# Autonomous and Tele-Operated Navigation of Aerial Manipulator Robots in Digitalized Virtual Environments

Christian P. Carvajal(✉), María G. Méndez(✉), Diana C. Torres(✉), Cochise Terán(✉), Oscar B. Arteaga(✉), and Víctor H. Andaluz(✉)

Universidad de Las Fuerzas Armadas ESPE, Latacunga, Ecuador
chriss2592@hotmail.com, {mgmendez,dtorres,hcteran, obarteaga,vandaluzl}@espe.edu.ec

**Abstract.** This paper presents the implementation of a 3D virtual simulator that allows the analysis of the performance of different autonomous and tele-operated control strategies through the execution of service tasks by an aerial manipulator robot. The simulation environment is development through the digitalization of a real environment by means of 3D mapping with Drones that serves as a scenario to execute the tasks with a robot designed in CAD software. For robot-environment interaction, the Unity 3D graphics engine is used, which exchanges information with MATLAB to close the control loop and allow for feedback to compensate for the error. Finally, the results of the simulation, which validate the proposed control strategies, are presented and discussed.

**Keywords:** Environment digitalization · Service robotic · Aerial manipulators

## 1 Introduction

In recent years, the robotics has taken boom in different application fields, introducing sustainable solutions to solve problems within multiple areas as military, industry, medicine, among many areas, taking advantage of the new technologies that allow to optimize the intelligence and the mobility of the robotic prototypes [1, 2]. The services that present the different types of robots depend on how they are structured mechanically, presenting a wide variety of robotic systems as they are terrestrial, aquatic, spatial, air, among others; and simultaneously each of these using shaped systems of locomotion of wheels, of paws, wings, helices, type caterpillar, between other [3, 4]. In this way, the mobile robotics incorporates applications to give support to the human being, appending robots that allow execute tasks that require greater precision and time optimization within the industrial field [5]. The vast majority of these robot's work in structured environments and partially structured environments, with the purpose of accomplishing tasks of transport, navigation, handling and translation of objects from one place to another [3, 5]. Although a vehicle can reach physical objectives, the lack of a robotic arm can be a limiting factor to run more complex tasks and precision tasks. In order to fulfill these tasks, the robotic systems can incorporate arms to their mechanical structure [6].


L. T. De Paolis and P. Bourdot (Eds.): AVR 2018, LNCS 10851, pp. 496–515, 2018.
https://doi.org/10.1007/978-3-319-95282-6_36

The physical unification between an air vehicle and a set of robotic arms is commonly known as aerial manipulator (VMA), and can be simple (a single arm), dual (two arms) or multiple arms [7]. Given the versatility presented by this type of vehicles, different tasks such as welding, transport of long objects, manipulation of elements placed in heights. The tasks can be executed given by the amount of redundancy presented by the robotic system. Works as [8] present real applications of this type of robotics sets, posing a solution for the adjustment of valves located at a distance or to heights that a human operator cannot access. Also, applications oriented to the service robotics can be developed [10]. In this aspect, [9] presents the use of a simple aerial manipulator for Perching and Door-opening tasks, showing the capabilities that can have this kind of vehicles in difficult tasks to run with a terrestrial robot. Besides, the applications that can be run with this type of robot require robust controllers that unify the mobile part with the robotic arm [11]. Focuses on the design, modeling and control of a mobile prototype manipulator, proposing an innovative configuration in order to guide this vehicle to applications indoor. Jobs as these emphasize the need to incorporate advanced control strategies to this type of robotic systems, which may vary according to the type of method that one glides to use [12].

Control strategies that are used in aerial manipulators are commonly studied from a general perspective, using the kinematics that conform the robotic system configuration [13]. Depending on the type of control required, the whole robotic can be analyzed jointly or separately (robotic arm and Air Vehicle), where the points of interest are dependent on this type of analysis. An unmanned air vehicle with two arms can be analyzed cinematically as three different robots (UAV, the right arm and left arm), as a UAV and a pair of arms, or as a single system with the point of interest in the middle of the ends operational of each arm [13, 14]. The cinematic model depends directly on this analysis, which contains the features of movement of a vehicle that can be used both for simulation and experimentation. Commonly, controllers require simulators to validate the generation of responses from the control strategy, where by facility many of the times is used the mathematical software where develops the control system. However, own graphs limitations of the Mathematical Software impede any precise assessment of the implementation, forcing them to seek better alternatives to resolve these shortcomings [15].

A simulation alternative can be implemented across platforms for the development of video games, such as Unity, Unreal, cryEnGINE, highlighting to Unity 3D for various relevant characteristics. In spite of that Unity 3D is an engine of videogames, the facilities of connection with external software and import of external information makes it an attractive solution for engineering applications [16]. Compatibility with different formats, low latency of exchange of data in real time, the versatility to interact with other software, supports integrated for video cards, physics engines, and support for devices of virtual reality. Are the characteristics that make Unity a highly, scalable and useful engine for simulation [17, 18]. In this context, various virtual environments can be designed for simulation tests, either using elements of the graphics engine or 3D models designed in another software. However, many of the times the requirements of simulation pose have similar environments to the reality to determine visually the performance of the task. In this way, the import to the simulation should not be provided only with reconstructed three-dimensional models, but also reconstructions made by photography, taking into consideration that all models require a format compatible with Unity [18].

The inclusion of rebuilt environments on a video games platform help visually to evaluate the performance of controllers, as well as to understand the functioning of imported 3D models. In this aspect, works like [19, 20] and [20] pose reconstruction methods of real environments such as buildings, plots or archaeological sites. The literature indicates a large amount of works related to reconstruction based on image, where the capture methods vary from vision by satellite to the use of manned and unmanned aerial vehicles. The addition of environments reconstructed presents advantages at the moment of knowing the real environments where the implementation is to be carried out, where features such as potentially dangerous areas, structural obstacles, fauna and flora and uneven terrain. Can be identified. In this context, this article proposes the virtualization of real environments for the autonomous navigation and tele-operated in an aerial robot, with the purpose of creating an alternative to the experimental tests combining Virtual Reality with a CAD software and a mathematical software. The simulation of all these parameters in a virtual environment close to reality allows you to interact with the 3D model of an aerial robot, which runs the trajectory tracking for the evaluation process for the different schemes of autonomous control.

This work is divided into 5 sections, including the introduction. In Sect. 2, it presents the approach the problem, while Sect. 3 presents the diagram multilayer. The kinematic model and the development of control strategies are presented in Sect. 4. The results simulated obtained are presented in Sect. 5 and finally, in Sect. 6 presents the conclusions of the work.

## 2 Problem Formulation

To execute experimental tests in purpose to evaluate control algorithms to perform different tasks with an aerial manipulator robot it is indispensable to be provided with the physical prototype; considering the high inversion not only in acquiring the robot but also to cover needs for use, maintenance and acquisition of licenses for program it, of equal way one adds the inconvenient caused by meteorological conditions of the environment where the experimental tests are executed, i.e., speed of the wind, temperature, turbulence, rain, between other meteorological conditions; affecting in a significant way the tests with the robot; To the different limitations mentioned that is includes the fact that is experimenting with the robot, i.e., the prototype will be put to multiple tests in which the it risk its physical integrity and therefore its functioning during the experimental procedure. Taking into account the problematics that are generated when performing experimental tests, is seen feasible evaluate the control strategies in a simulator, the which provides of way visually the operation of the aerial manipulator, incorporating features and configurations own the robot to correctly simulate their actual behavior; allowing functionally analyzing the stability and robustness of different advanced control strategies for autonomous and/or tele-operated navigation tasks.

At present it makes use of simulators that allow to reproduce the behavior of various aerial robots, most of them targeting for the most of them focused for the pilots' training or tests of autonomous control, i.e., simulators ranging from simple systems of training to sophisticated systems that allow to simulate the automatic control of the robot aerial; in [21–23] is performed flight tests self-nomos a UAV, presenting results of behavior in a visual manner with the help of the pretender Flight Gear that incorporates a wide variety of models of ships to select. Similarly, there are aerial vehicles that integrate their own simulator, e.g., DJI, offers a unique experience by controlling in a 3D environment The Phantom or other prototypes of the company DJI, allowing to familiarize with the control knobs and the different applications that incorporates the commercial platform, the main requirement of the simulator is to have connected the prototype physically in the computer in order to run the simulation [24, 25]. On the other hand, there are software like Stage that simulates a large variety of mobile robots, sensors and objects environmental, taking as original purposes; *(i)* Allowing the rapid development of control algorithms which eventually commanding real robots; and *(ii)* allowing experiments of robots without access to hardware and without real environments [26].

The different existing simulators have their own limitations as they are not flexible in terms of the number of robots that can be integrated into the environment and make use of not real scenarios; limitations that must be taken into account to make a more robust evaluation of the different control strategies. Therefore, it is proposed to develop a 3D simulator that requires its own characteristics consuming real ambient meteorological data, such as temperature, wind speed, weather conditions, among other data; the simulation scenario will be a digitized environment using the 3D mapping method with drones obtaining a georeferenced DTM (Digital Terrain Model), providing an environment close to reality and with GPS data of the location of the mapped terrain. At the same time, the simulator allows to implement different advanced control strategies for an aerial manipulator, consisting of a rotary wing air vehicle and two robotic arms at the bottom of the platform, the robot will be modeled in a CAD software, considering all the appropriate configurations to resemble a real aerial manipulator.

## 3 Estructure System

In Fig. 1, the proposed scheme for the simulator's implementation is presented, which is developed under the UNITY 3D platform. The system consists of Environment Simulation, programming SCRIPTS to control each game object of the system, input devices, Output devices and external sources that help the development of the virtual simulator.

The *Environment of Simulation* contains 3D models whit allows to create the virtual simulation environment. The characteristics of the environment are assigned, such as weather, gravity, audio data and other physical properties that simulate a real environment. The environment incorporates an aerial manipulator designed to perform control tests for autonomous or tele-operated navigation tasks, where the task is

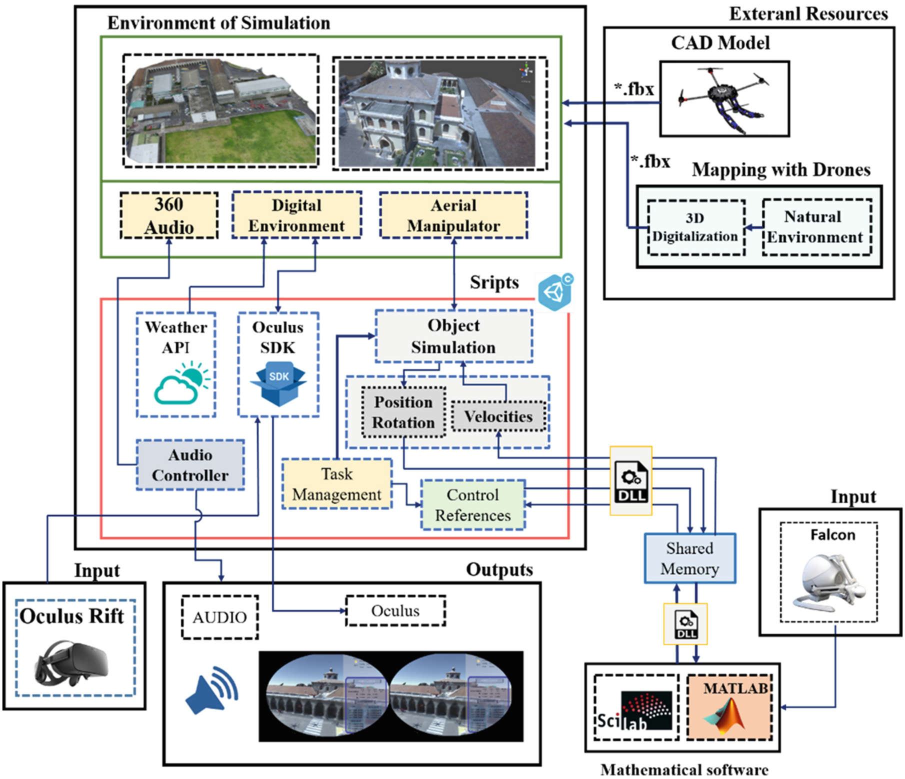



**Fig. 1.** Simulator's function scheme.

selected prior to the simulation within an interactive menu for the user. In addition, it allows to write the position and initial orientation in which the robot is placed.

The *External Resourses* are defined by 3D models: (i) Aerial manipulator developed in a CAD software with all the characteristics and appropriate configurations that a real aerial manipulator has and (ii) the 3D model of the terrain digitized by means of the 3D mapping technique with drones; achieving a simulation scenarios similar to the real and performing the control tests on it.

The set of *SCRIPTS* contains blocks of code which allow communication with input and output devices of the system. For the interaction with the virtual reality input device (OCULUS), the propietary library of the device (SDK) is used, making it possible to establish communication with the equipment in order to visually and aurally feedback the execution of the proposed task of the manipulator aerial. The climate data will be consumed from a Weather API, thus obtaining the status of the digitized terrain climate, to be incorporated in the simulation environment so that it influences the aerial manipulator control test; finally, the diagram contains SCRIPTS that allow the link with the mathematical software through the use a dynamic-link library (DLL) that generate a shared memory (SM) for the exchange of data between different software. By means of the SM the control actions are injected to the aerial manipulator and in the same way the position and rotation data of the robot is found, are read and later sent to the mathematical software, obtaining in this way the feedback of the simulation to compensate the control errors.

# 4 Virtual Simulator Environment Development

For the development of the application and its functionalities, 4 stages are defined as shown in Fig. 2. The stages of development are separated by: *(i)* 3D design of the aerial manipulator modeled from a CAD software, *(ii)* 3D digital model of the simulation environment, *(iii)* import of each 3D model and the respective programming to each game object that are port of the simulator and *(iv)* the control writing proposed for the aerial manipulator, which is done from Matlab mathematical software. All the layers are indispensable for the correct execution of the presented system. In this way, a simulator with sufficient characteristics to evaluate advanced control algorithms is developed.

## 4.1 3D Aerial Manipulator Design

The design of a computer or machine is done in a CAD software such as Solid-Works, which allows to create solid objects to be assembled allowing to obtain 3D models complete. Figure 3. presents the block diagram of the steps to obtain the final model of the aerial manipulator used for simulation in the virtual environment.

The design of the aerial manipulator is composed of an airborne platform and two robotic arms placed in the bottom of the main structure. The robot is previously modeled individually in SolidWorks, obtaining as a result a file of assemble *.sldasm. In accordance with the assembly, it is required to establish hierarchies of the aerial manipulator, including the number of items and restrictions of relative position of each part that makes up the robot. For the orientation and location of the parts of the model, points of reference are determined. Finally, the design is converted in a compatible model with the platform of UNITY 3D, using the software 3DS Max that allows to export the file obtained from SolidWorks to a file with the extension *.fbx.

### 4.1.1 Structural Analysis

The behavior of the structure of the aerial manipulator is modeled by the finite element method (FEM) [27], that allows to predict failures due to load conditions to which the design is subjected, showing the distribution of stresses in the material, stresses, allowable values of tensions, security factor, modulus of elasticity, tendency to deform, vibrate. The tasks necessary to perform the structural analysis are: *(i) Pre-process*, consists of the definition of geometry, generation of the mesh and assignment of properties to the materials. For the aerial manipulator, it is considered to be made up of carbon fiber Hexcel 3 K [28]; *(ii) structural analysis*, point and torque forces are applied, finally, *(iii) Post process*, in this analysis we observe the maximum and minimum values of the tension that the aerial manipulator supports, in order to avoid exceeding its elastic limit, static displacement, unit deformation and safety factor. The results of the post-process are indicated in Table 1.

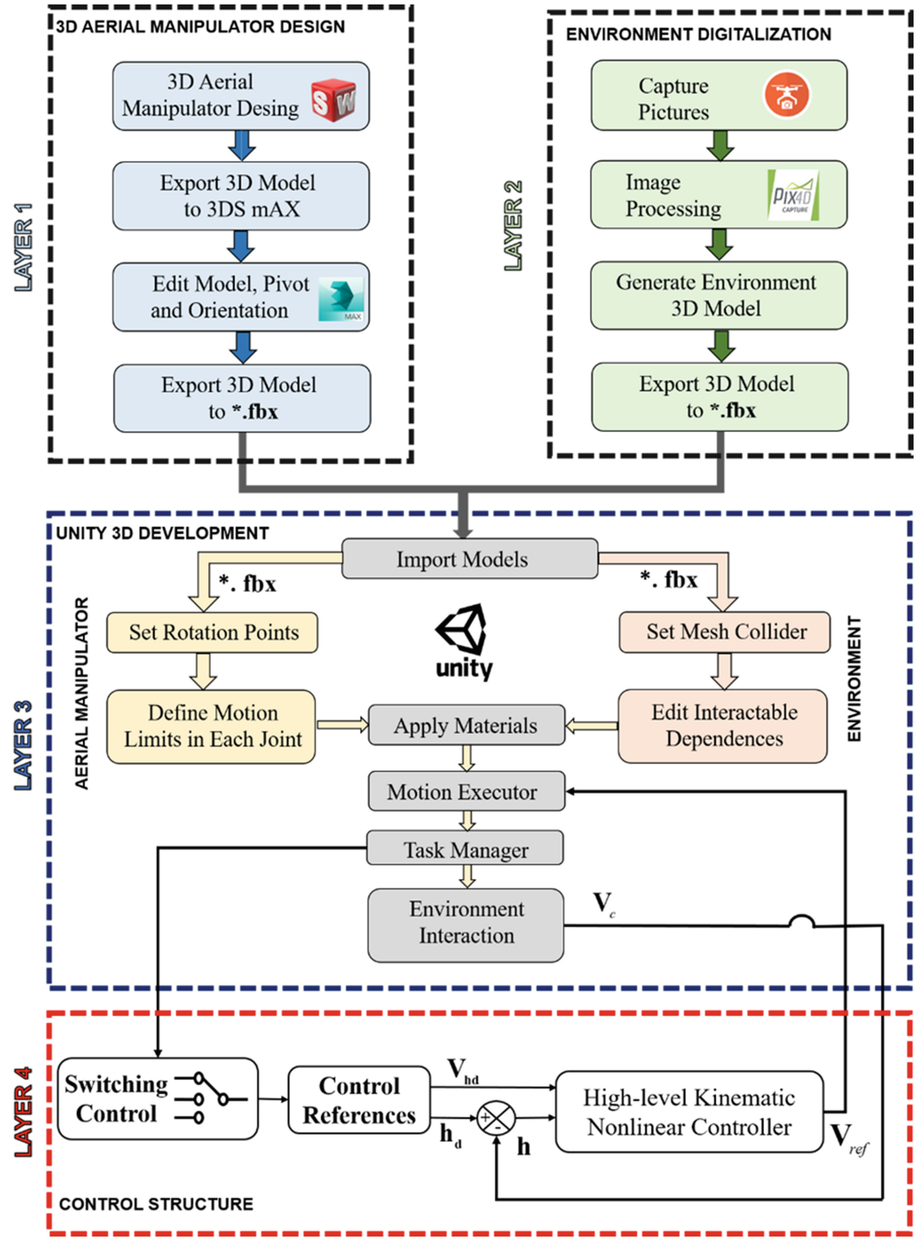



**Fig. 2.** Multilayer Diagram-Design of the 3D Simulator.

In the analysis, an applied force of 25 [N] and a moment of 5 [Nm] was considered; it is obtained as a result that the aerial manipulator does not have stresses that exceed the elastic limit, the maximum displacement value of the manipulator is 22.6 [cm],

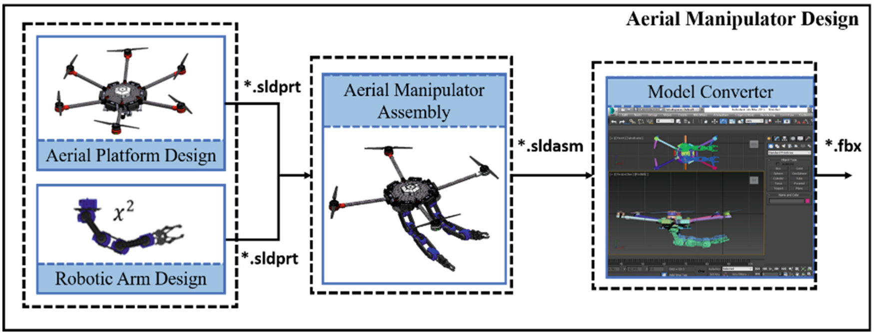
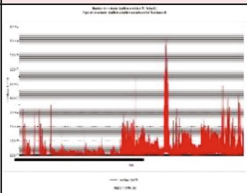
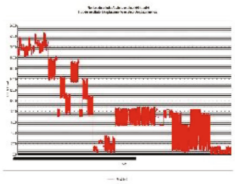
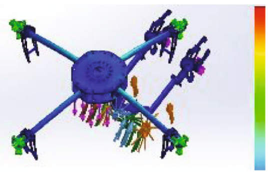
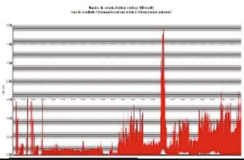
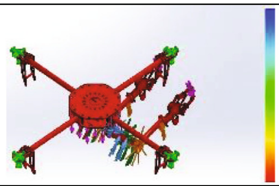
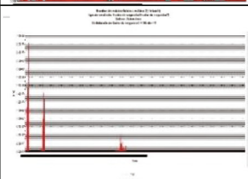



**Fig. 3.** Scheme of the design of the manipulator aerial 3D

**Table 1.** Result of structural analysis of the aerial manipulator

| Analysis Type | Aerial Manipulator | Results | Min. and Max. values |
|---|---|---|---|
| Nodal Tension (von MISES) |  |  | Max.Value 810390272 (N/m^2) Min Value 9.11 (N/m^2) |
| Static Displacement |  |  | Max.Value 226.31 mm Min Value 0 mm |
| Unitary Deformation |  |  | Max. Value 0.06834 Min Value 0 |
| Security Factory |  |  | Max. Value 263,472,256 Min Value 2.96 |

according to the deformation analysis it is determined that it does not have a unitary deformation and finally the Safety factor is constant over the entire surface.

### 4.2 Environment Digitalization

In the second stage of development, the digital model of the environment is created using the method of 3D mapping with drones, which consists in generating a digital terrain model (MDT) by means of the processing of photographs obtained with a drone. In Fig. 4, a block diagram of how the 3D mapping of a real environment is performed is shown.

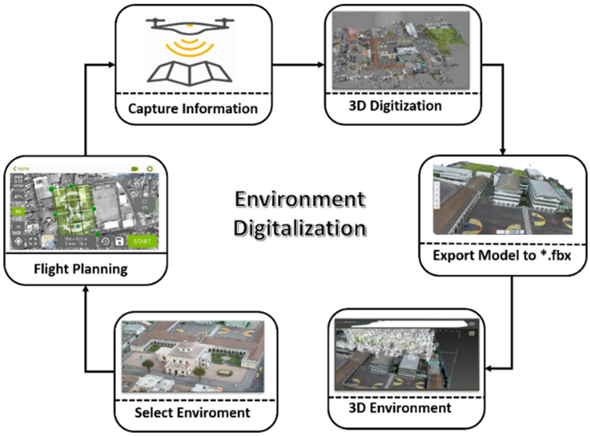



**Fig. 4.** Diagram of the process of digitalization of a real environment.

The digitalization of the environment starts for choosing the place and analyze it in such a way that is known the appropriate height to overfly with the drone and capture as many photos as possible of the place of interest, also consider conditions in which the terrain is suitable and without much presence of people, animals or other objects that affect the model. Finally, it is necessary take into account an appropriate hour in which a good photographic is captured without the presence of shadow of the environment of interest to map. For the lifting of photographic information, it is indispensable count on a drone that allows flight routes to be planned and incorporates in its physical part a camera with high resolution characteristics. Pix4D offers software and applications which facilitate the lifting and processing of information of a terrain, since it has a developed flight planning App for optimal mapping data with the drone (Pix4Dcapture) and desktop application to generate the 3D model from the information obtained by the UAV.

By means of the restored information of the terrain to be mapped, drone flight plans are set in the application as is shown in Fig. 5, The Smartphone application allows to zoom in on the site of interest and store the photographic images taken sequentially. The flight plans more used for 3D mapping are the Double Grid, Circular and Free Flight.

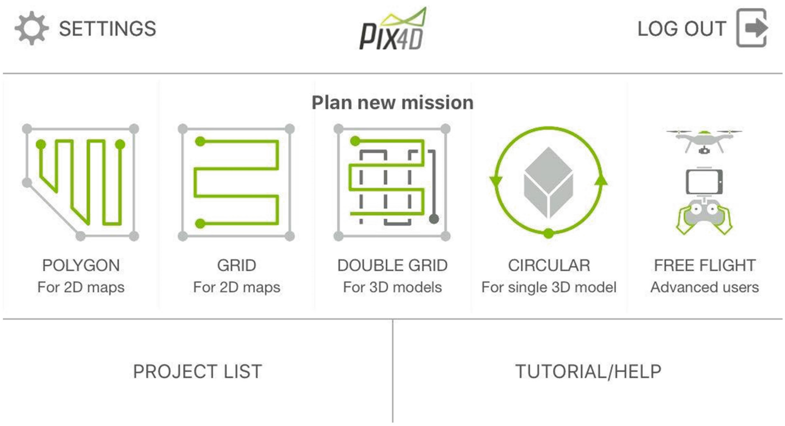



**Fig. 5.** Type of mission plans in PIx4D.

***Remark 1:*** The “Double Grid” flight plan is used for the wide-land model in places where there is a large number of elevations or buildings. The “Circular” flight plans are combined combining flights with “Free Flights” to capture images of difficult places.

***Remark 2:*** In consideration of the capture of photographs suitable for the 3D model, the angle of inclination of the camera must be adequate, thus achieving the total capturing of all details present in the terrain Figure 6.

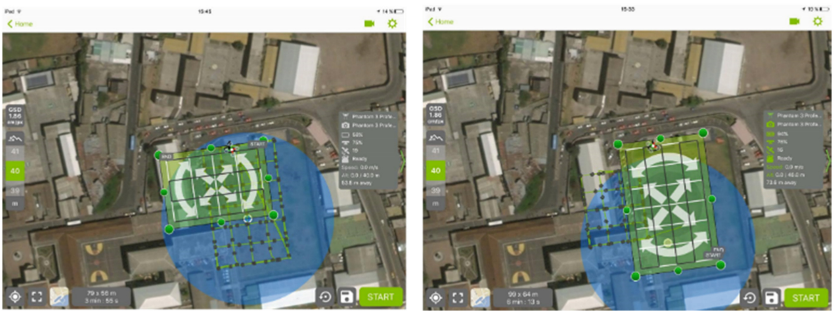


**Fig. 6.** Lifting photographic information with the drone.

Once is captured as many photos of the terrain, the images are processed in the software Pix4D, Fig. 6. In which all the images obtained with the drone are added. These images possess metadata, storing the referenced information of each photograph and the characteristic of the photo acquired through the flight. The processing of information takes time as it is responsible for processing each image and assemble the 3D model of the land (Fig. 7).

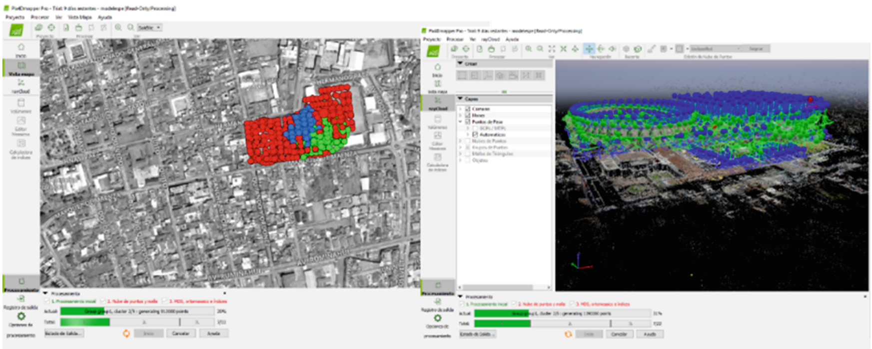


**Fig. 7.** Digital processing of the information raised in Pix4D.

With the digitized model, it is finally exported in *.fbx format, readable format for the UNITY 3D platform, the digitized terrain will represent the simulation environment of the aerial manipulator Fig. 8.

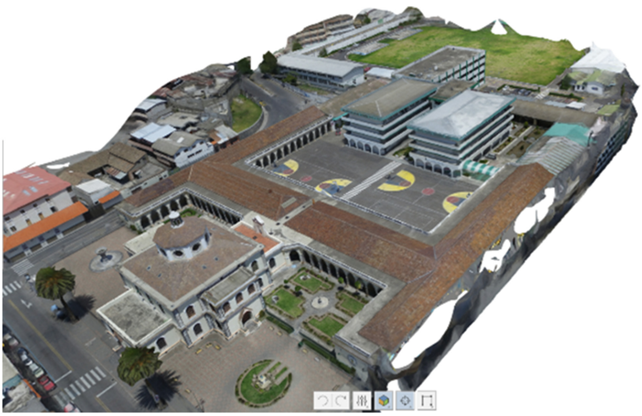


**Fig. 8.** 3D digital model of the terrain in format *.fbx.

### 4.3 Unity 3D Development

At the stage of the application development in Unity 3D, they are imported the 3D models that are part of the simulator. In the case of the aerial manipulator, the rotation points of each joint are placed and defines the minimum and maximum values of each part of the robot Fig. 9 can work are defined. For the creation of the scenario, the 3D model of the digitized terrain is imported, adding physical properties to be able to move and collide objects Fig. 10.

An operation interface is developed where the user selects the type of task to be performed with the aerial manipulator, sending the data of the selected task to Matlab in a shared way to carry out the control Fig. 11, in the same way the system allows to locate the manipulator to establish the initial conditions of the Robot.

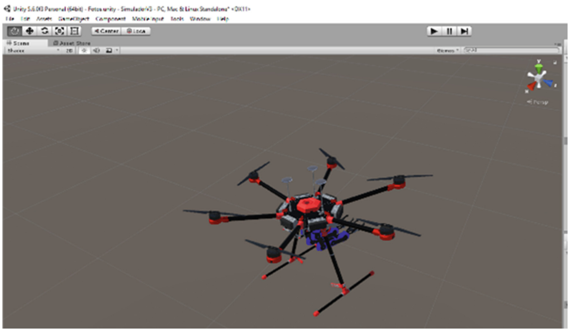


**Fig. 9.** Import model of the aerial manipulator to Unity 3D.

**Fig. 10.** Import of the digital terrain model to Unity 3D.


**Fig. 11.** Task management menu.

## 4.4 Control Structure

Control is designed under the Matlab software, allowing the programming of control algorithms. The control structure is feedback with the behavior of the aerial manipulator within the virtual environment. In the same way, the controller sends the respective control actions to correct the error. The control scheme proposed according to Fig. 12 in which it can be seen that depending on the task, Matlab will execute the appropriate control to comply with the user's order.

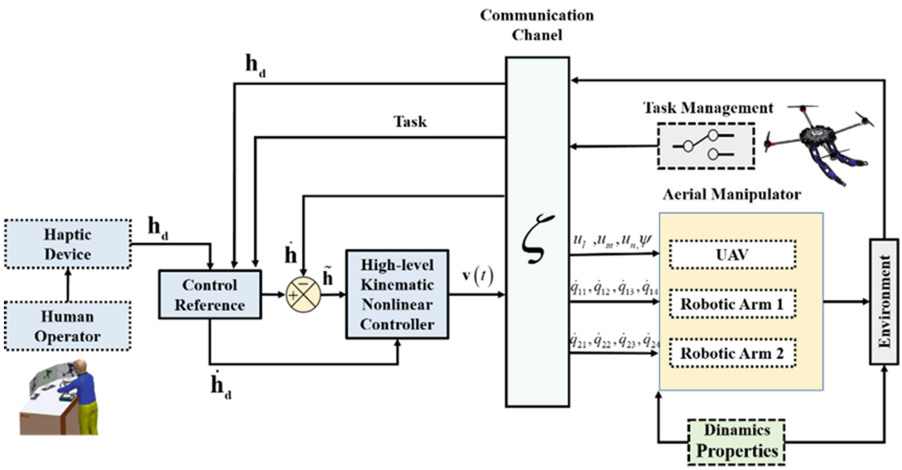



**Fig. 12.** Aerial manipulator control scheme for tele-operation and autonomous control.

The control stage is formulated in such a way that the operator can interact to control the robotic arms manually with a haptic device, through which direct reference commands are given to the controller to reach the object of interest with the aerial manipulator. If the task is a navigation one, the controller automatically generates the references so that the aerial manipulator can be positioned autonomously in the position desired by the operator. For the purpose of aerial manipulator control, the kinematic model is used which results in the location of the final effectors according to the configuration of the robotic arms and the location of the air vehicle (or its operational coordinates as functions of the generalized coordinates of the two robotic arms and the operational coordinates of the air vehicle), represented by:

$$\dot{\mathbf{h}}(t) = \mathbf{J}\left(\mathbf{q_q}, \mathbf{q}_{\mathbf{a}1}, \mathbf{q}_{\mathbf{a}2}\right)\mathbf{v}(t) \qquad (1)$$

where, $\dot{\mathbf{h}}(t)$ is the velocity vector of the final effectors represented by, $\mathbf{v} = \left[\mathbf{v_q} \mathbf{v_{qa1}} \mathbf{v_{qa2}}\right]^T$ is the mobility control vector of the mobile manipulator and $\mathbf{J}\left(\mathbf{q_q}, \mathbf{q_{a1}}, \mathbf{q_{a2}}\right)$ is the Jacobin matrix that defines a linear mapping between the aerial manipulator velocity vector and the velocity vector of the final effectors. The kinematic model is composed of a set of twelve speeds represented in the spatial reference frame $\langle Q \rangle$. The movement of the airborne mobile manipulator is guided by the three linear speeds $u_l$, $u_m$ and $u_n$, y, and is defined in a spatial frame of reference and the angular velocity $\omega$, as shown in the Fig. 13.

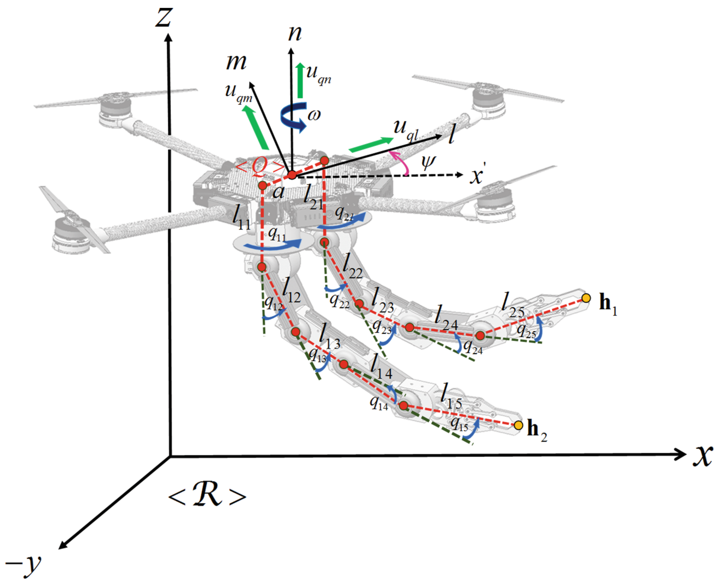



**Fig. 13.** Analysis of Aerial Manipulator Kinematics.

The kinematic model of the manipulator is defined as follows:

$$\begin{bmatrix} \dot{h}x_1 \\ \dot{h}y_1 \\ \dot{h}z_1 \\ \dot{h}x_2 \\ \dot{h}x_2 \\ \dot{h}x_2 \end{bmatrix} = [\mathbf{J}(\mathbf{q},\psi)][ul\, um\, un\, \omega\, \dot{q}_{11}\, \dot{q}_{12}\, \dot{q}_{13}\, \dot{q}_{14}\, \dot{q}_{21}\, \dot{q}_{22}\, \dot{q}_{23}\, \dot{q}_{24}]^{\mathbf{T}} \tag{2}$$

where, $\mathbf{J}(\mathbf{q}) \in \Re^{mxn}$ con $m = 6$ y $n = 12$ represents the Jacobian matrix that defines a linear mapping between the velocity vector of the mobile aerial manipulator $\mathbf{v} \in \Re^{n}$ where $\mathbf{v} = [u_l\, u_m\, u_n\, \psi\, \dot{q}_{11}\, \dot{q}_{12}\, \dot{q}_{13}\, \dot{q}_{14} \quad \dot{q}_{21} \quad \dot{q}_{22} \quad \dot{q}_{23} \quad \dot{q}_{24}\,]^{T}$ and the speed vector of the operating end $\dot{\mathbf{h}} \in \Re^{m}$ where $\dot{\mathbf{h}} = \left[\, \dot{h}_{x1} \quad \dot{h}_{y1} \quad \dot{h}_{z1} \quad \dot{h}_{x2} \quad \dot{h}_{y2} \quad \dot{h}_{z2} \,\right]^{T}$.

## 5 Experimentals Results

To carry out the simulation tests, it is necessary to implement a control law based on the kinematics (2) of the aerial manipulator described in the previous section, which is defined as:

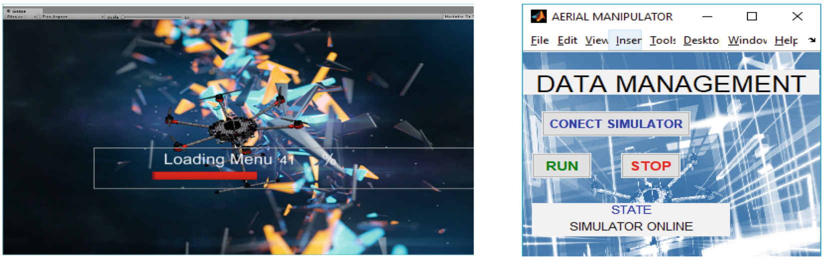
$$\mathbf{v} = \mathbf{J}^{\#}\left(\dot{\mathbf{h}}_{\mathbf{d}} + \mathbf{K_1}\tanh\left(\mathbf{K}_2\,\tilde{\mathbf{h}}\right)\right) \tag{3}$$

where $\dot{h}_d$ is the reference velocity input of the aerial mobile manipulator for the controller; $\mathbf{J}^{\#}$ is the matrix of pseudoinverse kinematics for the aerial mobile manipulator; while that $\mathbf{K_1} > 0$ and $\mathbf{K}_2 > 0$ area gain constants of the controller that weigh the control error respect to the inertial frame $<\mathrm{R}>$; and the **tanh**(.) represents the function saturation of maniobrability velocities in the aerial mobile manipulator [14].

With the purpose of validating the performance of the simulator and the control law implemented, two experiments are carried out, in which the behaviour of the aerial manipulator is observed within the virtual environment.

**Experiment 1:**

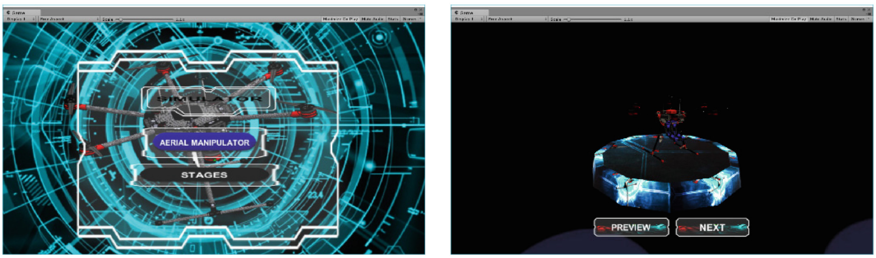
The first experiment consists of designating that the aerial manipulator positions itself autonomously at a point designated by the operator, enters the simulator waiting for all the necessary data to be loaded and selects what we want following the developed menu of the simulator including the Matlab part to establish the data link (Fig. 14).




a) Start of the virtual simulator b) Data management in Matlab

**Fig. 14.** Beginning of the communication between Matlab and Unity.

Once the communication with Matlab is loaded and established, the aerial manipulator is selected in the simulator menu Fig.15.




a) Simulator Menu. b) Select Aerial Manipulator.

**Fig. 15.** Data loading from the simulator and the aerial manipulator.

With the selection of the aerial manipulator for experiment 1, proceed to the "Task Management" menu and select the Autonomous Navigation Task Fig. 16, establishing the initial conditions of the aerial manipulator and the desired position to be reached autonomously Fig. 17.


**Fig. 16.** Selection autonomous navigation in task management menu.

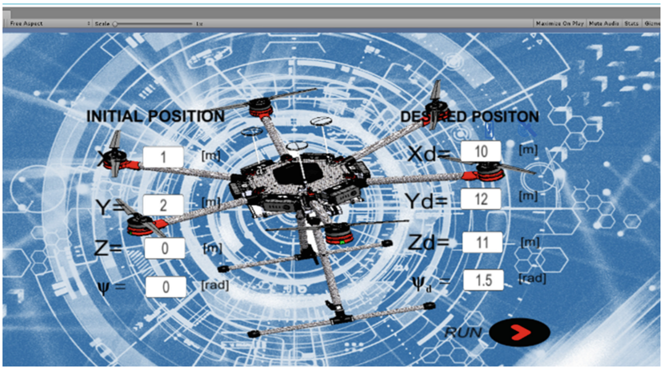



**Fig. 17.** Initial conditions of the aerial manipulator and the desired positions.

The data entered in the simulator are received in Matlab and the proposed task is executed, in Fig. 18, indicating the stroboscopic movement of the aerial manipulator movement within the virtual environment, observing the speeds applied and the positions of the current operating ends of the aerial manipulator.

**Experiment 2:**

The second experiment consists of the execution of a teleoperation task to transport an object from one place to another, using haptic devices to control the operating end of each robotic arm and at the same time command the mobility of the aerial platform. In the Task Management menu, we choose the Manipulation option and the user proceeds

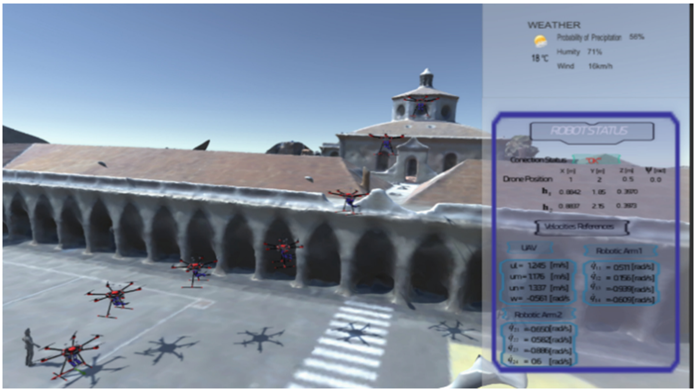
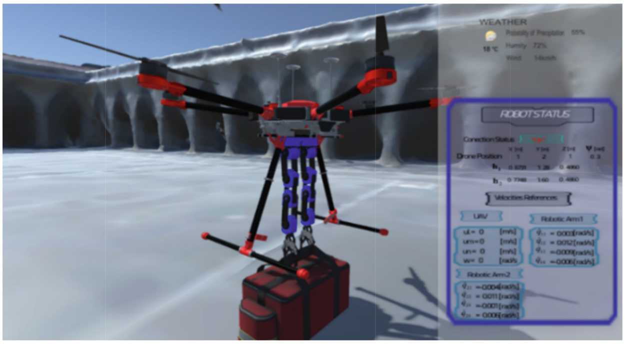


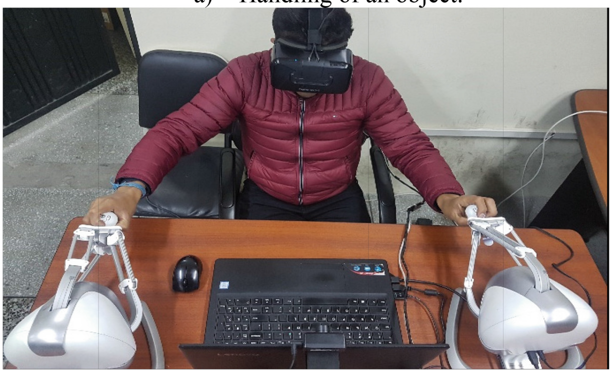

**Fig. 18.** Autonomous control operation for positioning the aerial manipulator in the virtual environment.

to tele-operate the robot, managing to hold the object with the operating ends of the manipulator as shown in Fig. 19.

When the operator is able to hold the desired object with the operating ends, he proceeds to transport it, controlling in this case the speeds towards the aerial platform to move to the point desired by the user Fig. 20.

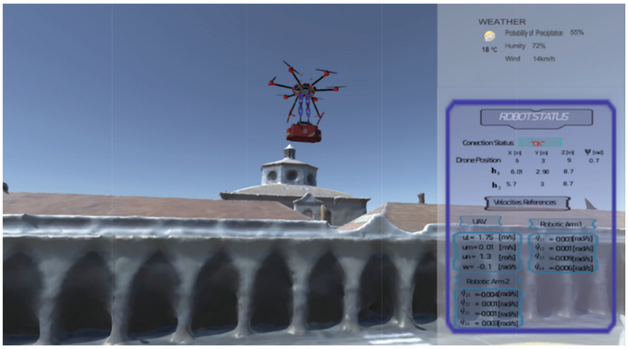



a) Handling of an object.



b) User teleoperating the Aerial manipulator.

**Fig. 19.** Teleoperation of the aerial manipulator to transport an object.




a) Transport of object with the aerial manipulator.


b) Handling of the aerial manipulator with the Haptic device.

**Fig. 20.** Transport of an object through control with a Haptic device.

## 6 Conclusions

To make the 3D simulator of an aerial vehicle and two robotic arms, several 3D design tools were linked, in the digital terrain model Unity is used in conjunction with a CAD software obtaining an environment very close to the real one. In the handling of the manipulator, a haptic device is used which simulates the transport and handling of objects with the different control strategies verifying the ability to respond to disturbances as meteorological data from a real environment.

**Acknowledgements.** The authors would like to thanks to the Corporación Ecuatoriana para el Desarrollo de la Investigación y Academia–CEDIA for the financing given to research, development, and innovation, through the CEPRA projects, especially the project CEPRA-XI-2017-06; *Control Coordinado Multi-operador aplicado a un robot Manipulador Aéreo;* also to Universidad de las Fuerzas Armadas ESPE, Universidad Técnica de Ambato, Escuela Superior Politécnica de Chimborazo, and Universidad Nacional de Chimborazo, and Grupo de Investigación en Automatización, Robótica y Sistemas Inteligentes, GI-ARSI, for the support to develop this work.

## References

1. Andaluz, Víctor H., et al.: Robust control with dynamic compensation for human-wheelchair system. In: Zhang, X., Liu, H., Chen, Z., Wang, N. (eds.) ICIRA 2014. LNCS (LNAI), vol. 8917, pp. 376–389. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13966-1_37
2. Jayawardena, C., Kuo., Unger, I., Igic, A., Wong, R., Watson, C., Stafford, R., Broadbent, E., Tiwari, P., Warren, J., Sohn, J., MacDonald, B.: Deployment of a service robot to help older people. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2010)
3. Andaluz, Víctor H., Ortiz, Jessica S., Sanchéz, Jorge S.: Bilateral control of a robotic arm through brain signals. In: De Paolis, L.T., Mongelli, A. (eds.) AVR 2015. LNCS, vol. 9254, pp. 355–368. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22888-4_26
4. Doriya, R., Chakraborty, P., Nandi, G.: Robotic services in cloud computing paradigm. In: 2012 International Symposium on Cloud and Services Computing (2012)
5. Pillajo, C.: Calculation of SCARA manipulator optimal path subject to constraints. In: II International Congress of Engineering Mechatronics and Automation (2013)
6. Naldi, R., Gentili, L., Marconi, L.: Modeling and control of the interaction between flying robots and the environment. IFAC Proc. Vol. **43**(14), 975–980 (2010)
7. Fumagalli, M., Naldi, R., Macchelli, A., Forte, F., Keemink, A., Stramigioli, S., Carloni, R., Marconi, L.: Developing an aerial manipulator prototype: physical interaction with the environment. IEEE Robot. Autom. Mag. **21**(3), 41–50 (2014)
8. Orsag, M., Korpela, C., Bogdan, S.: Valve turning using a dual-arm aerial manipulator. In: Unmanned Aircraft Systems (ICUAS), vol. 2014(1), pp. 1–7 (2014)
9. Tsukagoshi, H., Watanabe, M., Hamada, T., Ashlih, D., Iizuka, R.: Aerial manipulator with perching and door-opening capability. Robot. Autom. (ICRA) **2015**(1), 4663–4668 (2015)
10. Marconi, L., Basile, F., Caprari, G., Carloni, R., Chiacchio, P., Hurzeler, C., Lippiello, V., Naldi, R., Nikolic, J., Siciliano, B., Stramigioli, S., Zwicker, E.: Aerial service robotics: the AIRobots perspective. In: 2nd International Conference on Applied Robotics for the Power Industry CARPI (2012)
11. Fumagalli, M., Naldi, R., Macchelli, A., Forte, F., Keemink, A.Q., Stramigioli, S., Carloni, R., Marconi, L.: Developing an aerial manipulator prototype: physical interaction with the environment. IEEE Robot. Autom. Mag. **21**(3), 41–50 (2014)
12. Albers, A., Trautmann, S., Howard, T., Nguyen, T.A., Frietsch, M., Sauter, C.: Semi-autonomous flying robot for physical interaction with environment. In: IEEE Conference on Robotics, Automation and Mechatronics (2010)
13. Guerrero-Sanchez, M., Abaunza, H., Castillo, P., Lozano, R., Garcia-Beltran, C., Rodriguez-Palacios, A.: Passivity-based control for a micro air vehicle using unit quaternions. Appl. Sci. **7**(1), 13 (2016)
14. Ortiz, Jessica S., et al.: Modeling and kinematic nonlinear control of aerial mobile manipulators. In: Zeghloul, S., Romdhane, L., Laribi, M.A. (eds.) Computational Kinematics. MMS, vol. 50, pp. 87–95. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-60867-9_11
15. Andaluz, Víctor H., et al.: Modeling and control of a wheelchair considering center of mass lateral displacements. In: Liu, H., Kubota, N., Zhu, X., Dillmann, R., Zhou, D. (eds.) ICIRA 2015. LNCS (LNAI), vol. 9246, pp. 254–270. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22873-0_23
16. Wang, S., Mao, Z., Zeng, C., Gong, H., Li, S., Chen, B.: A new method of virtual reality based on Unity3D. In: 8th International Conference on Geoinformatics (2010)

17. Oliveira, M., Pereira, N., Oliveira, E., Almeida, J. E., Rossetti, R.J.: A multi-player approach in serious games: testing pedestrian fire evacuation scenarios. In: Oporto, DSIE15, January 2008
18. Indraprastha, A., Shinozaki, M.: The investigation on using Unity3D game engine in urban de-signstudy. J. ICT Res. Appl. **3**(1), 1–18 (2009)
19. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: dense 3d reconstruction in real-time. In: Intelligent Vehicles Symposium (IV), vol. 2011(1), pp. 1–6 (2011)
20. Flores, D.A., Saito, C., Paredes, J.A., Trujillano, F.: Aerial photography for 3D reconstruction in the Peruvian Highlands through a fixed-wing UAV system. In: Mechatronics (ICM), vol. 2017(1), pp. 1–7 (2017)
21. Kurnaz, S., Cetin, O., Kaynak, O.: Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles (2018)
22. Sorton, E., Hammaker, S.: simulated flight testing of an autonomous unmanned aerial vehicle using FlightGear (2018)
23. Kurnaz, S., Cetin, O., Kaynak, O.: Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles. J. Intell. Robot. Syst. **54**, 229–244 (2018)
24. Psirofonia, P., Samaritakis, V., Eliopoulos, P., Potamitis, I.: Use of Unmanned Aerial Vehicles for Agricultural Applications with Emphasis on Crop Protection: Three Novel Case - studies (2018)
25. Trujano, G.B.R.R.F., Chan, B., Beams, G., Rivera, R.: Security Analysis of DJI Phantom 3Standard (2016)
26. Gerkey, B.P., Vaughan, R.T., Howard, A.: Tools for multi-robot and distributed sensor systems. In: Proceedings of the International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal, pp. 317–323, 30 June–3 July 2003
27. Figueroa, R., Müller-Karger, C.: Effort analysis by the finite element method in the artificial foot design process. In: IV Latin American Congress on Biomedical Engineering 2007, Bioengineering Solutions for Latin America Health, pp. 732–735 (2007)
28. Xu, Y., Hoa, S.: Mechanical properties of carbon fiber reinforced epoxy/clay nanocomposites. Compos. Sci. Technol. **68**(3–4), 854–861 (2008)