Cooperative Control of UGV-UAV with Focus on Teaching-Learning Processes.

Moreno Lovato, Alba Margarita y Velasco Quispe, Miguel Alexander.

Departamento de Eléctrica, Electrónica y Telecomunicaciones.

Carrera de Ingeniería en Electrónica e Instrumentación.

Artículo académico, previo a la obtención del título de Ingeniero en Electrónica e Instrumentación.

Mgs. Ortiz Moreano, Jessica Sofía.

08 de agosto del 2023

Latacunga

Cooperative Control of UGV-UAV with Focus on Teaching-Learning Processes

Miguel A. Velasco, Alba M. Moreno, Jessica S. Ortiz
and Víctor H. Andaluz

Universidad de las Fuerzas Armadas ESPE, Sangolquí Ecuador

{mavelasco8, ammoreno4, jsortiz4, vhandaluz1}@espe.edu.ec

Abstract. This article presents an application based on the learning of cooperative control of robots, for its development is used a 3D environment made in the UNITY 3D software; and a mathematical simulation software MATLAB where advanced control algorithms are developed, this with the objective that the unmanned ground vehicle (UGV) and the unmanned aerial vehicle (UAV) execute an autonomous locomotion and can perform cooperative tasks, analyzing the stability of the proposed control based on the behavior of control errors and showing the effectiveness of the controller, as in previous works are shown training controllers between robot of the same type, in this case the training control is carried out between heterogeneous robots. Finally, the experimental tests are presented with a user iteration and immersion in the virtual environment where the results obtained from the proposed cooperative control are validated, in order to execute the best design of the cooperative controller in the physical omnidirectional terrestrial and unmanned aerial mobile robots obtaining a significant cost reduction in the development of the experimental tests.

Keywords: Learning, Cooperative Control, UNITY 3D, Advanced Control.

1 Introduction

Robotics, due to its diverse applications, has been focused in multiple research works as defined in [1],[2],[3]. Robots are found in various fields of knowledge and research: *i)* space exploration, with robots designed for territory reconnaissance and sample collection, such as NASA's Curiosity robot. [2]; *ii)* industrial sector, using robot arms focused on infrastructure construction with higher speed and precision, which are visually controlled, such as the KUKA robot from KUKA AG [3]; *iii)* in health sector, assistance, rehabilitation and operation robots such as the DaVinci Ro-bot have been created. This robot has presented robots that interact with great accuracy and precision in the intervention with patients [4]; *vi)* in security, QinetiQ's Talon robot has been developed for surveillance on land and in water [5]; *vii)* in transport logistics, there are robots

with mobile autonomy capable of navigating in or out of storage facilities for merchandise, such as RB-THERON robot, and among others. Robot manipulation tasks and robot control can be performed autonomously, semi-autonomously or tele-operated, which requires the design of advanced controllers. These controllers are developed under the knowledge of kinematic and/or dynamic characteristics of the robot [7]. In this context we can categorize two research groups defined for mobility, terrestrial or aerial. First the Unmanned Ground Vehicles (UGV), which operates on the ground, *i.e*. on surfaces such as soil, pavement or grass[4], to move in their environment are equipped with wheels, tracks or legs [8][9]; and secondly, Unmanned Aerial Vehicles (UAV) [7], which are aircraft that can be remotely controlled or fly autonomously using a reprogrammed automation system [5]. These robots are a versatile tool for solving complex mobility tasks in various fields because each robot has different locomotion capabilities. Therefore, several works [10] focus on the development of cooperative controllers for these robots [6],[7],[8].

Cooperative control is based on the coordinated work of two or more robots to execute different tasks together, sharing information and making decisions in real time. Its purpose is to improve effectiveness and efficiency by achieving a common goal [12],[5]. For decision-making can be executed in two ways: I) Centralized Control, where the robots depend on a central unit (computer) who is mainly in charge of the control of the robots that are part of it [13]; and II) Decentralized Control this type of control allows the robots to work independently, *i.e*., the actions of one of them should not be connected with the robots that are part of the task which allows the system to be faster and more versatile [14]. For the design of cooperative control, a cascade formation scheme is used. The centralized part of the controller is in responsible for the training between robots, while the decentralized part of the individual kinematic control of each robot, by dividing into these two parts, it is possible to achieve greater efficiency and coordination of each robot allowing that in case of failure of the central control do not lose control over their movement [7], [9],[10].

In the study and development of this type of controllers, it is essential to have the physical robots to evaluate the performance of the controller [14],[11] however, the cost of each robot is high, therefore, it is not accessible in the educational field. The implementation and development of a simulation environment turns out to be helpful for the user to test the control algorithms, and especially by having a virtual environment, it allows realistically validating the controller and behavior of the robots [15],[12].

This work proposes to implement a heterogeneous cooperative control algorithm between a UGV ground robot and a UAV unmanned aerial vehicle which allows complex tasks to be carried out through the coordination of the UGV with the UAV [13]. In addition, a virtual environment is developed and implemented that allows the immersion and interaction of the user with the cooperative system to evaluate the proposed controller. The virtual environments have to be as exact as possible for their proper use and the evaluation of the errors for the correct corrections of the controller.[14],[15] With this, the implementation of the controller in the robots in the real part becomes safer and more reliable. This leads to reducing implementation costs and different tests.

This article is organized as follows: In Section 2, conceptualization of the process, while in Section 3 the mathematical modeling is described, subdivided into the terrestrial and aerial kinematics section, in Section 4 the controller design in conjunction with stability analysis; Section 5 presents the virtualization of the simulation environment; and finally, the results are presented in Section 6.

2 Process Conceptualization

This section presents the methodology implemented for cooperative autonomous control between an aerial robot and a ground robot. Figure 1 shows the proposed methodology, which considers six main stages: digitization resources, control scheme, virtual simulation, and experimentation environment.

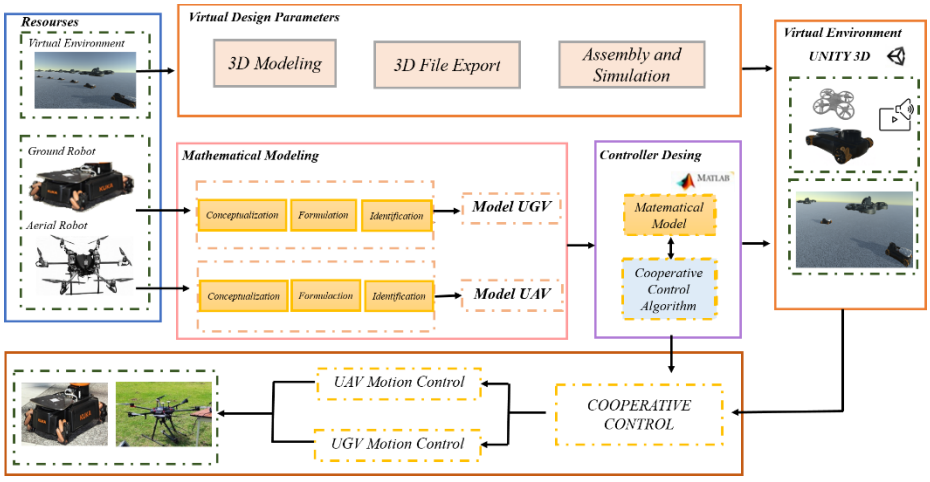

Fig. 1. Process conceptualization,

(*i*) ***Resources***, at this stage the resources needed for the development of the proposal are presented, such as: taking into account the characteristics and properties of the virtual reality environment; the ground robot to be used; and the aerial robot; *(ii)* ***Digitalization***, this stage is responsible for digitizing external resources (Robots, environments, users), in order to be incorporated into a 3D virtual simulation environment. For digitalization, the 3D models of each robot and its movement characteristics are mainly considered.; *(iii)* ***Mathematical modeling***, this is responsible of determining the mathematical models that represent the restrictions and characteristics of movement of each robot. Independent modeling processes are considered for each robot, however, for each robot the sub-stages of conceptualization, theoretical formulation, and identification-validation of robot dynamic parameters are considered; ***(iv) Control scheme,*** at this stage the design of the cooperative control algorithm is carried out, in order to execute autonomous tasks between a UGV and UAV**.** In addition, the mathematical analysis of

the evolution of cooperative control errors is considered.; ***(v) Virtual simulation***, Here the implementation of the proposed control scheme is considered to analyze the behavior of robots in a virtual environment when executing autonomous cooperative tasks. The virtual simulator is implemented in the Unity3D graphics engine and considers the user's interaction with the environment through virtual reality devices, such as: HTC Vive, Oculus Quest 2, among others; and finally *(vi)* ***Experimentation environment***, the control algorithms proposed and previously validated in the virtual simulator are implemented experimentally.

3 Mathematical Modeling

The mathematical model of the robot allows describing the motion of the robot over the working space. The model considers the movement restrictions of the terrestrial robot and the aerial robot. The mathematical model used in this work is the kinematic model, which allows the relation of the robot maneuvering velocities with the velocities of the working space or in other words relates the operational coordinates of each robot with the generalized coordinates

3.1 UGV Kinematic Model

Figure 2 shows the kinematic configuration of the UGV on the fixed workspace $\{R\}$. For this case study an omnidirectional terrestrial robot is considered.

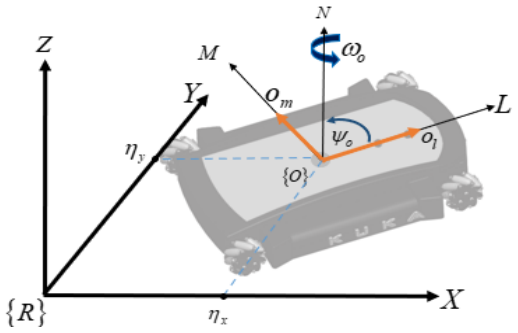

Fig. 2. Kinematic configuration of the UGV.

The kinematic model of the omnidirectional robot is composed of three velocities, o_l, o_m and ω_o. Where, o_l is the frontal linear velocity; o_m is the lateral linear velocity; and ω_o is the angular velocity that rotates around the N axis. These velocities are defined within the moving reference frame of the robot. The motion of the robot about the fixed reference system $\{R\}$ is defined by:

$$\begin{bmatrix} \dot{\eta}_x \\ \dot{\eta}_y \\ \dot{\eta}_\psi \end{bmatrix} = \begin{bmatrix} \cos(\psi_o) & -\sin(\psi_o) & 0 \\ \sin(\psi_o) & \cos(\psi_o) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} o_l \\ o_m \\ \omega_o \end{bmatrix} \tag{1}$$

Writing in matrix form the equation (1), which represents the kinematic model of the omnidirectional robot results:

$$\dot{\boldsymbol{\eta}}(t) = \mathbf{J}_{\mathbf{o}}(\psi_o)\,\mathbf{v}_{\mathbf{o}}(t) \tag{2}$$

where, $\dot{\boldsymbol{\eta}}(t) \in \Re^3$ represents the velocity vector at the interest point of the robot with respect to the fixed inertial reference frame $\{R\}$; $\mathbf{v}_{\mathbf{o}}(t) \in \Re^3$ is the vector of robot maneuverability velocities with respect to the moving reference system $\{O\}$; and $\mathbf{J}_{\mathbf{o}}(\psi_o) \in \Re^{3x3}$ is the Jacobian matrix of the omnidirectional robot that represents the motion characteristics of the robot and maps maneuverability velocities of the robot to motion velocities within the fixed reference frame $\{R\}$.

3.2 UAV Kinematic Model

Figure 2 shows the kinematic configuration of the UAV robot to move in 3D space as a function their maneuverability velocities. In this case study, a quadrotor is considered.

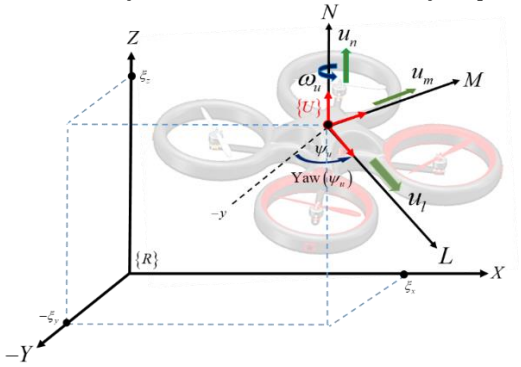

Fig. 3. Kinematic configuration of the UAV

The UAV robot considered in this work has four maneuverability velocities described in the mobile reference system $\{U\}$. Three linear velocities: u_l, u_m and u_n that allow the lateral, frontal and elevation displacement in the 3D space respectively and an angular velocity ω_u that allows to rotate the UAV around the N axis of the mobile reference system $\{U\}$. To obtain the kinematic model of the UAV, it is considered that it works at small velocities, therefore, the pitch (θ) and roll (ϕ) angles are zero. Then, the kinematic model describing the motion in space of the UAV is represented by

$$\begin{bmatrix} \dot{\xi}_x \\ \dot{\xi}_y \\ \dot{\xi}_z \\ \dot{\xi}_\psi \end{bmatrix} = \begin{bmatrix} \cos(\psi_u) & -\sin(\psi_u) & 0 & 0 \\ \sin(\psi_u) & \cos(\psi_u) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_l \\ u_m \\ u_n \\ \omega_u \end{bmatrix} \tag{3}$$

Written in a matrix form (3), the kinematic model of the UAV can be denoted as:

$$\dot{\boldsymbol{\xi}}(t) = \mathbf{J}_{\mathbf{u}}(\psi_u)\mathbf{v}_{\mathbf{u}}(t) \tag{4}$$

where, $\dot{\boldsymbol{\xi}}(t) \in \Re^4$ represents the velocity vector of the interest point with respect to the inertial reference frame $\{R\}$; $\mathbf{v}_{\mathbf{u}} \in \Re^4$ is the maneuverability vector of the UAV with respect to the moving reference frame $\{U\}$; and $\mathbf{J}_{\mathbf{u}}(\psi_u) \in \Re^{4x4}$ is the Jacobian matrix that represents the motion characteristics of the UAV and maps the UAV velocities to velocities in the fixed frame $\{R\}$.

4 Controller

Figure 4 details the multilayer scheme of the control strategy proposed for the cooperative control of two heterogeneous robots. The multilayer scheme is composed by five layers.

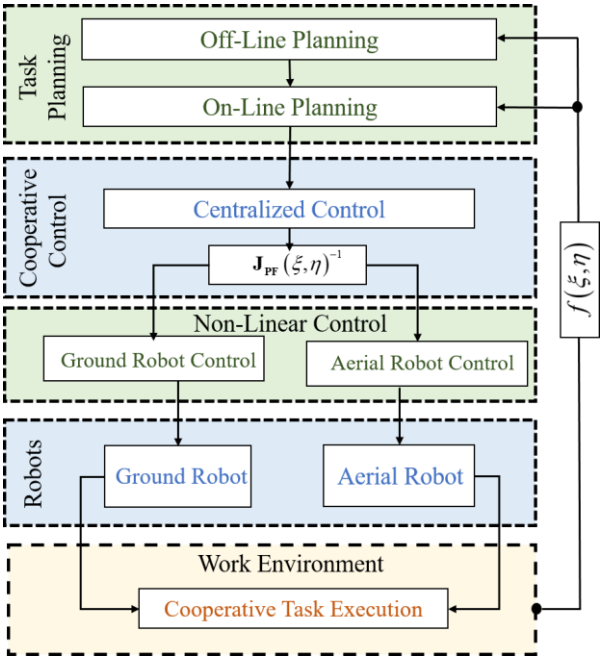

Fig. 4. Multilayer control scheme.

1. Task planning. In Off-Line planning, the task is defined, *i.e.*, the initial parameters of the robots are established; the desired structure of the cooperative control is specified, in other words, the desired shape and position of the structure; and the desired trajectory and velocity parameters for the formation are established. On-line planning allows to generate desired trajectory changes in order to avoid obstacles within the working environment. Changes of the formation structure, *e.g.*, passage through aisle, change of position of each robot, among other parameters that can modify the structure or velocity changes while the task is in progress.

2. Cooperative Control. This layer contains the cooperative controller, here the references for each robot are generated, that is, depending on the desired structure and depending on each current position of each robot, the controller is responsible for generating the references of positions and velocities for UGV and for the UAV. Thus, couples the operation of the two robots to achieve the desired goal.

3. Non-Linear Control. This layer performs the non-linear control of each robot, in other words, it is a decentralized control layer, which is responsible to accomplish that each robot fulfill the references provided by the previous layer of cooperative control. The controller used for each robot is a controller based on inverse kinematics.

4. Robots. This control layer contains the available robots (omnidirectional ground mobile robot and the unmanned aerial vehicle). In this layer the references of the maneuverability velocities of each robot are injected, allowing to observe their behavior in the working environment.

5. Working Enviroment. Finally, there is the working environment stage, where the structure of the environment is defined, in this case a partially structured environment is presented to observe the behavior of the robots in presence of various obstacles.

4.1 Control Scheme

Figure 5 shows the proposed control scheme for robot training (cooperative control). It consists of two sections: *i)* the shaping controller, which is responsible for generating the desired positions and velocities for each robot; and *ii)* the decentralized controller section, where the controller is set for each robot.

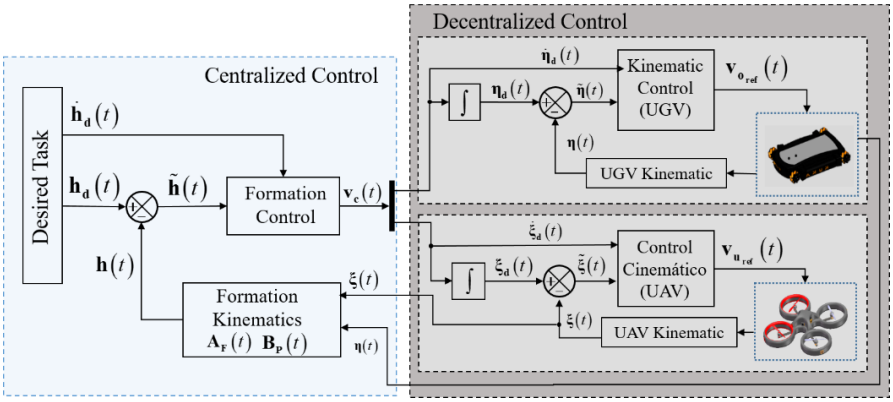

Fig. 5. Control scheme.

First, there is the shaping control, which results in a centralized control that unifies the kinematics of both robots and provides a control structure for the robots to execute a desired task together. In order to establish the formation controller, it is established to obtain the kinematics describing the position and shape of the cooperative control.

4.2 Formation Kinematics

The shaping structure proposed for this case study is defined on the workspace $\{R\}$. The shaping approach considers two aspects: *i)* the position of interest to be formed by the two robots, defined as $\mathbf{B}_\mathbf{p}$; and *ii)* the shape that this structure should be moved over the desired task $\mathbf{A}_\mathbf{F}$. Figure 6 presents the proposed shaping scheme.

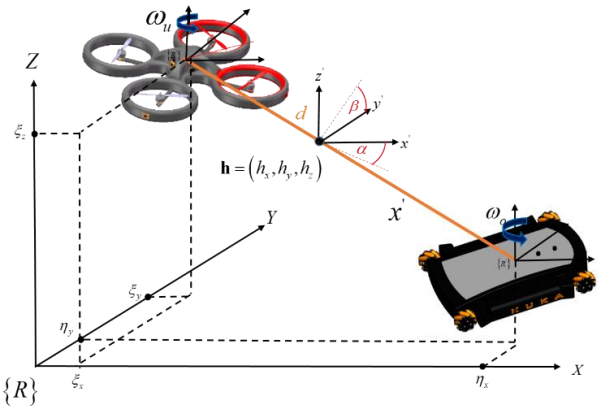

Fig. 6. Kinematics of the formation structure for cooperative control.

For the formation controller, the model that relates the velocities of each interest point of the robots to the velocities of the virtual formation is obtained. The direct kinematics of the virtual formation is defined by:

$$\mathbf{B}_\mathbf{p}(t) = \begin{bmatrix} h_x(t) \\ h_y(t) \\ h_z(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\eta_x + \xi_x) \\ \frac{1}{2}(\eta_y + \xi_y) \\ \frac{1}{2}(\eta_z + \xi_z) \end{bmatrix}$$

$$\mathbf{A}_\mathbf{F}(t) = \begin{bmatrix} d(t) \\ \alpha(t) \\ \beta(t) \end{bmatrix} = \begin{bmatrix} \sqrt{(\xi_x - \eta_x)^2 + (\xi_y - \eta_y)^2 + (\xi_z - \eta_z)^2} \\ \tan^{-1}\left(\frac{\xi_z - \eta_z}{\sqrt{(\xi_x - \eta_x)^2 + (\xi_y - \eta_y)^2}} \right) \\ \tan^{-1}\left(\frac{\xi_y - \eta_y}{\xi_x - \eta_x} \right) \end{bmatrix}$$

where, $\mathbf{B_P}(t) \in \Re^3$ is the vector that represents the center point of the formation in 3D space; and $\mathbf{A_F}(t) \in \Re^3$ is the vector of the virtual shape that contains the virtual distance between the two robots, the inclination angle $\alpha(t)$ and the orientation angle $\beta(t)$. Now, in order to get a model that allows to describe the formation as a function of the velocities of each robot, the partial derivative of $\mathbf{B_P}(t) \in \Re^3$ and $\mathbf{A_F}(t) \in \Re^3$ is determined, thus giving the kinematic model of the formation defined as follows:

$$\dot{\mathbf{B}}_\mathbf{p}(t) = \begin{bmatrix} \dot{h}_x(t) \\ \dot{h}_y(t) \\ \dot{h}_z(t) \end{bmatrix} = \mathbf{J_P}\mathbf{v_c}(t) \quad \dot{\mathbf{A}}_\mathbf{F}(t) = \begin{bmatrix} \dot{d}(t) \\ \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} = \mathbf{J_F}\mathbf{v_c}(t) \tag{5}$$

rewriting (6) in a compact form the kinematic model of the virtual formation results:

$$\dot{\mathbf{h}}(t) = \begin{bmatrix} \mathbf{J_P} \\ \mathbf{J_F} \end{bmatrix} = \mathbf{J_{PF}}\dot{\mathbf{h}}_\mathbf{c}(t) \tag{6}$$

where, $\dot{\mathbf{h}}(t) \in \Re^6$ is the velocities vector of the virtual formation; $\mathbf{h_c}(t) = \begin{bmatrix} \dot{\xi}_\mathbf{d}(t)^T & \dot{\eta}_\mathbf{d}(t)^T \end{bmatrix}^T \in \Re^5$ is the vector of reference velocities for each robot in the fixed frame $\{R\}$; and $\mathbf{J_{PF}} \in \Re^{6\times5}$ is the Jacobian matrix of the virtual formation that relates the velocities of the point of interest of each robot to the velocities of the formation in the 3D space.

4.3 Formation Controller (Centralized Control)

The proposed formation controller is based on inverse kinematics. The control objective is to drive the formation errors close to zero, in other words, $\lim_{t\to\infty}\tilde{\mathbf{h}}(t) = \lim_{t\to\infty}\left(\mathbf{h_d}(t) - \mathbf{h}(t)\right) \to \mathbf{0}$. Then, a control action must be obtained to carry out this objective, the proposed control law is defined by:

$$\mathbf{v_c}(t) = \mathbf{J}^+_{\mathbf{PF}}\left(\dot{\mathbf{h}}_\mathbf{d}(t) + \mathbf{K}\tilde{\mathbf{h}}(t)\right) \tag{7}$$

where, $\dot{\mathbf{h}}_\mathbf{d}(t)$ is the vector of the desired velocity of the task (Trajectory Tracking); $\mathbf{J}^+_{\mathbf{PF}} \in \Re^{5\times6}$ is the pseudoinverse matrix of $\mathbf{J_{PF}}$; $\tilde{\mathbf{h}}(t) = \begin{bmatrix} \tilde{\mathbf{B}}_\mathbf{p}{}^T & \tilde{\mathbf{A}}_\mathbf{F}{}^T \end{bmatrix}^T \in \Re^6$ is the vector of formation control errors; and $\mathbf{K} \in \Re^{6\times6}$ is a positive diagonal matrix that weighs the control errors. Now, the reference parameters for each robot are given by the vector $\mathbf{v_c}(t) = \begin{bmatrix} \dot{\eta}_\mathbf{d}(t)^T & \dot{\xi}_\mathbf{d}(t)^T \end{bmatrix}^T$ and the desired positions for each robot are determined by $\int \mathbf{v_c}(t)\,dt = \begin{bmatrix} \eta_\mathbf{d}(t)^T & \xi_\mathbf{d}(t)^T \end{bmatrix}^T$.

4.4 Nonlinear Controller (Decentralized Control)

The decentralized control block is responsible for each robot to comply with the references provided by the formation control block. For this purpose, each robot receives its velocity and position command, *i.e.*, the UGV and UAV receive the references and internally execute the controller to accomplish the desired task. The control law proposed for each robot is defined as a function of the inverse kinematics of each robot, the law of each robot is given by:

$$\mathbf{v}_{\mathbf{o}_{\mathbf{ref}}}(t) = \mathbf{J}_{\mathbf{o}}^{-1}\left(\dot{\boldsymbol{\eta}}_{\mathbf{d}}(t) + \mathbf{K}_{\mathbf{o}}\tilde{\boldsymbol{\eta}}(t)\right) \tag{8}$$

$$\mathbf{v}_{\mathbf{u}_{\mathbf{ref}}}(t) = \mathbf{J}_{\mathbf{u}}^{-1}\left(\dot{\boldsymbol{\xi}}_{\mathbf{d}}(t) + \mathbf{K}_{\mathbf{u}}\tilde{\boldsymbol{\xi}}(t)\right) \tag{9}$$

Equation (9) represents the controller for the omnidirectional robot (UGV). Where, $\dot{\boldsymbol{\eta}}_{\mathbf{d}}(t) \in \Re^3$ is the desired velocity vector; $\tilde{\boldsymbol{\eta}}(t) = \left[\tilde{\eta}_x, \tilde{\eta}_y, \tilde{\eta}_\psi\right]^T$ is the vector of control errors of UGV; $\mathbf{K}_{\mathbf{o}} \in \Re^{3\times3}$ is a diagonal matrix with positive values that weights the control errors; and $\mathbf{J}_{\mathbf{o}}^{-1} \in \Re^{3\times3}$ is Jacobian inverse matrix of the omnidirectional terrestrial robot. Finally, equation (10) is the controller for the UAV, where $\dot{\boldsymbol{\xi}}_{\mathbf{d}}(t) \in \Re^4$ is the desired velocity for UAV; $\tilde{\boldsymbol{\xi}}(t) = \left[\tilde{\xi}_x, \tilde{\xi}_y, \tilde{\xi}_z, \tilde{\xi}_\psi\right]^T$ is the control error vector; $\mathbf{K}_{\mathbf{u}} \in \Re^{4\times4}$ is a positive diagonal matrix weighing errors; and $\mathbf{J}_{\mathbf{u}}^{-1} \in \Re^{4\times4}$ is the inverse matrix of the Jacobian of the aerial robot.

4.5 Formation Controller Stability Analysis

For the stability analysis of the formation controller a perfect velocity tracking is considered, *i.e.*, $\dot{\mathbf{h}}(t) \equiv \mathbf{v}_{\mathbf{c}}(t)$. Therefore, equating equations (6) and (7) we obtain the closed-loop equation:

$$\dot{\tilde{\mathbf{h}}}(t) + \mathbf{K}\tilde{\mathbf{h}}(t) = \mathbf{0} \tag{10}$$

Now, let us consider a Lyapunov $V\left(\tilde{h}\right) = \frac{1}{2}\tilde{h}^{\mathrm{T}}\tilde{h}$ candidate. Then, deriving $V\left(\tilde{h}\right)$ with respect to time and replacing (11) results:

$$\dot{V}\left(\tilde{h}\right) = -\tilde{\mathbf{h}}^T\mathbf{K}\tilde{\mathbf{h}} \tag{11}$$

The sufficient condition to guarantee stability is that $\dot{V}\left(\tilde{h}\right) < 0$. Which is true since $\mathbf{K} \in \Re^{6\times6}$ is a positive diagonal matrix, then, $-\tilde{\mathbf{h}}^T\mathbf{K}\tilde{\mathbf{h}} < 0$. And therefore, $\lim_{t\to\infty}\tilde{\mathbf{h}}(t) \to \mathbf{0}$ then the control errors of the formation are asymptotically stable.

The stability analysis for the robot controllers is similar to the previous analysis, since the control law is equal and the gains of each controller $\mathbf{K}_{\mathbf{o}} \in \Re^{3\times3}$ and $\mathbf{K}_{\mathbf{u}} \in \Re^{4\times4}$ are positive definite matrices. Then, also the control errors of each robot are asymptotically stable.

5 Virtualization

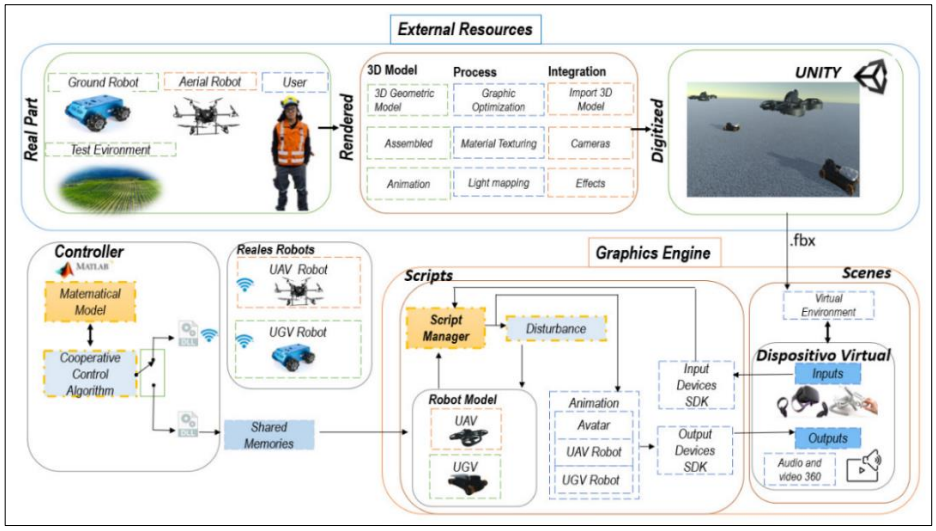

Fig. 7. Virtualization scheme.

For the virtualization of the terrestrial and aerial robot, digital models of a robot with omnidirectional traction and an unmanned aerial robot with four rotating propellers are used. For the environments where the cooperative tasks are executed with the robots, outdoor field and urban environments are taken into account. It is important to mention that it is necessary to generate real life scenes where the same conditions are considered and that guarantees the similar behavior to the real one of the robots in the environment. This virtual environment has the devices, resources and essential elements for the immersion and interaction with the audio and video of the environment. The stages implemented for this virtualization are described as follows:

1. External Resources: These resources comprise four elements that are part of the virtual environment: the scenery, the terrestrial and aerial robot, including the avatar that represents the user immersed in the virtual environment. To carry out the modeling and 3D digitalization of these resources. CAD design tools are used such as: Solidwork, Autocad 3D, Blender 3D, among others. Subsequently, using Auto-Desk 3D, 3DS Max or Rinho 7 software, we export the 3D design as. fbx extension. This format allows us to integrate the models and incorporate animations using Unity 3D software.

2. Graphics engine: The Unity3D graphics engine is used, for which there are two basic elements when developing a virtual environment: one is the Virtual Scenario, which includes all the external resources digitized in. fbx extension, audios among other elements that allow the user to feel as if they were in a real environment, and the second element is the programming scripts or programming code that allow through mathematical models to emulate the real behavior of the UGV and the UAV.

3. Control Scheme: It allows to implement the advanced control algorithms for driving the UAV and UGV autonomously, in order to execute cooperative tasks in partially structured environments. For this case study, a multilayer scheme is considered (see Figure 4) that allows the implementation and validation of different control strategies in the virtual environment.

4. Human Operator: Through a virtual interface it is possible to change the references of the different controller parameters within the virtual simulation. It also allows adding perturbations to determine the behavior of the UGV and the UAV. It is also possible to perform wireless communication to transfer the reference velocities from the MATLAB software to the real robots.

6 Analysis and Results

Figure 8.a presents the UAV considered in this work, together with its virtualization for the Unity3D environment. Figure 8.b presents the real and digitized UGV for the Unity environment. Each digitized robot has the motion characteristics that allow emulating the behavior of each of them within the virtual environment.

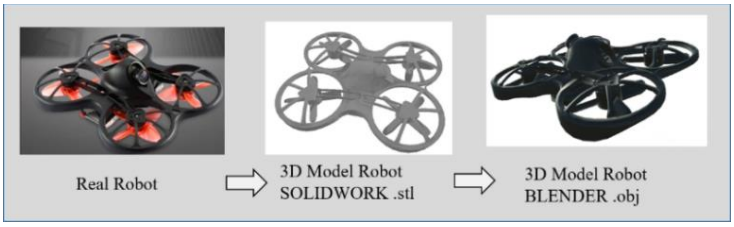

Fig 8.a. Quadcopter robot digitization.

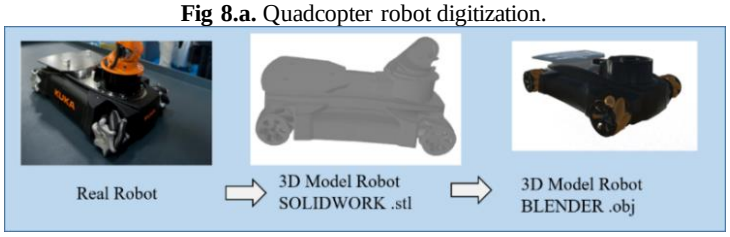

Fig. 8.b. Omnidirectional robot digitalization.

Fig. 8. UAV and UGV Real Robots digitized for Unity 3D environment**.**

In order to validate the operation of the proposed scheme, we proceed to perform an experiment which consists of executing a desired task simulating within the virtual reality environment developed in Unity. The experiment consists of following a desired trajectory (Virtual Formation) defined by the user. Figure 9 shows the stroboscopic movement captured inside the virtual reality environment, the movement executed between the UAV and the UGV are shown.

Fig. 9.a Top view of motion executed by robots in the Unity3D environment.

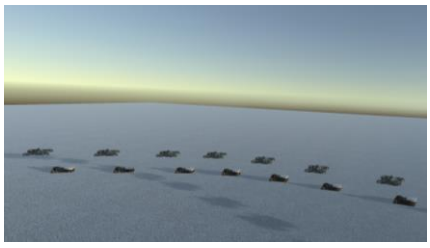

Fig. 9.b. Stroboscopic movement of robots in the Unity3D environment on XYZ space.

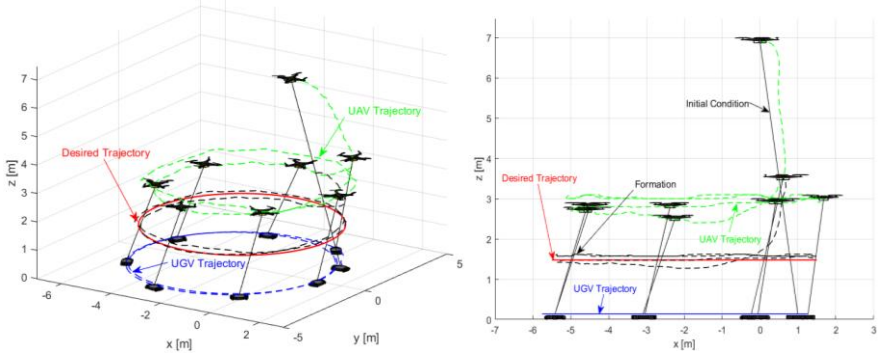

Fig. 9.c. Stroboscopic motion executed by the robots in the XZY space and XZ plane with experimental test.

Fig. 9. Heterogeneous formation in Unity3D environment.

An experiment is performed to validate the effectiveness of the proposed controller, this involves performing a specific tracking task in a physical environment. The experiment consists of performing different tests in a real space, the robot formation has to follow a specific route defined by the user. Figure 10 shows the movement performed simultaneously by the unmanned aerial vehicle (UAV) and the unmanned ground vehicle (UGV) during the real tests of the cooperative control between the two robots.

Fig. 10.a Top view of movement executed by the robots in the real environment.

Fig. 10.b. Movement of the robots in the real environment seen in XYZ space.

Fig. 10. Heterogeneous training between terrestrial and aerial robot in the real environment.

On the other hand, Figure 11 shows the control errors of the desired formation. It can be seen how the error $\tilde{\mathbf{B}}_{\mathbf{p}}(t)=\begin{bmatrix}\tilde{h}_x(t) & \tilde{h}_y(t) & \tilde{h}_z(t)\end{bmatrix}^T$ as time progresses approaches close to zero, *i.e.*, the centroid of the virtual training follows the desired trajectory. Similarly, it can be seen in Figure 12 how the errors $\tilde{\mathbf{A}}_{\mathbf{F}}(t)=\begin{bmatrix}\tilde{d}(t) & \tilde{\alpha}(t) & \tilde{\beta}(t)\end{bmatrix}^T$ are close zero. In addition, it can be seen how the error of the measured distance between the aerial robot and the ground robot tends to close to zero, in other words, during the experiment the distance between the two robots is maintained at the desired distance. These two figures show how the centralized control works to accomplish the desired virtual training.

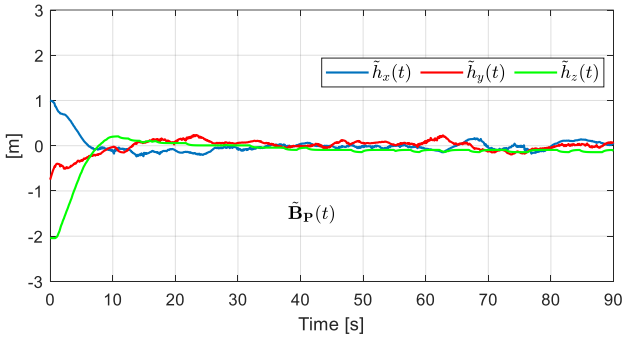

Fig. 11. Errors of the formation control during the experimental test $\tilde{\mathbf{B}}_{\mathbf{p}}(t)$.

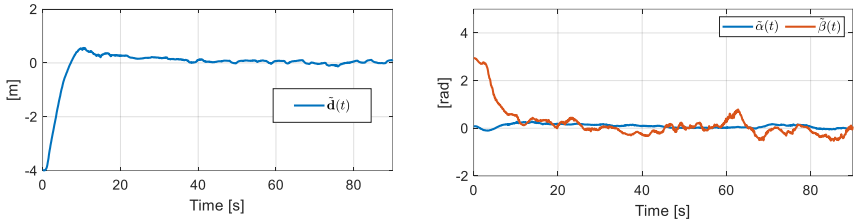

Fig. 12. Errors of the formation control during the experimental test $\tilde{\mathbf{A}}_{\mathbf{F}}(t)$.

Figure 13 shows the control errors of the UAV, it can be observed how the error $\tilde{\xi}(t)$ is close to zero as the simulation time increases, accomplishing the desired references generated by the centralized control. Likewise, it can be seen how the robot complies with the desired orientation, in this experiment it was considered that the desired orientation is tangent to the trajectory generated by the centralized control.

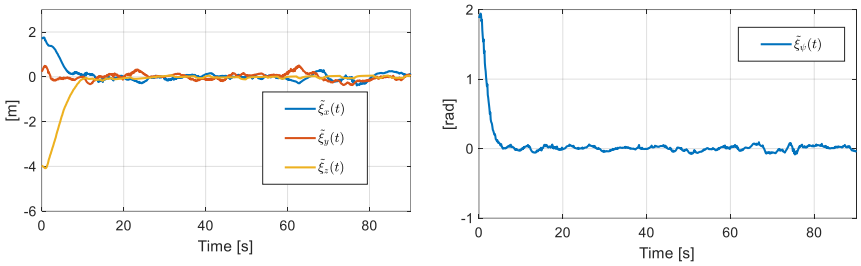

Fig. 13. UAV control errors $\tilde{\boldsymbol{\xi}}(t)$.

Finally, Figure 14 shows the behavior of the UGV control errors $\tilde{\boldsymbol{\eta}}(t)$. It is evident how the position errors $\tilde{\eta}_x(t)$ and $\tilde{\eta}_y(t)$ are close to zero, therefore, the robot follows the desired references of the decentralized control to maintain the omnidirectional platform on the XY plane. The desired orientation of the robot is defined as tangent to the trajectory generated by the centralized control, and it can be seen that the orientation error $\tilde{\eta}_\psi(t)$ is close to zero, achieving the control objective.

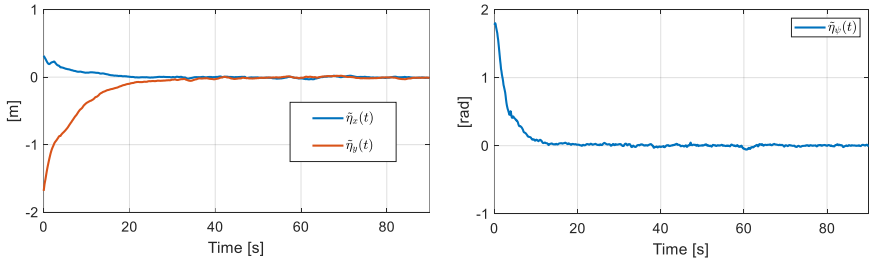

Fig. 14. UGV control errors $\tilde{\boldsymbol{\eta}}(t)$.

7 Conclusions

The development of a virtual environment in Unity3D for the simulation of control schemes is of great help, since it allows to validate and verify the operation and performance of autonomous controls for different robots. In addition, as it is an environment that allows communication with MATLAB, it is possible to design and implement several advanced control strategies for robots, allowing the user to understand and perform in the virtual environment in a versatile way to carry out different controller tests. Through the results obtained from the simulation in the virtual environment and the experiments with the robots, the fulfillment of the following objectives is verified: i) the first is to keep the robots in virtual training throughout the trajectory. This virtual formation can vary depending on time, which allows execute a variable formation task.; and ii) the second objective that is validated is decentralized control, which controls

each robot to maintain the formation. In other words, it is possible to verify that the formation control errors and the control errors of each robot are close to zero, which verifies that the implemented controllers are stable. The main advantage of this scheme is that one or more robots can be incorporated into the formation scheme, allowing to obtain a formation (cooperative) control with multiple robots, and include other types of robots for later work, both in aerial and terrestrial cooperative tasks.

Acknowledgements. The authors would like to thank the Universidad de las Fuerzas Armadas ESPE for their contribution to innovation, especially in the research project "Advanced Control of Unmanned Aerial Vehicles", as well as the ARSI Research Group for their support in developing this work.

8 References

[1] G. Piumatti, F. Lamberti, A. Sanna, y P. Montuschi, «Robust Robot Tracking for Next-Generation Collaborative Robotics-Based Gaming Environments», *IEEE Trans. Emerg. Topics Comput.*, vol. 8, n.º 3, pp. 869-882, jul. 2020, doi: 10.1109/TETC.2017.2769705.

[2] A. Ayub *et al.*, «Robot Curiosity in Human-Robot Interaction (RCHRI)», en *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, mar. 2022, pp. 1231-1234. doi: 10.1109/HRI53351.2022.9889478.

[3] H. Niu, Z. Ji, Z. Zhu, H. Yin, y J. Carrasco, «3D Vision-guided Pick-and-Place Using Kuka LBR iiwa Robot», en *2021 IEEE/SICE International Symposium on System Integration (SII)*, ene. 2021, pp. 592-593. doi: 10.1109/IEEECONF49454.2021.9382674.

[4] V. P. Bacheti, A. Santos Brandão, y M. Sarcinelli-Filho, «Path-Following by a UAV-UGV Formation Using Null Space-Based Control», en *2021 XIX Workshop on Information Processing and Control (RPIC)*, nov. 2021, pp. 1-6. doi: 10.1109/RPIC53795.2021.9648417.

[5] C. Liu, G. Wang, Y. Yan, C. Bao, Y. Sun, y D. Fan, «Pint-sized military UAV engine's tele-adjusting arm based on force feedback», en *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, mar. 2010, pp. 205-209. doi: 10.1109/CAR.2010.5456566.

[6] G. Wang, C. Wang, Q. Du, L. Li, y W. Dong, «Distributed Cooperative Control of Multiple Nonholonomic Mobile Robots», *J Intell Robot Syst*, vol. 83, n.º 3, pp. 525-541, sep. 2016, doi: 10.1007/s10846-015-0316-x.

[7] W. Yao, Z. Zeng, X. Wang, H. Lu, y Z. Zheng, «Distributed encirclement control with arbitrary spacing for multiple anonymous mobile robots», en *2017 36th Chinese Control Conference (CCC)*, jul. 2017, pp. 8800-8805. doi: 10.23919/ChiCC.2017.8028755.

[8] V. Andaluz, M. Molina, Y. Erazo, y J. Ortiz, «Numerical Methods for Cooperative Control of Double Mobile Manipulators», ago. 2017, pp. 889-898. doi: 10.1007/978-3-319-65292-4_77.

[9] V. Andaluz, C. Carvajal, J. Pérez, y L. Proaño, «Kinematic Nonlinear Control of Aerial Mobile Manipulators», ago. 2017, pp. 740-749. doi: 10.1007/978-3-319-65298-6_66.

[10] G. P. Moreno, N. D. De La Cruz, J. S. Ortiz, y V. H. Andaluz, «Human-Robot Collaborative Control for Handling and Transfer Objects», en *Applied Technologies*, M. Botto-Tobar, S. Montes León, O. Camacho, D. Chávez, P. Torres-Carrión, y M. Zambrano Vizuete, Eds., en Communications in Computer and Information Science, vol. 1388. Cham: Springer International Publishing, 2021, pp. 96-110. doi: 10.1007/978-3-030-71503-8_8.

[11] J. Roldán Gómez *et al.*, «Multi-robot Systems, Virtual Reality and ROS: Developing a New Generation of Operator Interfaces», en *Studies in Computational Intelligence*, 2019, pp. 29-64. doi: 10.1007/978-3-319-91590-6_2.

[12] M. Wu, S.-L. Dai, y C. Yang, «Mixed Reality Enhanced User Interactive Path Planning for Omnidirectional Mobile Robot», *Applied Sciences*, vol. 10, n.º 3, Art. n.º 3, ene. 2020, doi: 10.3390/app10031135.

[13] S. G. M. Gutiérrez, A. C. Pazarán, R. G. Mejía, y L. N. O. Moreno, «Desarrollo de plataforma para implementación de robots colaborativos», *Visión electrónica*, vol. 12, n.º 1, Art. n.º 1, may 2018, doi: 10.14483/22484728.13308.

[14] R. J. Ruiz, J. L. Saravia, V. H. Andaluz, y J. S. Sánchez, «Virtual Training System for Unmanned Aerial Vehicle Control Teaching–Learning Processes», *Electronics*, vol. 11, n.º 16, Art. n.º 16, ene. 2022, doi: 10.3390/electronics11162613.

[15] J. S. Ipiales, E. J. Araque, V. H. Andaluz, y C. A. Naranjo, «Virtual Training System for the Teaching-Learning Process in the Area of Industrial Robotics», *Electronics*, vol. 12, n.º 4, Art. n.º 4, ene. 2023, doi: 10.3390/electronics12040974.