Sliding Mode Control for UAV Trajectory Tracking: Focus on Virtual Learning Environments

Vallejo Morales, Roberto Carlos

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica e Instrumentación

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

Mgs. Ortiz Moreano, Jessica Sofía

22 de enero del 2024

Latacunga

Sliding Mode Control for UAV Trajectory Tracking: Focus on Virtual Learning Environments

Roberto Vallejo and Jessica S. Ortiz

Universidad de las Fuerzas Armadas ESPE, Sangolqui-Ecuador.
{rcvallejo1, jsortiz4}@espe.edu.ec.

Abstract. The present work presents a cascade control scheme, in the initial stage, there is a kinematic controller based on the sliding mode technique for the trajectory tracking of a UAV. Secondly, a dynamic controller is cascaded, this controller allows to compensate the perturbation and velocity errors of the robot. For this purpose, the simplified dynamic model of the UAV is used. For testing purposes, a virtual reality simulator is presented to test the proposed control scheme at laboratory level. The virtual reality environment will allow evaluating the UAV performance as much as possible. The virtual reality environment has the digitized physical robot, with its dynamic model and all the movement characteristics. The controller is simulated and then experimental tests are performe with the physical UAV in a partially structured environment and with wind disturbances at the time of the experiment, an $RMSE = 0.32$ [m] is obtained for the UAV position.

Keywords: sliding mode, trajectory tracking, virtual reality, dynamic model.

1 Introduction

In the last decades, robotics has grown significantly, presenting a wide variety of service robots that can perform various tasks in multiple environments: industrial [1], healthcare [2], public services [3], military [4], among other fields. These service robots can be autonomous or controlled by humans and can come in various shapes and forms, from robots with legs to robots with wheels or propellers. Robots today are used to solve dangerous tasks, thus preventing the risk of occupational accidents and increasing productivity. One of these types of robots are called unmanned aerial vehicle (UAV) or more frequently referred to as drones [5]. UAVs or drones are robots that have the ability to move from one place to another by their propellers. They are used in various applications, such as: for transporting objects over long distances, inspection of structures [6], agriculture [7], aerial photography [8], search operations [9], among other applications.

For a UAV to operate autonomously and effectively, it is essential to have a control system that allows complex tasks to be performed autonomously and safely within the working environment. Most autonomous controls use the mathematical model of the robot. Having a mathematical model of these robots is crucial to ensure the safety and efficiency of the drones in their specific task. There are controllers that use only the kinematic model. In [10] presents the autonomous control of a UAV using inverse kinematics. This control allows to accomplish three tasks, position control, trajectory

tracking and path following. The inverse kinematics control is normally used when working with tasks that do not require much precision and especially at low velocity [11]. However, for tasks that require high velocities, there are works in the literature that present controllers based on the dynamic model of the UAV [12], for example, in [13] they use the dynamic model to perform the trajectory tracking control of a UAV, the authors validate the proposal through simulation. However, in order to have higher accuracy and compensate for velocity errors or model errors, there are cascading methods for kinematic and dynamic control [14], the authors perform experimental tests with a quadrotor [15].

Most of the controller tests for its implementation require to be tested at laboratory level, *i.e.*, simulate the control scheme to adjust the gains and controller parameters, this in order to avoid damage to the physical robots, there are several robotic simulators, such as: Webots [16]; Gazebo [17]; Flightmare [18] and among others involving mobile and aerial robots. However, having the physical robot in a simulation environment is usually complicated, since most simulators only have commercial robots. In function to this, in this article a sliding mode controller is proposed for the trajectory tracking of a UAV. There are several applications of sliding control, e.g., in [19] they present a study of sliding mode control strategies for wind energy conversion systems; in [20] they present a sliding mode control strategy that guarantees that the spacecraft station (Helianthus) maintains feasible attitude maneuvers; and among other applications. As can be noted, this control strategy is used thanks to its robustness and because it works with nonlinear systems. Finally, a cascade control based on robot dynamics is proposed., which allows to compensate the errors of velocities that are presented in the experimentation.

The proposed controller is first validated in a virtual simulation environment, which has environmental characteristics similar to those of a real environment and has the molding of the robot to be used for real tests. The virtual reality environment is designed in Unity and at this stage the controller is evaluated and adjusted. Then, we proceed to test the controller with the physical robot in a partially structured environment. This paper is divided into 6 sections, including the Introduction. In Section II, the conceptualization of the process is presented, while Section III presents the mathematical models of the UAV together with the proposed controller. Section IV details the development of the virtual environment for simulation. The results obtained are presented in Section V and finally, Section VI presents the conclusions of the work.

2 Process Conceptualization

Figure 1 presents the conceptualization process for the development of the controller in combination with the virtual reality simulator. The process is divided into five parts or blocks that allow structuring the control system and the virtual environment together.

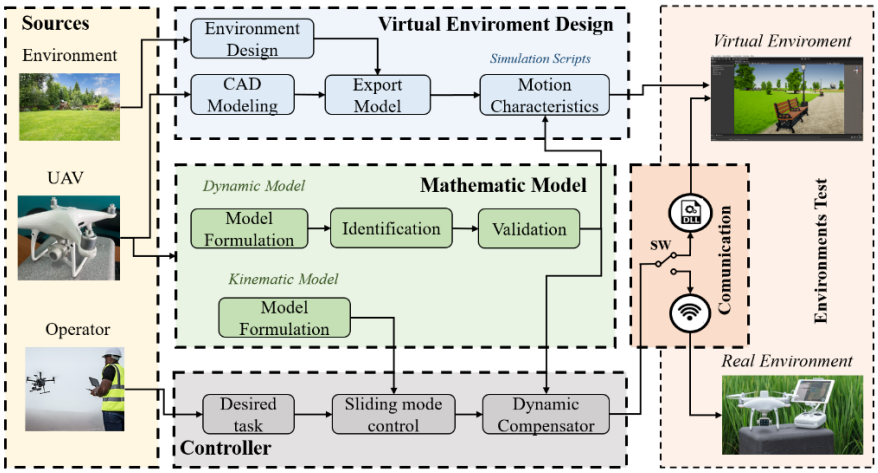


Figure. 1. Process Scheme.

i) Sources, these are the physical sources that are essential and consist of: the physical environment where the control tests will be performed, this environment is digitized considering the disturbances that may exist in reality; the UAV which is the robot in which the work will be done; and the operator who is the person in charge of task definition, *i.e.*, the desired trajectory for the robot. ***ii) Mathematic Model.*** To define the motion characteristics of the robot, the kinematic model and dynamic model of the robot are obtained. These models allow to control and define the behavior of the robot in the virtual environment. ***iii) Virtual Environment Design***. After obtaining the robot model and defining the sources to be used for the system, the virtual environment is created. For the development of the virtual environment it is essential to have the predefined structure of the environment to be used, the digitized robot and above all the mathematical model of the robot. The mathematical model of the robot allows emulating the behavior of the physical robot. ***iv) Controller***. It allows to obtain the robot maneuvering references, for the development of the controller the kinematic model and the dynamic model are used. The controller is executed in a mathematical software, in this case in Matlab software. Finally, ***v) The test environment***. This block consists of a subprocess that allows the communication of the software where the controller runs with the virtual reality environment or with the physical robot. This depends on whether you want to perform the simulations and adjustments of the controller and proceed to perform real experiments with the robot.

3 Modeling and Control

The mathematical model allows to determine the behavior of the robot according to its state variables. We have two types of model: *i) Kinematic model*, the kinematics of a robot studies the behavior of the robot without considering the forces that originate the

movement; and *ii) Dynamic model*, allows to determine the behavior of the robot according to the forces and torque acts on the robot.

3.1 Direct Kinematic

The kinematics of the aerial robot relates the movements of the robot's joints to its position and orientation in space. In other words, it is a function that, from the positions and angles of its joints, allows to determine the position and orientation of the robot in the working space [21]. In this case study, the position and rotation of the aerial robot in are defined as:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_b(t) \\ y_b(t) \\ z_b(t) \end{bmatrix} \quad \boldsymbol{\eta}(t) = \begin{bmatrix} \phi(t) \\ \theta(t) \\ \psi(t) \end{bmatrix} \tag{1}$$

where, $\boldsymbol{x}(t) \in \mathbb{R}^3$ is the position vector of the robot; $\boldsymbol{\eta}(t) \in \mathbb{R}^3$ is the rotation vector of the robot formed by: Roll $\phi(t)$ which is the rotation that occurs around the X axis $R(x, \phi)$; Pitch (θ) is the rotation that is generated around the Y axis $R(y, \theta)$; and Yaw $\psi(t)$ which is the rotation around the Z axis. Figure 2 represents the kinematic configuration of an aerial robot in the workspace.

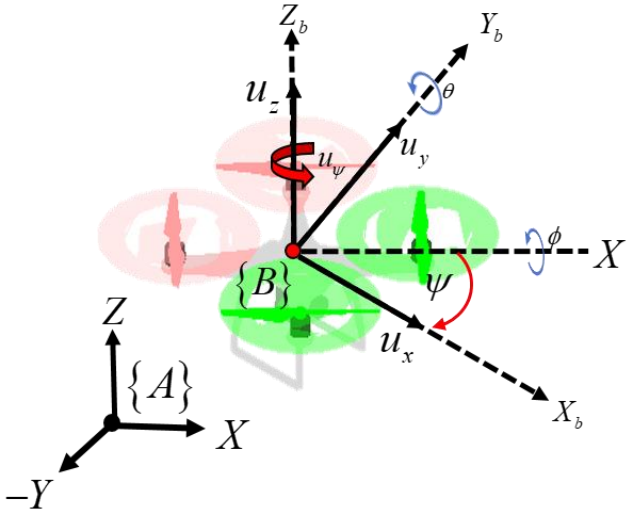


Figure. 2. UAV kinematic configuration.

The matrix relating the moving reference frame of the aerial robot $\{B\}$ to the fixed reference frame $\{A\}$ is determined by multiplying the rotation matrices $\boldsymbol{R}(\phi, \theta, \psi) = \boldsymbol{R}(x, \phi)\boldsymbol{R}(y, \theta)\boldsymbol{R}(z, \psi)$.

3.2 Differential Kinematic

In order to obtain the motion model of the aerial robot, the differential kinematic model is established. It allows to relate the robot maneuverability velocities with and its velocity and orientation in the workspace $\{B\}$.

Remark: Remark: for the purpose of the present work, which is the trajectory tracking, it is considered that the robot works at small velocities, then, the angular rotations of roll and pitch are very close to zero and the matrices $R(x,\phi) = R(y,\theta) = \boldsymbol{I}$ result in identity matrices. Therefore, the relationship between $\{B\}$ and $\{A\}$ depends only on the Yaw orientation.

The aerial robot used in this work has four maneuverability velocities defined as $\boldsymbol{u}(t) = [u_x \quad u_y \quad u_z \quad u_\psi]^T$. They allow direct control of the kinematic movement of the robot. Three linear velocity u_x, u_y and u_z that allow to move the robot frontally, laterally and elevation respectively and an angular velocity that allows to control the Yaw rotation of the robot. As the relationship between the velocities of the moving frame $\{B\}$ and $\{A\}$ depends only $\boldsymbol{R}(z,\psi)$ and the rotational velocity depends only on the angular velocity Yaw, the following equations are obtained:

$$\begin{bmatrix}\dot{x}_b(t)\\ y_b(t)\\ \dot{z}_b(t)\end{bmatrix} = \begin{bmatrix}c_\psi & -s_\psi & 0\\ s_\psi & c_\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u_x(t)\\ u_y(t)\\ u_z(t)\end{bmatrix} \quad \begin{bmatrix}\dot{\phi}(t)\\ \dot{\theta}(t)\\ \dot{\psi}(t)\end{bmatrix} = \begin{bmatrix}0\\ 0\\ u_\psi(t)\end{bmatrix} \tag{2}$$

Now, writing in a compact form the kinematic model that relates all the maneuverability velocities of the drone with the velocities in the workspace is obtained:

$$\dot{\boldsymbol{\zeta}}(t) = \boldsymbol{\Gamma}(\psi)\boldsymbol{u}(t) \tag{3}$$

where, $\dot{\boldsymbol{\zeta}}(t) = [\dot{\boldsymbol{x}}^T \quad \dot{\psi}]^T \in \mathbb{R}^4$ is the vector containing the robot velocities in the fixed reference frame $\{A\}$; $\boldsymbol{u}(t) = [u_x \quad u_y \quad u_z \quad u_\psi]^T$ is the vector of robot maneuverability velocities within the moving frame $\{B\}$; and $\boldsymbol{\Gamma}(\psi) \in \mathbb{R}^{4\times4}$ is the matrix relating the velocities of the moving frame and the velocities of the fixed reference frame $\{A\}$. The matrix is defined as:

$$\boldsymbol{\Gamma}(\psi) = \begin{bmatrix}\boldsymbol{R}(z,\psi) & \boldsymbol{0}_{3\times1}\\ \boldsymbol{0}_{1\times3} & 1\end{bmatrix} \tag{4}$$

3.3 Dynamic Model

The structure of the dynamic model used in this work is similar to that of the AR. Drone presented in [22]. Simplified dynamics is taken into account, accounting for the dominant dynamics. Therefore, the UAV used in this work (DJI phantom) has the following dynamic model structure:

$$\boldsymbol{u_{ref}}(t) = \boldsymbol{D}\dot{\boldsymbol{u}}(t) + \boldsymbol{F}\boldsymbol{u}(t) \tag{5}$$

where, $\boldsymbol{u_{ref}} = [u_{x_{ref}} \quad u_{y_{ref}} \quad u_{z_{ref}} \quad u_{\psi_{ref}}]^T$ is the vector of reference velocities; $\dot{\boldsymbol{u}}(t) \in \mathbb{R}^4$ is the vector of UAV accelerations; $\boldsymbol{u}(t) \in \mathbb{R}^4$ is the vector of UAV velocities; $\boldsymbol{D}, \boldsymbol{F} \in \mathbb{R}^{4\times4}$ are diagonal matrices obtained through the identification and validation of the UAV phantom 4, described as:

$$\boldsymbol{D} = diag([0.6599 \quad 0.6725 \quad 0.4642 \quad 0.2756])$$
$$\boldsymbol{F} = diag([0.5978 \quad 0.6628 \quad 1.1145 \quad 0.9398]) \tag{6}$$

3.4 Controller

The scheme of the controller can be seen in Figure 3, in the first stage there is the kinematic sliding mode control in cascade is the dynamic control to compensate for velocity errors.

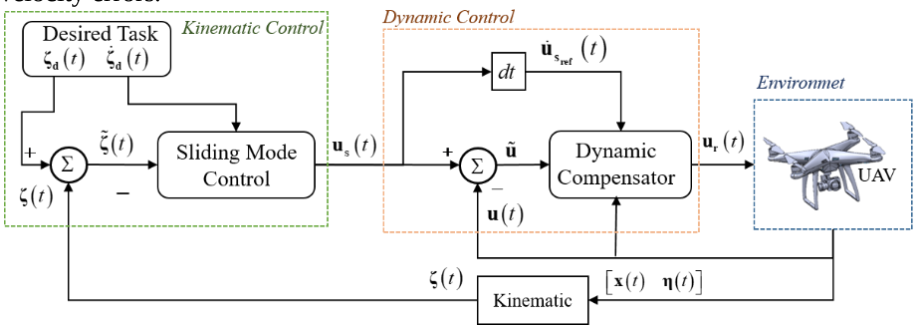


Figure. 3. UAV controller schematic.

Sliding Mode Controller

For the design of the controller, the theory about this type of controller is considered [10]. Now, used for trajectory tracking it consists of obtaining control actions to drive the robot along a desired trajectory. It consists of two control actions:

$$\boldsymbol{u_s}(t) = \boldsymbol{v_C}(t) + \boldsymbol{v_D}(t) \tag{7}$$

where, $\boldsymbol{u_s}(t) \in \mathbb{R}^4$ is the vector of robot maneuverability actions; $\boldsymbol{v_C}(t) \in \mathbb{R}^4$ is the continuous control action, which allows the system to stay on the sliding surface; $\boldsymbol{v_D}(t) \in \mathbb{R}^4$ and represents the discontinuous part of the controller. The discontinuous action allows the system to search for the sliding surface. Now, we propose the following sliding Surface:

$$s = \tilde{\boldsymbol{\zeta}}(t) + \boldsymbol{K_s} \int \tilde{\boldsymbol{\zeta}}(t)\, dt \tag{8}$$

where $\tilde{\boldsymbol{\zeta}}(t) = \boldsymbol{\zeta_d}(t) - \boldsymbol{\zeta}(t)$ is the vector of trajectory tracking errors; $\boldsymbol{K_s} \in \mathbb{R}^{4\times4}$ and λ is a design constant; $\boldsymbol{K_s} \in \mathbb{R}^{4\times4}$ is the gain matrix. Now, taking the partial derivative of (8) we obtain:

$$\dot{s} = \dot{\tilde{\boldsymbol{\zeta}}}(t) + \boldsymbol{K_s}\tilde{\boldsymbol{\zeta}}(t) \tag{9}$$

Taking the derivative of the trajectory tracking error and considering that the continuous part $\boldsymbol{v_C}(t)$ is on the sliding surface, then $\dot{s} = 0$ and the equation (9) results in $\boldsymbol{0} = \dot{\boldsymbol{\zeta}}_{\boldsymbol{d}} - \dot{\boldsymbol{\zeta}} + \boldsymbol{K_s} \in \mathbb{R}^4\tilde{\boldsymbol{\zeta}}$. Now, replacing the kinematic model (3) we obtain and as the robot is on the sliding surface, therefore, $\boldsymbol{u} = \boldsymbol{v_C}$. Then it turns out that the continuous control action is:

$$\boldsymbol{v_C}(t) = \boldsymbol{\Gamma}(\psi)^{-1}\left(\dot{\boldsymbol{\zeta}}_{\boldsymbol{d}}(t) + \boldsymbol{K_s}\tilde{\boldsymbol{\zeta}}(t)\right) \tag{10}$$

Now, for the discontinuous part $\boldsymbol{v_D}$ is done by means of the Lyapunov analysis theory, a Lyapunov candidate is proposed $V = ss^T$ and its derivative results in $\dot{V} = s^T\dot{s}$. Substituting $\dot{s}$ in $\dot{V}$ results $\dot{V} = s^T\left(\dot{\boldsymbol{\zeta}}_{\boldsymbol{d}} - \boldsymbol{\Gamma}(\psi)\boldsymbol{u} + \boldsymbol{K_s}\tilde{\boldsymbol{\zeta}}\right)$. Then, taking the control actions defined in (7) we make $\boldsymbol{u_s} = \boldsymbol{u}$ resulting in:

$$\dot{V} = s^T\left(\dot{\boldsymbol{\zeta}}_{\boldsymbol{d}} - \boldsymbol{\Gamma}(\psi)\left(\boldsymbol{\Gamma}(\psi)^{-1}\left(\dot{\boldsymbol{\zeta}}_{\boldsymbol{d}} + \boldsymbol{K_s}\tilde{\boldsymbol{\zeta}}\right) + \boldsymbol{v_D}\right) + \boldsymbol{K_s}\tilde{\boldsymbol{\zeta}}\right) \tag{11}$$

Simplifying (11) we obtain $\dot{V} = s^T(-\boldsymbol{\Gamma}(\psi)\boldsymbol{v_D})$, so that $\dot{V} < 0$, the control action $\boldsymbol{v_D}$ is defined as follows:

$$\boldsymbol{v_D}(t) = \boldsymbol{\Gamma}(\psi)^{-1}\boldsymbol{K_D}f(s) \tag{12}$$

where, $\boldsymbol{K_D} \in \mathbb{R}^4$ is a diagonal matrix defined positive; $f(s) = \frac{s}{|s|+\sigma}$ it is a function that allows maintaining the stability of the controller with . Then substituting (12) in $\dot{V}$ we obtain $\dot{V} = s^T(-\boldsymbol{K_D}f(s))$. Then, if $\mathbf{K}_\mathbf{s}$ is positive definite of $\boldsymbol{v_C}$, it means that $s(t) \to 0$, therefore, the system reaches the sliding surface and that the control errors $\tilde{\boldsymbol{\zeta}} \to \mathbf{0}$ have to be zero. And with the odd function $f(s)$ we reduce the oscillations in the compensation asymptotically to the value $s(t) = 0$.
In other words, $\dot{V} < 0$ it is negative definite and the control error is asymptotically stable. The sliding mode control law is defined as:

$$\boldsymbol{u_s}(t) = \boldsymbol{\Gamma}(\psi)^{-1}\left(\dot{\boldsymbol{\zeta}}_{\boldsymbol{d}}(t) + \boldsymbol{K_s}\tilde{\boldsymbol{\zeta}}(t) + \boldsymbol{K_D}f(s)\right) \tag{13}$$

Dynamic Controller

Dynamic control allows us to correct the velocity errors $\tilde{\boldsymbol{u}}(t) = \boldsymbol{u_s}(t) - \boldsymbol{u}(t)$. Then we need a control action $\boldsymbol{u_r}(t)$ that allows us to correct these errors. Therefore, we propose the controller of the form:

$$\boldsymbol{u_r}(t) = \boldsymbol{D}\boldsymbol{a}(t) + \boldsymbol{F}\boldsymbol{u}(t) \tag{14}$$

where, $\boldsymbol{a}(t) = \dot{\boldsymbol{u}}_{\boldsymbol{s_{ref}}}(t) + \boldsymbol{K_r}\tilde{\boldsymbol{u}}(t)$ is the control action that allows to correct the velocity errors; $\dot{\boldsymbol{u}}_{\boldsymbol{s_{ref}}}(t)$ is the desired acceleration; and $\boldsymbol{u}(t)$ is the actual velocity of the robot.

4 Virtual Environment Design

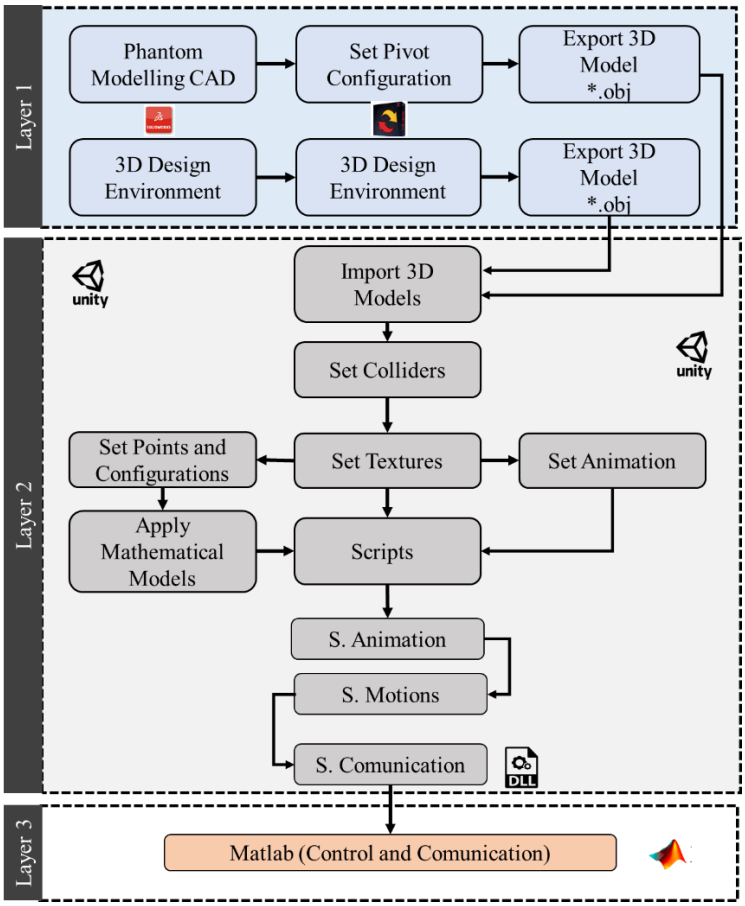


Figure. 4. Multilayer scheme of the simulator design in Unity

The multilayer scheme in Figure 4 shows the steps for the development of the simulator, consisting of three stages: ***i) Layer 1***. In the first stage, the 3D models are developed, the UAV model is developed in SolidWorks software, which is an appropriate software for the CAD model of prototypes. And as a second part the model of the environment, this model is developed in Sketchup for the creation of the buildings. These models are exported in a format admissible for Unity (*.obj). ***ii) Layer 2.*** In the second stage there is the programming of the simulator in virtual reality. This is done in the UNITY software. First the model of the robot and the environment is exported. The animation characteristics of each model are placed. The scrpits for the animations are programmed, especially the one for the robot animation. The communication script is made to link Matlab with Unity, the link consists of a DLL (Dynamic Link Library). And finally ***iii) Layer 3.*** In this stage the controller developed in Matlab is linked with

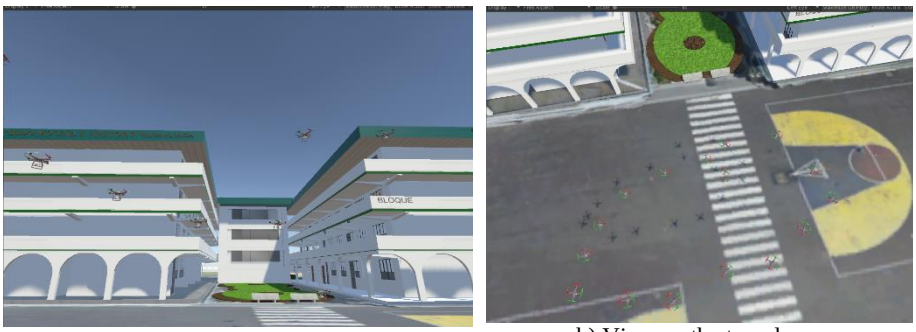
Unity, the DLL is created to send the maneuverability commands of the UAV to the robot emulated in Unity. Figure 5 shows the developed environment and the UAV within the environment and the UAV export to Unity.

a) Unity environment. b) UAV in the virtual environment.

Figure. 5. Environment in Unity with Pahntom 4 drone.

5 Results and Discussions

To validate the controller, the simulation result of the controller simulation is presented to adjust the controller. Figure 6 shows the movement executed by the UAV in the virtual environment. As it is a system that first allows to emulate the behavior of the robot, the values of the control gains in sliding mode and the control gains with dynamics are adjusted.



a) XZ plane view. b) View on the top plane.

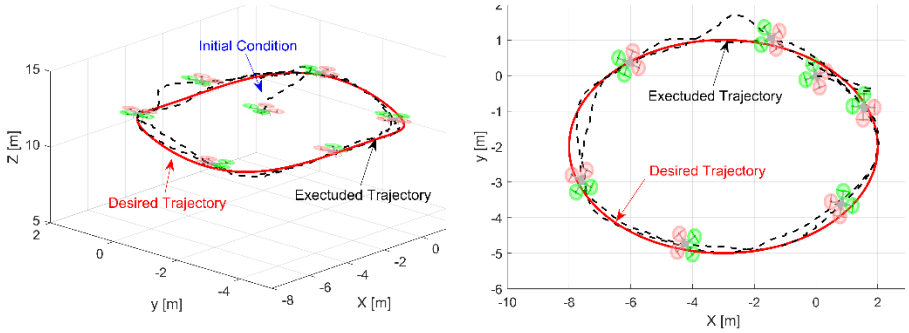
c) Perspective view.

Figure. 6. Movement of the robot in the virtual reality simulator.

Figure 7 presents the movement with the DJI Phantom 4 aerial robot during the experimental test.

Figure. 7. Stroboscopic motion executed during the experiment.

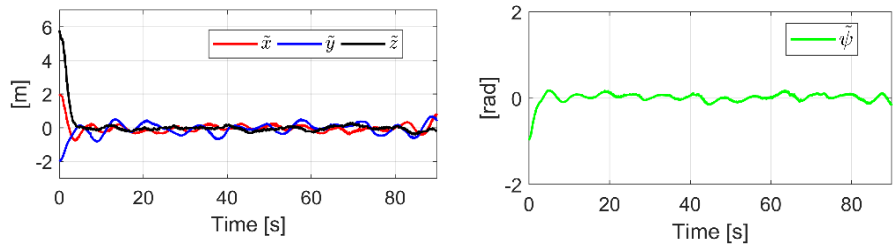
After the controller has been adjusted, we proceed to test the controller experimentally. The experiment consists of defining a trajectory and executing the cascade control with the dynamic compensator to observe the behavior of the UAV in a real environment. Figure 8 shows the actual movement of the UAV.



a) Movement of the robot in XYZ space. b) Robot movement in the XY plane.

Figure. 8. Executed movement of the UAV during the experiment.

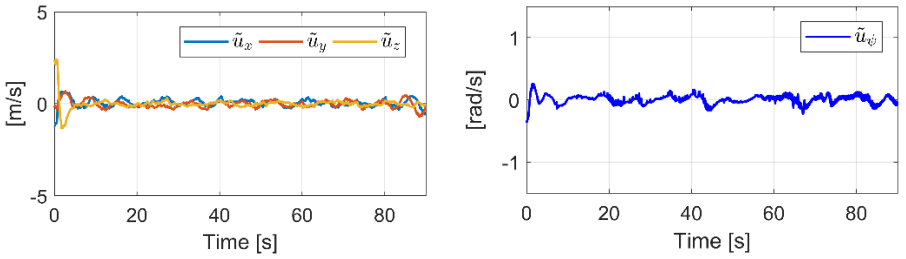
Figure 9 shows the trajectory tracking control errors. Figure a) shows the position errors, it is evident how they are very close to zero. Figure b) shows the orientation error, the desired orientation is perpendicular to the trajectory and the error is close to zero. It is worth mentioning that at the beginning the controller corrects the error, that is, until it finds the sliding surface, once on the sliding surface it is maintained and therefore the errors are very close to zero. El error cuadrático medio para la posición del UAV es de $RMSE_p = 0.32\ [m]$ y para la orientación resulta $RMSE_\psi = 0.12\ [rad]$.



a) Position errors. b) Orientation error.

Figure. 9. UAV trajectory tracking errors.

Finally, Figure 10 shows the velocity errors, it can be seen how the linear and angular velocity errors tend to zero. This is thanks to the cascade dynamic compensator. This control allows correcting the dynamic effects of the robot. El error cuadrático medio de la velocidad lineal resulta $RMSE_u = 0.29\ \left[\frac{m}{s}\right]$ y de la velocidad de orientación es $RMSE_{u_\psi} = 0.017\ \left[\frac{rad}{s}\right]$.



a) Lineal velocity errors. b) Orientation velocity error.

Figure. 10. UAV velocity errors.

6 Conclusions

With the mathematical models of the UAV, the behavior of the robot was emulated, which allows the robot to be incorporated into a simulator developed in a virtual reality environment. Thanks to this, the controllers were adjusted by simulation and then experimental tests were carried out with the real robot without causing damage to the physical robot. This simulator can include more robots and will allow simulations to test different control schemes for UAVs. The controller proposed for the trajectory tracking behaves stably and allowed to perform the UAV trajectory tracking autonomously. The robustness of the controller can be observed when there is a disturbance induced by the environmental conditions of the experiment (wind) the controller keeps the robot within the desired trajectory. While the controller with dynamics allowed correcting the velocity errors by compensating the disturbances or non-modeled parameters of the system, the controller with dynamics allowed correcting the velocity errors by compensating the disturbances or non-modeled parameters of the system.

Acknowledgments: 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.

References

1. Helms, E., Schraft, R.D. and Hagele, M. (2002) 'rob@work: Robot assistant in industrial environments', in 11th IEEE International Workshop on Robot and Human Interactive Communication Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication Proceedings, pp. 399–404. https://doi.org/10.1109/ROMAN.2002.1045655.
2. Ali, S. et al. (2019) 'An Adaptive Multi-Robot Therapy for Improving Joint Attention and Imitation of ASD Children', IEEE Access, 7, pp. 81808–81825. https://doi.org/10.1109/ACCESS.2019.2923678.

3. Jeffares, S. (2021) ‘Robots and Virtual Agents in Frontline Public Service’, in S. Jeffares (ed.) The Virtual Public Servant: Artificial Intelligence and Frontline Work. Cham: Springer International Publishing, pp. 183–210. https://doi.org/10.1007/978-3-030-54084-5_8
4. Singer, P.W. (2009) ‘Military Robots and the Laws of War’, The New Atlantis, (23), pp. 25–45.
5. Suzuki, S. (2018) ‘Recent researches on innovative drone technologies in robotics field’, Advanced Robotics, 32(19), pp. 1008–1022. https://doi.org/10.1080/01691864.2018.1515660.
6. Hu, M. et al. (2019) ‘On the joint design of routing and scheduling for Vehicle-Assisted Multi-UAV inspection’, Future Generation Computer Systems, 94, pp. 214–223. https://doi.org/10.1016/j.future.2018.11.024.
7. Tsouros, D.C., Bibi, S. and Sarigiannidis, P.G. (2019) ‘A Review on UAV-Based Applications for Precision Agriculture’, Information, 10(11), p. 349. https://doi.org/10.3390/info10110349.
8. Barlow, J., Gilham, J. and Ibarra Cofrã, I. (2017) ‘Kinematic analysis of sea cliff stability using UAV photogrammetry’, International Journal of Remote Sensing, 38(8–10), pp. 2464–2479. https://doi.org/10.1080/01431161.2016.1275061.
9. An Autonomous Multi-UAV System for Search and Rescue | Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use (no date). https://dl.acm.org/doi/abs/10.1145/2750675.2750683 (Accessed: 14 May 2023).
10. Andaluz, V.H. et al. (2018) ‘Robot nonlinear control for Unmanned Aerial Vehicles’ multitasking’, Assembly Automation, 38(5), pp. 645–660. https://doi.org/10.1108/AA-02-2018-036
11. Lugo-Cárdenas, I., Salazar, S. and Lozano, R. (2017) ‘Lyapunov Based 3D Path Following Kinematic Controller for a Fixed Wing UAV**Research supported in part by the Mexican National Council for Science and Technology (CONACYT) under grant 218857’, IFAC-PapersOnLine, 50(1), pp. 15946–15951. https://doi.org/10.1016/j.ifacol.2017.08.1747
12. Chuang, H.-M., He, D. and Namiki, A. (2019) ‘Autonomous Target Tracking of UAV Using High-Speed Visual Feedback’, Applied Sciences, 9(21), p. 4552. https://doi.org/10.3390/app9214552.
13. Chuang, H.-M., He, D. and Namiki, A. (2019) ‘Autonomous Target Tracking of UAV Using High-Speed Visual Feedback’, Applied Sciences, 9(21), p. 4552. https://doi.org/10.3390/app9214552.
14. Ryll, M., Bülthoff, H.H. and Giordano, P.R. (2012) ‘Modeling and control of a quadrotor UAV with tilting propellers’, in 2012 IEEE International Conference on Robotics and Automation. 2012 IEEE International Conference on Robotics and Automation, pp. 4606–4613. https://doi.org/10.1109/ICRA.2012.6225129.
15. Santos, M.C.P. et al. (2019) ‘An Adaptive Dynamic Controller for Quadrotor to Perform Trajectory Tracking Tasks’, Journal of Intelligent & Robotic Systems, 93(1), pp. 5–16. https://doi.org/10.1007/s10846-018-0799-3.
16. Guyot, L. et al. (2023) ‘Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions’.
17. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo | SpringerLink (no date). https://link.springer.com/chapter/10.1007/978-3-642-34327-8_36
18. Song, Y. et al. (2021) ‘Flightmare: A Flexible Quadrotor Simulator’, in Proceedings of the 2020 Conference on Robot Learning. Conference on Robot Learning, PMLR, pp. 1147–1157. https://proceedings.mlr.press/v155/song21a.html

19. Mousavi, Y. et al. (2022) ‘Sliding mode control of wind energy conversion systems: Trends and applications’, Renewable and Sustainable Energy Reviews, 167, p. 112734. https://doi.org/10.1016/j.rser.2022.112734.
20. Bassetto, M. et al. (2022) ‘Sliding mode control for attitude maneuvers of Helianthus solar sail’, Acta Astronautica, 198, pp. 100–110. https://doi.org/10.1016/j.actaastro.2022.05.043.
21. Carvajal, C.P. et al. (2020) ‘Optimal Trajectory Tracking Control for a UAV Based on Linearized Dynamic Error’, in H. Fujita et al. (eds) Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 83–96. https://doi.org/10.1007/978-3-030-55789-8_8.
22. Santos, M.C.P. et al. (2019) ‘An Adaptive Dynamic Controller for Quadrotor to Perform Trajectory Tracking Tasks’, Journal of Intelligent & Robotic Systems, 93(1), pp. 5–16. https://doi.org/10.1007/s10846-018-0799-3.