe a trajectory with the wand, while the VD is already in his hand. Besides, the color of the VD changes to help the user known its state; a).- red when it is not possible to be taken, b).- yellow when it is ready to be taken and c).- black when has been already taken, as is depicted in Figure 7.

(a)



(b)



(c)

Fig. 7: Different states of the virtual drone; (a) - cannot be taken, (c) - can be taken and (b) - has been taken.

In DrEAM, two different feedbacks for helping the user are conceived. The first one is a visual feedback of the position and orientation of the real drone. For this, a 'phantom' drone is designed and placed with the information of the real drone coming from the flight arena. Therefore, inaccuracies in the drone's location and/or errors when tracking references can be visible for the user via this phantom drone. The second feedback is a speed indicator of the real drone emulated by a sound feedback in the CAVE. For this, the intensity of the sound is varied as a function of the motor's speed. A pre-recording of the drone's sound is previously done and matched in Unity with a proportional gain matching the set of speed $[0; v_{max}]$ with the set of intensities $[0; 1]$.

3) Virtual representation: When the virtual drone (VD) is manipulated via the *PS Move*, it describes a trajectory that is a function of its position and velocity (translational and rotational), i.e. $\mathbf{x}_v(t) = f(\xi_v(t), \dot{\xi}_v(t), \mathbf{q}_v(t), \Omega_v(t))$ where the subindex v refers the virtual drone. All the data are in the virtual frame $\mathscr{V}$.

The data information $\mathbf{x}_v$ is periodically sent to the RWE, as desired references for the autonomous navigation of the real drone, using the Windows asynchronous socket API in DrEAM's platform and the Linux socket API employing a string-based protocol in the ground station. The UDP protocol is used to minimize communication errors and reduce Jitter. However if there were these errors in real-time flights, our aerial prototypes include a protocol in the control system for safe landing.

The VD is referred in the frame $\mathscr{V}$ and the real drone in the inertial frame (or body frame), in the real world. Both frames are different and a mapping between them (Rotation matrix) is necessary in order to use the data of each one in their respective frame. For example, for a given point $(\xi_v, \mathbf{q}_v)$ of the virtual drone in the virtual frame, with $\xi_v = \begin{bmatrix} x_v & y_v & z_v \end{bmatrix}^T$ and $\mathbf{q}_v = \begin{bmatrix} q_{0_v} & q_{1_v} & q_{2_v} & q_{3_v} \end{bmatrix}^T$, its representation in the inertial frame (real world) is $\begin{bmatrix} z & x & y \end{bmatrix}^T$ for the translation part and $\begin{bmatrix} \cos(q_{2_v}/2) & 0 & 0 & \cos(q_{2_v}/2) \end{bmatrix}^T$ for the orientation part. Observe that the pitch and roll information is not yet used in the virtual drone.

Therefore, for a good correlation between frames the following mapping is defined

$$\mathbf{q}_{r\leftarrow v} = \frac{1}{2} + \frac{1}{2}\mathbf{i} + \frac{1}{2}\mathbf{j} + \frac{1}{2}\mathbf{k}. \tag{7}$$

Hence, the position of the virtual drone can be correctly represented in the real world as a desired value by

$$\xi_d = \mathbf{q}_{r\leftarrow v} \otimes \xi_v \otimes \mathbf{q}^*_{r\leftarrow v}. \tag{8}$$

Similarly, for the visual feedback in the virtual environment, the attitude and position of the real drone are expressed as

$$\begin{aligned} \xi_p &= \mathbf{q}_{v\leftarrow r} \otimes \xi \otimes \mathbf{q}^*_{v\leftarrow r} \\ \mathbf{q}_p &= \mathbf{q}_{v\leftarrow r} \otimes \mathbf{q} \otimes \mathbf{q}^*_{v\leftarrow r}. \end{aligned} \tag{9}$$

where ξ_p and $\mathbf{q}_p$ define the attitude and position of the phantom drone in the virtual environment. This information is updated in DrEAM every frame ($100Hz$ rate) using the last information received from the real drone and mapping it to the virtual frame using the above equations.

The virtual environment can be fitted to any space, this means that, it can represent a complex or simple structure, such as a building, a cube or even a sphere. Using this property, it is then possible to impose bounds in the area where the real drone is evolving and at the same time, keep safe the prototype. For our room test (flight arena), the virtual drone will be bounded inside of a virtual cubic scene with the following properties

$$\begin{aligned} x_{b_1}(t) \leq x_v(t) \leq x_{b_2}(t) \\ y_{b_1}(t) \leq y_v(t) \leq y_{b_2}(t) \\ z_{b_1}(t) \leq z_v(t) \leq z_{b_2}(t) \end{aligned} \tag{10}$$

where $x_{b_i}, y_{b_i}, z_{b_i}$ with $i : 1, 2$ are the bounds that can be functions of time or constants delimiting the testing room.

IV. DrEAM'S EXPERIMENTAL FATIGUE TESTS RESULTS

In order to test the advantages of DrEAM when controlling a real drone over a control in direct view, an experimental study was led with eight volunteers evaluating their performance among six criteria: Mental demand, Physical Demand, Temporal Demand (these depict the stress involved by the control task), performance (this pictures the sensation of success), Effort, Frustration (these show the sensation of UAV's obedience).

A preliminary form has given to each participant to assign their order in the tests. Groups were balanced according to six factors: gender, age, experience in virtual worlds, experience in immersive virtual world and experience in control of UAV. Even though we made efforts to motivate different genders, only men were motivated to participate. One of them had some experience in piloting a drone and no one had experience in virtual or immersive virtual world, the average age of the participants was 22. In addition, each participant performed a specific task using either a joystick or DrEAM, and then they switched for using the other control mode.

The test consists to fly manually the vehicle with a conventional joystick and with the DrEAM architecture, when performing a specific task. The participants had four minutes to learn how to use the platforms and the task must be finished in three minutes, i.e., for each passage, the participant was briefed on controls, teaching him how to translate and rotate the UAV. Then they were brought in the test area (virtual or real) where an experimented pilot explained them the task physically, imitating the flight that the UAV must do. The goal was to avoid misinterpretation of the task. After this short introduction, the UAV takes-off in safe and autonomous mode and the pilot moves it to the start point of the task, he gives the control to the participant and then he has four minutes to train. After this time, the experimented pilot activates the safe landing for the aerial robot for changing the battery and beginning the test. In this test, the pilot starts logging flight information and the participant has three minutes to perform the task.

The scenario of the task is settled as follows: the volunteers take the control of the drone when it is hovering around the start point (S) with a fixed altitude z_t. The goal is to move the vehicle from the point (S) to the point (C) and then to the point (A) keeping, all the time, the heading of the vehicle pointing to a desired target as illustrate in Figure 8.

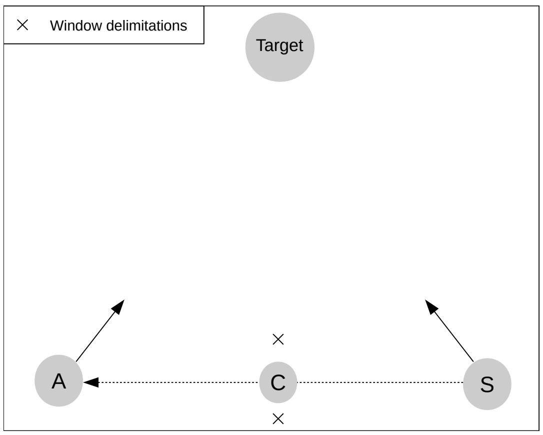


Fig. 8: Navigation task: start point (S), checkpoint (C) and the final point (A) keeping the heading of the vehicle pointing to the target.

During the test each participant must follow the following rules

- The altitude of the drone should be kept constant, i.e. $z(t) \approx z_t = 1$m.
- The real drone must not be put in danger in any case.
- The task must be accomplished with accuracy as high as possible.
- The task must be achieved as fast as possible.
- The roll and pitch angles of both experiments were previously stabilized with an inner controller.

After each flight test, a NASA-TLX form was filled by each participant. Figure 9 depicts the results of this questionnaire. Note from this figure that for every index when controlling the aerial robot with a conventional joystick, for beginner pilots, it can be complicated to have success even for a simple task as proposed in the test.

From our experience, we have observed that users generally had better performances when controlling the aerial drone with the immersive environment than with the joystick, and felt less mental and temporal demand. These results allow us to presume our initial hypothesis, which was that DrEAM would increase the maneuverability, without losing precision in the task, with respect to a direct view control where involving movements in the heading of the vehicle. It can be explained as follows, in direct view, the user needs to activate two or more controls of the joystick, some to move the drone in translation, and other one for controlling its heading. Moreover, the user must see all the time the drone movement at the same time that he uses the joystick to control it, producing in novice users a considerable cognitive load. In DrEAM, the user can orient and place the vehicle as wishes in one simple task.

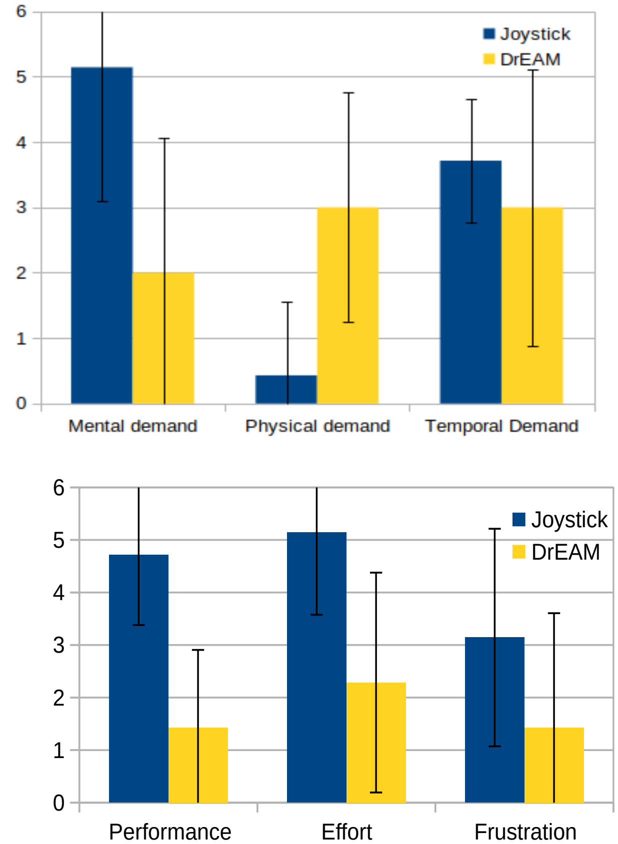


Fig. 9: Graphics results from the NASA-TLX questionnaire. From graphs 0 means low and 6 high demand or impact on the presented criteria.

In addition, almost every participant begun the task in CAVE asked the necessity to perform a so simple task, because he considered the task so easy and sometimes boring. Conversely, it was not the case for users starting with the joystick in direct view, because they considered the task hard, and they asked if with DrEAM the task would be easier.

From experimental tests, flight data information as timestamp, attitude, position and velocities are used to compute the mean lateral error (MLE) with respect to the desired path, the mean completion time (MCT) of all laps (a lap is every time user finishes the task from the initial point to the final one) and the mean yaw error (MYE) with respect to the target. In Table I the results of these indexes are presented. During the tests, 162 laps were successfully performed by the users: 81 with a joystick (direct view) and 81 using DrEAM. Latency between CAVE and the flight arena was around $0.06s$ and no jitter has been registered, which could have disturbed the flight tests. DrEAM was sending data at $100Hz$ as well the aerial drone.

TABLE I: Data results of MLE, MCT and MYE indexes

Index	Joystick	DrEAM
MLE	0.389 m	0.104 m
MCT	6.810 s	5.130 s
MYE	0.252 rad $(14, 43$ deg)	0.140 rad (8.02) deg

To corroborate the outcomes obtained with the NASA-TLX questionnaire, a one sample t-Test was carried out considering a maximum percentage of confidence $(5\%, \alpha = 0.05)$ in the six performance criteria. Results of this test are shown in Table II, where H_0 means DrEAM has no impact on the performance criteria. Notice from this table that the $p-$value on almost all the performance criteria, excepts for the 'temporal demand' and 'frustration', is less than 5% $(p < 0.05)$. Therefore, it is possible to state that DrEAM has no impact on these criteria. Nevertheless, for the case of the 'temporal demand' and 'frustration', with a $p-$value bigger than 0.05, it is not possible, at the moment, to state that there is not demand of this criteria.

TABLE II: Results from the one sample t-Test.

Hypothesis	p-value	H_0
Mental Demand	0.013	Rejected
Physical Demand	0.007	Rejected
Temporal Demand	0.269	Not Rejected
Performance	0.003	Rejected
Effort	0.022	Rejected
Frustration	0.140	Not Rejected

The final conclusion in these tests was that DrEAM increases the control ergonomics for inexperienced users when controlling the aerial robot, without loss of precision, in comparison when controlling the robot in direct view, in particular, concerning complex movements with more than two DoF. In addition, participants evaluated the virtual world more physically demanding. This is obvious, because in DrEAM user has to move for doing the task, that it is not the case when controlling in direct view.

Finally, we can conclude that this preliminary study presents a first step in the field of exocentric metaphors for aerial robot control. And according with this analysis (using the NASA-TLX forms) the exocentric interaction seems to have better user experience performances than controlling the vehicle in direct view. Moreover, further experiments could be done with other devices (gloves, etc) or scenarios for controlling the drone in the virtual environment.

However, the previous results need to be used very cautiously because they are based on a small sample of novice users. Nevertheless, they can also be seen as a first encouraging step in the study of exocentric metaphors for control robots.

V. DrEAM for robotics applications

Motivated and encouraged by the previous results, we have proposed to emulate real missions for robotics applications using the proposed virtual architecture. In these experiments two difficult tasks were done implying the user the need to change the orientation and altitude of the aerial drone (adding more degrees of freedoms to control that in the previous tests). The pilot for the experiments has not experience in controlling aerial robots.

In the first task, the scenario was the aerial inspection in reduced spaces. For this, the goal is to navigate with the real robot inside a building crossing small surfaces, e.g. as windows. In the second task, the scenario was a cellar inspection, therefore, the objective is to send the aerial vehicle inside of the cellar and navigate there to survey the zone. The cellar is composed by pillars that the aerial vehicle needs to avoid. Both tasks aside from needing the aerial navigation in the environment, need a precise control of the task, a good communication between arenas and good relation between the real and virtual worlds. For example, when the drone is crossing a window or turning in the cellar (changing its heading), without any of them, the drone could crash because it has no sensors for obstacle avoidance or does not know the position of the obstacles. Even if some of them (e.g. communication errors) can be solved using observer/predictors schemes, a good practical validation is always necessary.

Therefore, the practical goals of these experiments are : a) to corroborate and compare the performance of the real robot against references coming from the virtual world, b) to verify (graphically) if a neophyte user can do a hard tasks with good precision, and c) to verify (graphically) the matching between real and virtual worlds.

The tests are carried-out with the following procedure

- The take off and landing of the real robot are not controlled using DrEAM. They are done in a safety mode by a user in the flight arena.
- The initial positions of the real drone for the tests are $\xi(t_0) = (0, 0, 0.5)$m. At this position, DrEAM can take the control of the vehicle.
- At any moment, the user in the virtual world can always put the vehicle at hover position, to change/rotate/expand the view of the WIM.
- For simplifying the test, when the user takes the virtual drone and moves it, the real drone will be aligned to the direction of the trajectory. Nevertheless, the user can rotate it, in any desired yaw angle.
- No sensors for detecting obstacles are embedded into the drone. The OptiTrack system does not compute also the obstacles positions. Therefore, large errors in the virtual commands should produce the crash of the robot with the obstacles.

A video with the experimental results can be seen at: https://youtu.be/jMYWIoCsj7I.

A. Scenario 1: crossing reduced spaces for rescuing

As previously explained, in this scenario, the vehicle must cross through some reduced spaces that can produces several stress when the user drives the aerial drone in direct view. For this scenario, two physical windows are located in the flight arena in different altitudes, positions and headings. These windows are also created in the WIM with the same characteristics. The first one is located at $\xi_{w_1} = (2, 2, 1.4)$m and the second one at $\xi_{w_2} = (-0.5, -1.7, 1.7)$m in the flight area. The test consists as follows: the user must move the virtual vehicle in any desired path while crossing the windows and repeating the experiment three times with different translational velocities. The real robot must imitate the same performance of the virtual drone in real time as can be seen in Figure 10.

From Figures 11 - 14, the states performance of the robots (real and virtual) when validating the scenario are depicted.

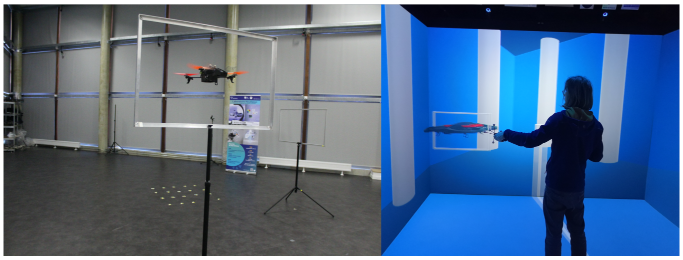


Fig. 10: Setup of the first experimental scenario.

In Figure 11, the translation motion in 3D is illustrated. Notice from this figure the correct match between worlds and frames. In addition, readers could interpret, also from this figure, that the controller is not capable to track the references coming from the virtual drone (visible differences between the reference and the real drone's performance). Nevertheless, in Figure 12 a detailed description of the experiment in each axis is depicted. Therefore, observe from this figure the good performance of the controller when tracking the references. For doing the task the user needs to change the drone's heading and the altitude for crossing the windows (windows are in different altitude), then, observe from Figures 13 and 12-down, these user movements (dotted line) and how the real drone (solid line) follows it, testifying the good match between worlds and frames.

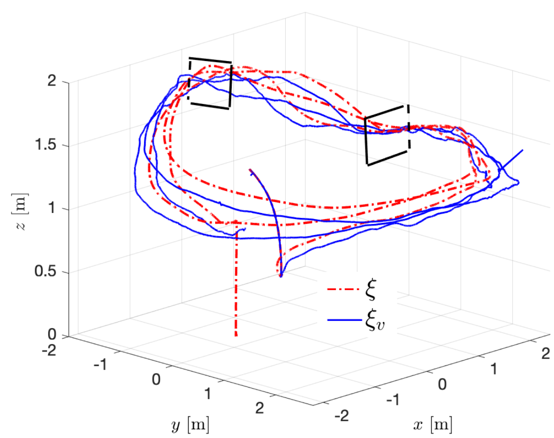


Fig. 11: 3D performance of the virtual drone and the real vehicle when the first scenario is validated in real time.

In this experiment, from $0 < t_1 \leq 5s$ the real vehicle takes off autonomously to be placed at 1.4m in z-axis. Once the vehicle switch to DrEAM's mode, it is posed in an altitude of 0.5m. At time $t = 10$s the user in DrEAM takes the virtual drone and manipulates it making a trajectory and repeating it three times at different velocities. The three laps are done at times: $10 < t_1 \leq 40s$, $40 < t_2 \leq 60s$ and $60 < t_3 \leq 80s$, respectively. The first one was performed with a slower velocity because the user did not have much confidence and he was tense. The second lap, the user increases the velocity of the trajectory (up to 1m/s) and in the last one, the user tries to do the displacement as fast as possible. The variations in the velocity can be checked in Figure 14. In figures, the

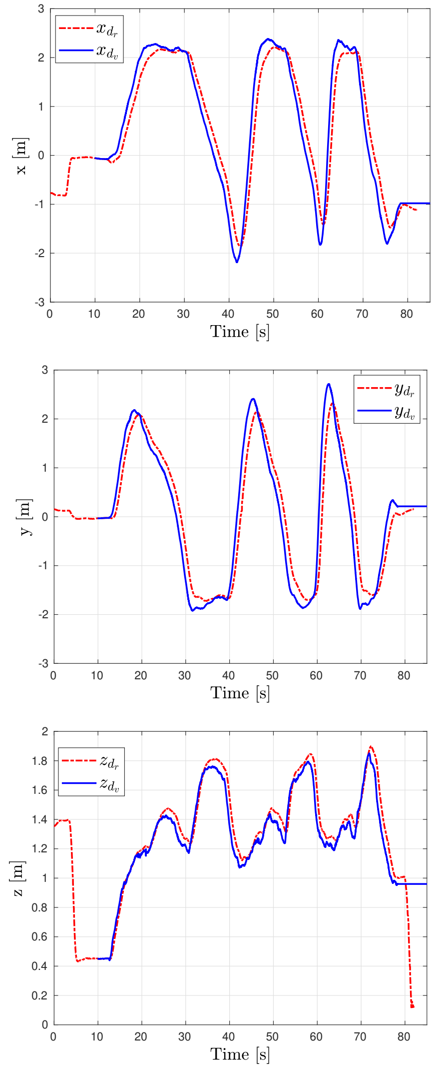


Fig. 12: x, y, z-position performances of the virtual drone and the real vehicle obtained from the first scenario. Observer the good behavior of the real drone when imitates the virtual drone trajectory.

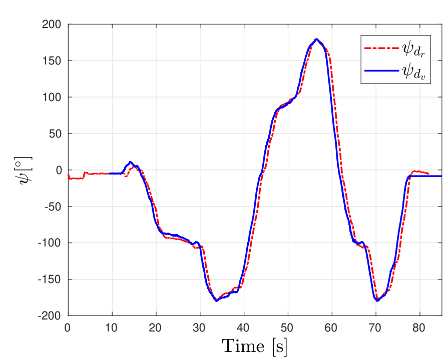


Fig. 13: Yaw angle behavior of the drones (real and virtual).

subscripts d_r and d_v represent the state of the real drone and the virtual drone respectively.

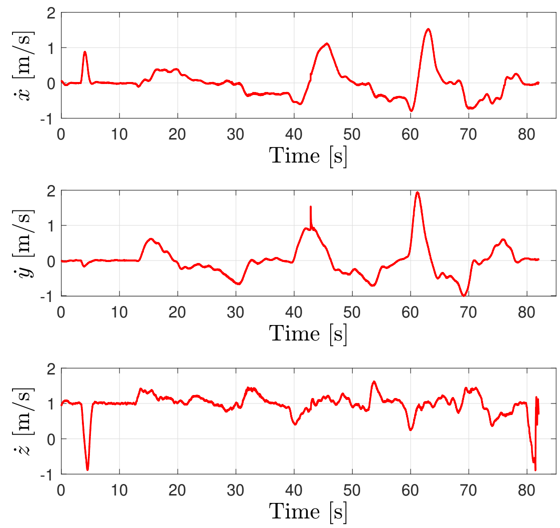


Fig. 14: Performance of the velocities of vehicle during the first scenario. Note that it is increased mainly in the x and y axes, which could be interpreted as better user comfort during testing.

B. Scenario 2: Building inspection

In this scenario, two tripods are required in the flight arena emulating pillars in the cellar. In DrEAM, two cylinders were drawn to simulate the corresponding 'pillars' as can be seen in Figure 15. The practical goal is to illustrate the maneuverability of DrEAM when the user manipulates the virtual drone between the emulated pillars and changing its heading. The experiment is repeated several times with different maneuvers trying to approach the drone as near as possible to the 'pillars'.

The performance of the states when the real drone follows the position references of the virtual vehicle are shown in Figures 16 - 18. A 3D representation of the trajectories is presented in Figure 16. As in the first scenario, the initial step was to take off the real vehicle autonomously in safety mode. In this mode the real drone is placed at an altitude of 1.4m and a few seconds later its altitude is reduced at 0.5m in hover position, ready to follows the virtual commands. This is done in the interval time $0 < t_0 \leq 20$s.

As depicted in Figure 16 the goal of the experiment was that the user manipulates the virtual drone between the pillars. Three laps were carried out by the user at different times;

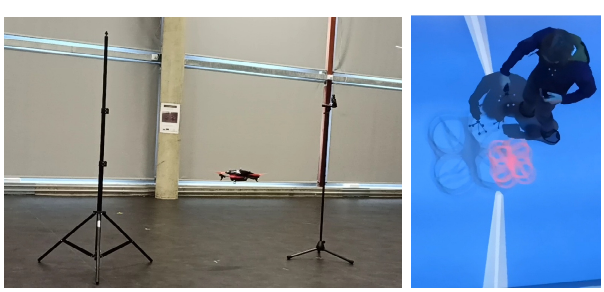


Fig. 15: Setup of the second experimental scenario.

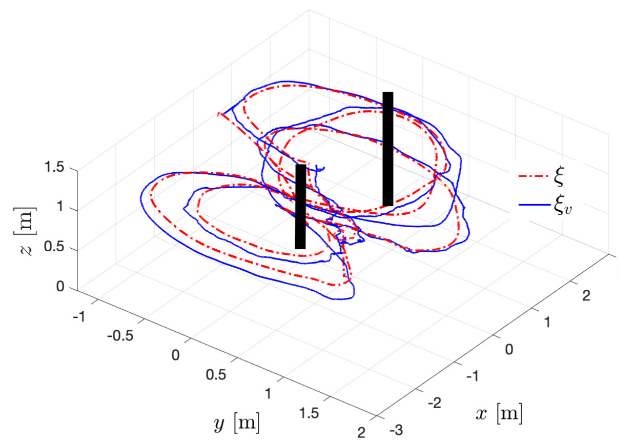


Fig. 16: 3D behavior of the virtual and the real aerial vehicles when performing the second scenario.

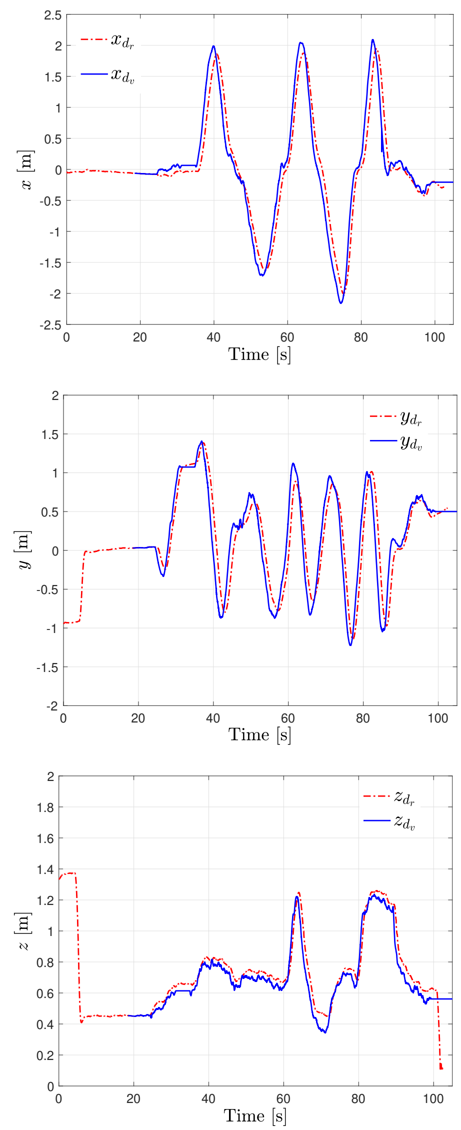


Fig. 17: Performance of the aerial robots in the x, y, z axes. Observe the good performance when the real robot imitates the virtual robot.

$20 < t_1 \leq 55$s for the first one, $55 < t_2 \leq 80$s for the second lap and $80 < t_3 \leq 100$s for the last one. This task was made in an area of $-2 \leq x \leq 2$, $-1.5 \leq y \leq 1.5$ and $0.5 \leq z \leq 1.4$, all in meters. Moreover, the mission demands to change the heading of the aerial robots several times as is illustrated in Figure 18. From these figures, observe the good performance of the real robot when it is controlled with the DrEAM architecture. Remark that the task has been developed with good precision, that it should not be possible when the vehicle is controlled in direct view. Therefore, it was proved experimentally (and shown in graphs) that a neophyte user can do complex missions.

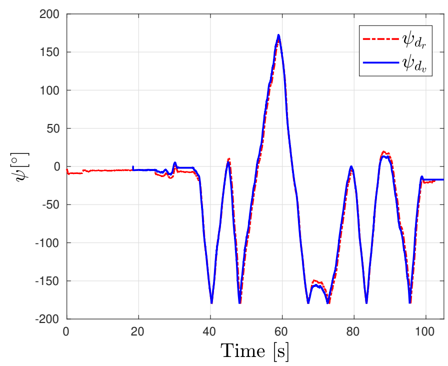


Fig. 18: Behavior of the heading of the virtual and real drone respectively.

VI. Conclusion

A new virtual control architecture for remote control robot was proposed in this paper. The architecture is composed of a new metaphor interaction called DrEAM (Drone Exocentric Advanced Metaphor) that uses a world-in-miniature located in a virtual environment, and a real aerial robot with an embedded controller based on quaternion formulation. The testing rooms (for the real and virtual drone) were connected using User Datagram Protocol (UDP) with their respective routers connected in the same network. The user controlled manually a virtual robot in DrEAM while the real robot imitate autonomously in real time the trajectory described by the user with the virtual drone. Two experiments were carried out for validating the proposed architecture. The first one was done for analyzing user's experimental fatigue when controlling a drone in direct view and using DrEAM. The participants were novice pilots without experience in drone flight or immersed environments. An easy task was imposed to each participant and results let see that DrEAM could be an interesting and viable solution for controlling robots without be in direct view of it.

The other experiments were done for validating robotics applications using a neophyte pilot. Here, experimental results illustrated the easy implementation and feasibility of the mission using DrEAM. In addition, the novice pilot was capable to do complex missions with good precision. From all the experiments, a reduction of the cognitive overload was observed when using DrEAM. Moreover, it was demonstrated, in the real-time experiments, the easy maneuverability and controllability of the real drone using DrEAM.

From experimental tests, we have noticed that the system's performance is appropriate for indoor applications, nevertheless for outdoor applications possible challenges can appear (static/dynamics objects in the navigation trajectory, delay in the communication, errors in the GPS measurements, windy conditions, etc.) that need to be solved. Therefore, from a robotics point of view, the major challenge will be to design a new control architecture including observers and predictors for giving robustness properties to the system for navigating in real scenarios (dynamic, unstructured and windy environments). From a VR point of view (and considering real scenarios) the challenge will be to propose an optimized exocentric metaphor for robot's control taking into account different parameters as, the size of the virtual and phantom drone, the device used in the VE for taking the drone (comfort), the control algorithms in the virtual environment, the network issues, the model of the environment, etc. We consider that new applications focused on the virtualization of the geometric world such as Google earth, and all the 3D scans of buildings and landscapes could help us in the future to navigate in virtual worlds reconstructed from real ones, i.e., the virtual environment could be modeled easily.

References

[1] V. J. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 194–203, 1993.

[2] M. Bergamasco, B. Allotta, L. Bosio, L. Ferretti, G. Parrini, G. Prisco, F. Salsedo, and G. Sartini, "An arm exoskeleton system for teleoperation and virtual environments applications," in *Proc. IEEE International Conference on Robotics and Automation (ICRA), San Diego, CA, USA*, 1994, pp. 1449–1454.

[3] C. Stanton, A. Bogdanovych, and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning," in *Proc. Australasian Conference on Robotics and Automation (ACRA), Victoria University of Wellington, New Zealand*, 2012.

[4] H. Hirukawa, F. Kanehiro, K. Kaneko, S. Kajita, K. Fujiwara, Y. Kawai, F. Tomita, S. Hirai, K. Tanie, T. Isozumi *et al.*, "Humanoid robotics platforms developed in hrp," *Robotics and Autonomous Systems*, vol. 48, no. 4, pp. 165–175, 2004.

[5] D. Cheng-jun, D. Ping, Z. Ming-lu, and Z. Yan-fang, "Design of mobile robot teleoperation system based on virtual reality," in *2009 IEEE International Conference on Automation and Logistics, Shenyang, China*, 2009, pp. 2024–2029.

[6] Dongjun Lee, O. Martinez-Palafox, and M. W. Spong, "Bilateral teleoperation of a wheeled mobile robot over delayed communication network," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, 2006, pp. 4193–4198.

[7] D.-S. Kwon, J.-H. Ryu, P.-M. Lee, and S.-W. Hong, "Design of a teleoperation controller for an underwater manipulator," in *IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA*, vol. 4, 2000, pp. 3114–3119.

[8] Q. Lin and C. Kuo, "Virtual tele-operation of underwater robots," in *Proceedings of International Conference on Robotics and Automation, Albuquerque, NM, USA*, vol. 2, 1997, pp. 1022–1027.

[9] E. Gray, *Nineteenth-century torpedoes and their inventors*. Annapolis, Md.: Naval Institute Press ; London : Greenhill, 2004.

[10] R. R. Murphy and J. L. Burke, "Up from the rubble: Lessons learned about hri from search and rescue," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 3, pp. 437–441, 2005.

[11] M. R. Endsley, "Design and evaluation for situation awareness enhancement," *Proceedings of the Human Factors Society Annual Meeting*, vol. 32, no. 2, pp. 97–101, 1988.

[12] G. Gioioso, M. Mohammadi, A. Franchi, and D. Prattichizzo, “A force-based bilateral teleoperation framework for aerial robots in contact with the environment,” in *2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA*, 2015, pp. 318–324.

[13] J. Li, B. You, L. Ding, J. Xu, W. Li, H. Chen, and H. Gao, “A novel bilateral haptic teleoperation approach for hexapod robot walking and manipulating with legs,” *Robotics and Autonomous Systems*, vol. 108, pp. 1 – 12, 2018.

[14] C. A. López Martínez, I. Polat, R. v. d. Molengraft, and M. Steinbuch, “Robust high performance bilateral teleoperation under bounded time-varying dynamics,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 206–218, 2015.

[15] D. Santiago, E. Slawinski, and V. Mut, “Human-inspired stable bilateral teleoperation of mobile manipulators,” *ISA Transactions*, vol. 95, pp. 392 – 404, 2019.

[16] Y. Ye, Y. Pan, and T. Hilliard, “Bilateral teleoperation with time-varying delay: A communication channel passification approach,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1431–1434, 2013.

[17] S.-J. Lee and H.-S. Ahn, “Controller designs for bilateral teleoperation with input saturation,” *Control Engineering Practice*, vol. 33, pp. 35 – 47, 2014.

[18] D. Mercado, P. Castillo, and R. Lozano, “Sliding mode collision-free navigation for quadrotors using monocular vision,” *Robotica*, vol. 36, no. 10, pp. 1493–1509, 2018.

[19] A. Kelly, E. Capstick, D. Huber, H. Herman, P. Rander, and R. Warner, “Real-time photorealistic virtualized reality interface for remote mobile robot control,” in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 211–226.

[20] D. Saakes, V. Choudhary, D. Sakamoto, M. Inami, and T. Lgarashi, “A teleoperating interface for ground vehicles using autonomous flying cameras,” in *2013 23rd International Conference on Artificial Reality and Telexistence (ICAT)*, 2013, pp. 13–19.

[21] K. Saitoh, T. Machida, K. Kiyokawa, and H. Takemura, “A 2d-3d integrated interface for mobile robot control using omnidirectional images and 3d geometric models,” in *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2006, pp. 173–176.

[22] R. Mahony, F. Schill, P. Corke, and Y. S. Oh, “A new framework for force feedback teleoperation of robotic vehicles based on optical flow,” in *2009 IEEE International Conference on Robotics and Automation, Kobe, Japan*, 2009, pp. 1079–1085.

[23] H. Rifaï, M.-D. Hua, T. Hamel, and P. Morin, “Haptic-based bilateral teleoperation of underactuated unmanned aerial vehicles,” *Proceedings of the 18th World Congress, Milano, Italy*, vol. 44, no. 1, pp. 13 782 – 13 788, 2011.

[24] X. Hou, H. Lan, X. Xing, Y. Qu, D. Yuan, J. Yan, and P. Huang, “Environmental force reflection in an admittance configured haptic interface for teleoperation of vtol aerial robots,” *20th IFAC World Congress, Toulouse, France*, vol. 50, no. 1, pp. 10 262 – 10 267, 2017.

[25] E. Slawiñski, D. Santiago, and V. Mut, “Control for delayed bilateral teleoperation of a quadrotor,” *ISA Transactions*, vol. 71, pp. 415 – 425, 2017.

[26] R. A. S. Fernández, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy, “Natural user interfaces for human-drone multi-modal interaction,” in *2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA*, 2016, pp. 1013–1022.

[27] J. Regenbrecht, A. Tavakkoli, and D. Loffredo, “A robust and intuitive 3d interface for teleoperation of autonomous robotic agents through immersive virtual reality environments,” in *2017 IEEE Symposium on 3D User Interfaces (3DUI), Los Angeles, CA, USA*, 2017, pp. 199–200.

[28] T. Abut and S. Soygüder, “Haptic industrial robot control and bilateral teleoperation by using a virtual visual interface,” in *2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey*, 2018, pp. 1–4.

[29] A. Naceri, D. Mazzanti, J. Bimbo, D. Prattichizzo, D. G. Caldwell, L. S. Mattos, and N. Deshpande, “Towards a virtual reality interface for remote robotic teleoperation,” in *2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, Brazil*, 2019, pp. 284–289.

[30] S. N. F. Nahri, S. Du, and B. Van Wyk, “Haptic system interface design and modelling for bilateral teleoperation systems,” in *2020 International SAUPEC/RobMech/PRASA Conference, Cape Town, South Africa, South Africa*, 2020, pp. 1–6.

[31] J. M. Teixeira, R. Ferreira, M. Santos, and V. Teichrieb, “Teleoperation using google glass and ar, drone for structural inspection,” in *XVI Symposium on Virtual and Augmented Reality*. IEEE, 2014, pp. 28–36.

[32] F. Okura, Y. Ueda, T. Sato, and N. Yokoya, “[paper] free-viewpoint mobile robot teleoperation interface using view-dependent geometry and texture,” *ITE Transactions on Media Technology and Applications*, vol. 2, no. 1, pp. 82–93, 2014.

[33] J. Thomason, P. Ratsamee, J. Orlosky, K. Kiyokawa, T. Mashita, Y. Uranishi, and H. Takemura, “A comparison of adaptive view techniques for exploratory 3d drone teleoperation,” *ACM Trans. Interact. Intell. Syst.*, vol. 9, no. 2–3, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3232232

[34] R. Temma, K. Takashima, K. Fujita, K. Sueda, and Y. Kitamura, “Third-person piloting: Increasing situational awareness using a spatially coupled second drone,” in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 507–519. [Online]. Available: https://doi.org/10.1145/3332165.3347953

[35] M. Chen, W. Saad, and C. Yin, “Echo-liquid state deep learning for 360 content transmission and caching in wireless VR networks with cellular-connected uavs,” *CoRR*, vol. abs/1804.03284, 2018. [Online]. Available: http://arxiv.org/abs/1804.03284

[36] A. Suarez, P. Sanchez-Cuevas, M. Fernandez, M. Perez, G. Heredia, and A. Ollero, “Lightweight and compliant long reach aerial manipulator for inspection operations,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain*, 2018, pp. 6746–6752.

[37] J.-S. Lin and Z.-Y. Jiang, “Implementing remote presence using quadcopter control by a non-invasive bci device,” *Computer Science and Information Technology*, vol. 3, no. 4, pp. 122–126, 2015.

[38] M. Macchini, T. Havy, A. Weber, F. Schiano, and D. Floreano, “Hand-worn haptic interface for drone teleoperation,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA), Paris, France*, 2020.

[39] A. H. Sanchez, L.F. and P. Castillo, “User-robot interaction for safe navigation of a quadrotor,” *Robotica*, vol. 38, no. 12, pp. 2189 – 2203, 2020.

[40] N. Liu, T. Lu, Y. Cai, J. Lu, H. Gao, B. Li, and S. Wang, “Design of virtual reality teleoperation system for robot complex manipulation,” in *2019 Chinese Automation Congress (CAC), Hangzhou, China, China*, 2019, pp. 1789–1793.

[41] S. Kim, Y. Kim, J. Ha, and S. Jo, “Mapping system with virtual reality for mobile robot teleoperation,” in *2018 18th International Conference on Control, Automation and Systems (ICCAS), Daegwallyeong, South Korea*, 2018, pp. 1541–1541.

[42] A. Yew, S. Ong, and A. Nee, “Immersive augmented reality environment for the teleoperation of maintenance robots,” *Procedia CIRP*, vol. 61, pp. 305 – 310, 2017, the 24th CIRP Conference on Life Cycle Engineering.

[43] O. Erat, W. A. Isop, D. Kalkofen, and D. Schmalstieg, “Drone-augmented human vision: Exocentric control for drones exploring hidden areas,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1437–1446, 2018.

[44] B. Walter, J. Knutzon, A. Sannier, and J. Oliver, “Virtual uav ground control station,” in *AIAA 3rd” Unmanned Unlimited” Technical Conference, Workshop and Exhibit, Chicago, Illinois*, 2004, p. 6320.

[45] B. Huang, D. Bayazit, D. Ullman, N. Gopalan, and S. Tellex, “Flight, camera, action! using natural language and mixed reality to control a drone,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada*, 2019.

[46] N. Li, S. Cartwright, A. Shekhar Nittala, E. Sharlin, and M. Costa Sousa, “Flying frustum: A spatial interface for enhancing human-uav awareness,” in *Proceedings of the 3rd International Conference on Human-Agent Interaction*, ser. HAI ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 27–31. [Online]. Available: https://doi.org/10.1145/2814940.2814956

[47] K. Yonezawa and T. Ogawa, “Flying robot manipulation system using a virtual plane,” in *2015 IEEE Virtual Reality (VR)*, 2015, pp. 313–314.

[48] J. Thomason, P. Ratsamee, K. Kiyokawa, P. Kriangkomol, J. Orlosky, T. Mashita, Y. Uranishi, and H. Takemura, “Adaptive view management for drone teleoperation in complex 3d structures,” in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, ser. IUI ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 419–426. [Online]. Available: https://doi.org/10.1145/3025171.3025179

[49] Dreamstime, url: https://www.dreamstime.com, accessed 20-04-2020.

[50] G. Sanahuja and et al., “Fl-air framework libre air,” urlhttps://devel.hds.utc.fr/software/flair, accessed 29-07-2019.

[51] E. Ibarra-Jimenez and P. Castillo, “Aerial autonomous catching ball using a nested second order sliding mode control,” *20th IFAC World Congress, Toulouse, France*, vol. 50, no. 1, pp. 11 415–11 420, 2017.

[52] H. Abaunza, E. Ibarra, P. Castillo, and A. Victorino, "Quaternion based control for circular uav trajectory tracking, following a ground vehicle: Real-time validation," *20th IFAC World Congress, Toulouse, France*, vol. 50, no. 1, pp. 11 453 – 11 458, 2017.

[53] M. Faessler, D. Falanga, and D. Scaramuzza, "Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 476–482, 2017.

[54] A. Castillo, R. Sanz, P. Garcia, W. Qiu, H. Wang, and C. Xu, "Disturbance observer-based quadrotor attitude tracking control for aggressive maneuvers," *Control Engineering Practice*, vol. 82, pp. 14 – 23, 2019.

[55] A. Chovancová, T. Fico, P. Hubinský, and F. Duchoň, "Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator," *Robotics and Autonomous Systems*, vol. 79, pp. 87 – 98, 2016.

[56] H. Abaunza, E. Ibarra, P. Castillo, and A. Victorino, "Quaternion based control for circular uav trajectory tracking, following a ground vehicle: Real-time validation," *20th IFAC World Congress, Toulouse, France*, vol. 50, pp. 11 453–11 458, 2017.

[57] J. Cariño, H. Abaunza, and P. Castillo, "Quadrotor quaternion control," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA*, 2015, pp. 825–831.

[58] L. Fricoteaux, I. Thouvenin, and M. Daniel, "Gulliver: A decision-making system based on user observation for an adaptive training in informed virtual environments," *Eng. Appl. Artif. Intell*, vol. 33, pp. 47–57, 2014.

[59] R. Frenoy, Y. Soullard, I. Thouvenin, and O. Gapenne, "Adaptive training environment without prior knowledge: Modeling feedback selection as a multi-armed bandit problem," in *UMAP*, 2016, pp. 131–139.

[60] E. Loup-Escande, R. Frenoy, G. Poplimont, I. Thouvenin, O. Gapenne, and O. Megalakaki, "Contributions of mixed reality in a calligraphy learning task: Effects of supplementary visual feedback and expertise on cognitive load, user experience and gestural performance," *Comput. Hum. Behav*, vol. 75, pp. 42–49, 2017.

[61] F. Jeanne, Y. Soullard, A. Oker, and I. Thouvenin, "Ebagg: Error-based assistance for gesture guidance in virtual environments," in *ICALT*, 2017, pp. 472–476.