2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

A Hardware-in-loop Platform for Rotary-Wing Unmanned Aerial Vehicles

Igor Henrique Beloti Pizetta[1], Alexandre Santos Brandão[2], and Mário Sarcinelli-Filho[3]

***Abstract*— This work describes the development of a platform to deal with simulated and real autonomous flights of rotary-wing aircrafts. Such platform (called *AuRoRa Platform*) was designed for use with commercial miniature helicopters and embeds the instrumentation necessary to autonomously guide the aircraft. Aiming at such objective, an electronic board specific to actuate over the servomotors of the helicopter and to get the data delivered by the sensors onboard the aircraft (all them "components of the shelf"), called *AuRoRa Board* was designed and manufactured. The *AuRoRa Platform* was implemented to exchange information with such a board, as a high level Hardware-in-the-Loop platform, capable of running simulations and real experiments, in the last case actuating over the servomotors of the aircraft through the electronic board aforementioned. Using the *AuRoRa Platform* one can choose amongst various aircrafts (we have already tested it with the quadrotor AR.Drone Parrot and the miniature helicopters ALIGN T-REX 450 and T-REX 600 - these two latter ones are the aircrafts dealt with in this work) and amongst different controllers as well. The *AuRoRa Platform* also presents a decentralized characteristic: to avoid overloading a single computer with the synthesis of the control signals and the online exhibition of the flight data, a feature enabled/disbled by the user, it is possible to run the online exhibition of the flight data in a second computer, using an UDP communication channel.**

I. Introduction

Over recent years there has been a growing interest in unmanned aerial vehicles (UAVs), associated to the increased processing capacity of micro controlled systems, leading to smaller embedded systems. The most remarkable dissemination in the scientific community is related to UAV rotorcrafts (helicopter and quad-rotor) due to their ability to take off and land vertically, to hover with the possibility of changing its orientation, to move ahead or laterally while keeping the height, to completely change their direction of flight and to stop their movement abruptly. Despite these considerable advantages, in comparison with fixed-wing UAVs (airplanes and gliders) or even with lighter-than-air UAVs (balloons and airships), rotary-wing UAVs are machines extremely complex to control, because of the inherent difficulty to generate propulsion and simultaneously move itself [1], [2].

However, their high mobility makes the rotary-wing UAVs very useful machines to accomplish three-dimensional tasks in places of difficult access and high risk, or even over large areas. Amongst such tasks, one can cite those related to public safety (supervision of air space and urban traffic), management of natural hazards (like volcanoes), environmental management (measurement of air pollution and monitoring of forests), intervention in hostile environments, maintenance of infrastructure (transmission lines, fluid or gas pipes), and precision agriculture. In such tasks, the UAVs are more efficient and their use is more advantageous, in comparison with unmanned ground vehicles, for instance, because of the vertical maneuverability [3], [4].

The aforementioned features have motivated many researchers from several areas to propose models and flight controllers capable of guiding UAVs in several and diverse applications. However, from the point of view of control, these aircrafts are inherently unstable, non-linear, multi-variable, with complex and highly coupled dynamics. Due to these characteristics, an experimental test of UAVs generates a great risk, not only for the equipment under development, but also to people nearby it (after taking off, such vehicles can fly with great speed and their blades can reach high velocities). Such issues have motivated the creation of quite complex simulators, mimicking the reality quite well, which are of extreme importance in the development of UAV flight control systems.

According to [5], two approaches can be adopted to develop a simulator: the *software-in-loop* (SIL) one, in which all components are simulated, i.e., sensors, actuators, aircraft model and so on, and the *hardware-in-loop* (HIL) one, in which just some parts are simulated, whereas other parts correspond to real pieces that are effectively operating acting on the aircraft (e.g., sensors or even actuators).

Simulation reduces the time spent to develop a technology. In other words, one can perform tests in a faster and less tiring way than running real experiments. Using HIL and simulations, one can validate a system

*This work was supported by CNPq - a Brazilian Agency supporting the scientific and technological development, and FAPES - an agency of the State of Espírito Santo supporting the scientific and technological development.

[1]I. H. B. Pizetta is with the Department of Mechanics, Federal Institute of Espírito Santo, Aracruz - ES, Brazil igor.pizetta@ifes.edu.br

[2]A. S. Brandão is with the Department of Electrical Engineering, Federal University of Viçosa, Viçosa - MG, Brazil alexandre.brandao@ufv.br

[3]M. Sarcinelli-Filho is with the Graduate Program on Electrical Engineering, Federal University of Espírito Santo, Vitória - ES, Brazil mario.sarcinelli@ufes.br

prior to an experimental test, and develop hardware and software that are compatible, avoiding injuring people and damaging equipment. In particular, HIL simulators are those whose simulation loop deals with the representation of the environment in which the hardware will work. Thus, after executing tests with a sufficient number of environmental conditions and analyzing the behavior of the hardware, a comprehensive assessment about the system can be done. Notice that it is the last step before a real experiment, since one should have the hardware to simulate the environment. Such approach is presented in [6], [7], [8]. In contrast, the SIL simulation does not depend directly on any hardware piece, since it is being developed to compose the onboard system. Examples of simulators using this approach can be found in [9], [10], [11].

In such context, this paper presents the development of a HIL platform, capable of simulating and testing UAVs with different mathematical models and control strategies, developed starting from smaller versions used in other works of our research group [12], [13], [14]. To give details about such development, the paper is hereinafter split in some sections. Section II presents an overview of the aircraft model, emphasizing the control variables and the degrees of freedom available for controlling. In the sequel, Section III gives a description of the main sensors available onboard the aircraft, emphasizing a specific board designed and manufactured to be the interface between the controller and the sensors/actuators of the aircraft. Following, Section IV describes the structure of the platform here implemented, emphasizing its distributed implementation, and Section V shows some examples of simulated and real flights managed by the proposed platform. Finally, Section VI highlights the main conclusions of the work.

II. Miniature Helicopter Model

To design a controller to guide the autonomous flight of an aircraft it is necessary to have its model. Although not being the main objective of this work, the model of the helicopter is required to understand how the proposed platform operates. For more details on this model, the interested reader should access [14] and [15], for instance.

The UAV is a solid body subject to external forces and torques, generated by its propellers. Therefore, it is a rigid body moving in the three-dimensional space. Two approaches can be adopted to describe it: Newton-Euler equations [16], [17], [18] or Euler-Lagrange equations [19], [20], [15]. Both approaches lead to the dynamic model of the rigid body. The main difference between them is the reference system: the Newton-Euler approach considers the reference system whose origin is the center of mass of the aircraft, whereas the Euler-Lagrange approach adopts the inertial frame as reference system.

The model adopted in the platform here described was developed using the Euler-Lagrange approach (other models could be adopted without any problem), and is described by [15]

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \tau + \mathbf{D}, \qquad (1)$$

where $\mathbf{q}$ is the vector containing the attitude of the aircraft, $\mathbf{M}$ is the matrix of inertia, $\mathbf{C}$ is the matrix of centripetal forces and Coriolis, $\mathbf{F}$ is the friction vector, $\mathbf{G}$ is the vector of gravitational forces, τ is the vector with the control signals applied to the vehicle and $\mathbf{D}$ is a vector of disturbances.

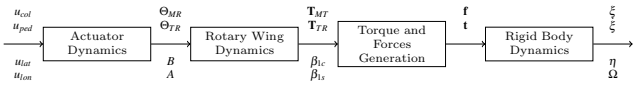


Fig. 1. Block representation of the Helicopter dynamic model.

According to [21], [22], [23], the whole model of a miniature helicopter is represented by four interconnected dynamic subsystems, shown in Figure 1, which are the actuator dynamics (responsible for transforming the *joystick* - or computer generated - commands in servomotor actuation), the rotary-wing dynamics (witch embeds the aerodynamic parameters and the thrust generation associated to the main and tail rotors), the forces and torques generation process (where occurs the thrust decomposition), and the rigid-body dynamics(which defines the rotorcraft displacement in the free space).

The block entitled Actuator Dynamics is where the commands are received and transformed in actuation commands for the servo motors. This can be remotely controlled by a human operator, or by a computer system onboard the aircraft or on the ground, in the case of autonomous flight. The second block, Rotary Wing Dynamics, relates the aerodynamic parameters and the propulsion generation associated with the tail and main engines. The following block, Torque and Forces Generation, decomposes the propulsion vectors, according the current orientation and geometry of the aircraft, and converts them into forces and torques applied to the vehicle. Finally, the Rigid Body Dynamics defines the displacement and the orientation of the aircraft in the three-dimensional space as a function of the applied forces and torques.

Therefore, the first three blocks shown in Figure 1 are responsible for receiving control signals and generating forces and torques which will effectively act on the aircraft. Such blocks correspond to what is here considered as low-level model. The last block receives abstract control signals, the forces and torques acting on the aircraft body, which will cause the movement of the rotorcraft in the 3-D space. Such block corresponds to the high-level model. In this work, however, just the low-level model is considered, since the objective is to describe how and why certain "components of the shelf"(COTS) are integrated to the aircraft, in connection with the developed platform.

As for the high-level model of the aircraft, it will not be directly addressed in this work, as well as the Rotary Wings Dynamics and Torques and Forces Generation. Actually, it is included in the complete model adopted to design the flight controller (discussed in [15]) used to test the implemented platform. Details on such topic, however, are out of the scope of this paper.

A. Principle of Operation of the Miniature Helicopter

To describe the helicopter dynamics, the notation and reference systems are now defined, for better understanding. The subscript e represents the terrestrial reference system (the inertial one), the subscript h refers to the reference system of the helicopter, centered at its center of mass, and the subscript s is associated to the spatial reference system, which is the h system rotated to have the same orientation of e (see Figure 2).

The attitude of the aircraft correspond to the position variables ξ_{he} (the subscript he indicates the helicopter reference system relative to the inertial frame), and the orientation variables η_{he}, relative to its own reference system. Therefore, one has that $\xi_{he} = [x\ y\ z] \in \Re^3$ represents the longitudinal, lateral and normal displacements, whereas $\eta_{he} = [\phi\ \theta\ \psi] \in \Re^3$ corresponds to the roll, pitch and yaw angles.

In order to establish the speeds, one only has to derive the position coordinates, resulting in $\dot{\xi}_{se} = [\dot{x}\ \dot{y}\ \dot{z}] \in \Re^3$. Deriving again one has the the accelerations $\ddot{\xi}_{se} = [\ddot{x}\ \ddot{y}\ \ddot{z}] \in \Re^3$. Notice that the reference frame is the spatial one (s) related to the terrestrial one (e). The same procedure is adopted for the angles, resulting in the angular velocities $\dot{\eta}_{sh} = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}] \in \Re^3$ and the angular accelerations $\ddot{\eta}_{sh} = [\ddot{\phi}\ \ddot{\theta}\ \ddot{\psi}] \in \Re^3$. The reference relationship is now between the spatial reference system and the reference system of the helicopter, both with origin at the center of gravity of the aircraft.

To perform the control of these variables, the operator has four input commands [22], namely

Airelon: (u_{lat}), which controls the main rotor lateral cyclic pitch (see Figure 3(a)) which produces the roll movement, resulting in the lateral displacement of the helicopter;

Elevator: (u_{lon}), which controls the main rotor longitudinal cyclic pitch (see Figure 3(b)), which causes the pitch of the aircraft, allowing it to move forward (or backward) in the longitudinal direction;

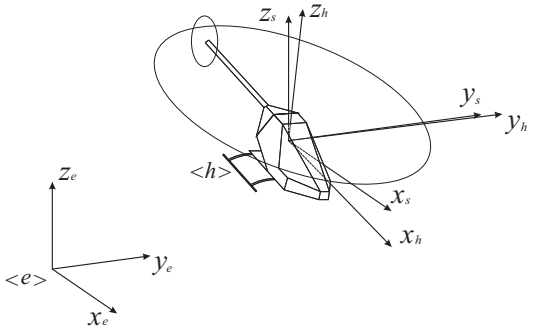
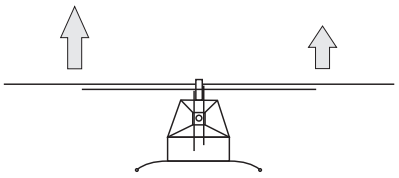
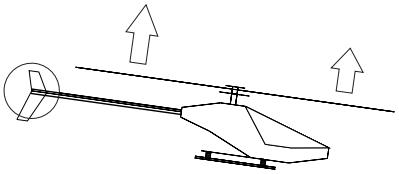


Fig. 2. The reference systems adopted.

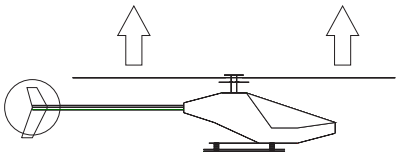


(a) Lateral cyclic pitch.

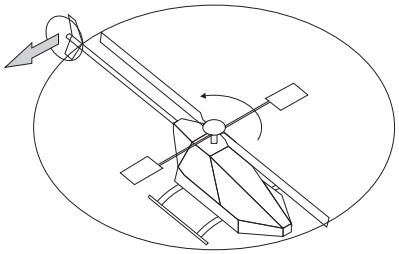


(b) Longitudinal cyclic pitch.

Fig. 3. Cyclic pitches.



(a) Main-rotor collective pitch.



(b) Tail-rotor collective pitch.

Fig. 4. Collective pitches.

Collective: (u_{col}), which controls the main rotor collective pitch (see Figure 4(a)), which results in the lift motion, causing the movement of the helicopter in the vertical direction;

Yaw: (u_{ped}), which controls the tail rotor collective pitch, necessary to compensate the anti-torque effect generated by the main rotor and to produce the yaw motion (see Figure 4(b)),

so that for autonomous systems the usual system input is the vector

$$\mathbf{u} = [u_{lat}\ u_{lon}\ u_{col}\ u_{ped}]. \tag{2}$$

On the other hand, the output variables are

$$\dot{\mathbf{q}} = [\dot{\xi}_s\ \dot{\eta}_s] = [\dot{x}_s\ \dot{y}_s\ \dot{z}_s\ \dot{\phi}_h\ \dot{\theta}_h\ \dot{\psi}_h]. \tag{3}$$

Looking for the input and output variables one realizes that there are more variables to be controlled than commands, what characterizes an underactuated coupled system, which is an important characteristic of rotorcrafts. The angular velocities $\dot{\phi}_h$, $\dot{\theta}_h$ and $\dot{\psi}_h$ are

controlled, respectively, by the entries u_{lat}, u_{lon} and u_{ped}, while the ascent and descent rate is controlled by the input u_{col}. Lateral and longitudinal movements $\dot{x}_s$ and $\dot{y}_s$ are obtained by moving the helicopter in order to produce a pitch, thus which causes the displacement in x, or a roll, which causes the displacement in y.

B. Actuator Dynamics

This subsection presents, in more details, the operation of the servo motors and the actuation system, since they are controlled by using the electronic board developed in this work (a micro-controlled board), and the way they act in the *swashplate* of the helicopter (they are the components of the block Actuator Dynamics).

There are several manufacturers and servomotors in the market, with different torques, speeds and sizes. However, all of them follow the same operation principle and have similar characteristics. The servomotors used here demand power supplies at voltages between $4,8V$ and $7,4V$, with $6kg.cm$ of torque. For a 60^o rotation it requires only $0.040s$, besides being very accurate devices, with 12-bit resolution. To perform its control, it is used a PWM signal with $22ms$ of duration, or $45Hz$ of frequency. A typical example of such signal is shown in Figure 5, where the useful control range is between $1ms$ and $2ms$. After this time the signal remains in low (idle) level.

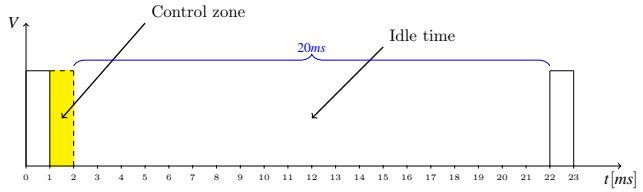


Fig. 5. PWM signal used in the control of the servomotors adopted.

The servomotors, installed onboard the miniature helicopter, are triggered by radio control. The radio generates a PPM (*Pulse Position Modulation*) signal of frequency $2,4Ghz$, which, when received by the receiver in the helicopter, is demodulated and decoded into a PWM signal for each channel (each servo motor, in a total of four, one for each controlled variable). Two radio controller models have been used, one having 7 and the other having 8 channels. Each channel carries different information. This is the main feature causing a small time interval to control the movement of one servo, because all channels should be accommodated in a message with the same frequency and time ($22ms$).

Figure 6 shows, in a simple way, how this signal is transmitted by the radio controller and the demodulation process (such example includes three channels). Each channel is associated to the time interval between two subsequent pulses, and the four of them are associated to PWM signals by arrival order, as exemplified in the figure.

Before explaining the radio control operation, regarding the movement generated, it is necessary to show

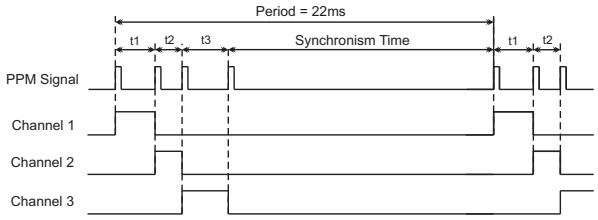


Fig. 6. Channel separation in the PPM signal.

how is the actuation system corresponding to the part between the servo motor and the rotary-wing, since they are in motion. The subject of this action is the *swashplate*. It has two main parts: a rotational part, which is connected with shovels and sway bars, and a fixed part, connected to the actuators or servo motors.

Usually, helicopters are equipped with stabilizing bars (*flybars*), whose purpose is to reduce the effect of external forces such as wind gusts, and smooth movements of the helicopter itself, increasing its inertia, thus making easier the pilotage. Because of the way they are connected to the *swashplate*, such bars only respond to cyclical movements, having no effect on collective movements.

As mentioned before, the *swashplate* is responsible for orienting the main rotor thrust. Figures 7 and 8 present the two simplest movements, which results only in the displacement of the *swashplate* up or down on its axis. Notice that the rudder also has a *swashplate*, but this one is much simpler, since its only function is to change the collective.

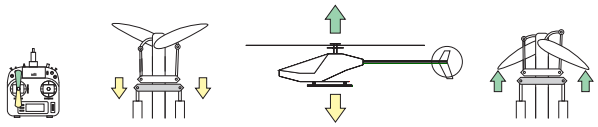


Fig. 7. Collective command diagram.

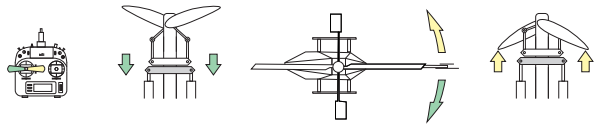


Fig. 8. Yaw command diagram.

In Radio Control the throttle is set with the collective, and the same stick performs the two commands. One can adjust the gain graph of both commands: for the case of an autonomous system this graph is adjusted to provide a saturated value of acceleration, thus respecting the modeling considerations.

The pitch and roll commands also requires the movement of the *swashplate*, but in a different way when compared to the collective. These can be seen in Figures 9 and 10 (notice that in 9 the reference is a side view of the *swashplate*, whereas in 10 it is a rear view), which illustrate longitudinal (pitch) and lateral (roll) movements, respectively.

Fig. 9. Pitch command diagram.

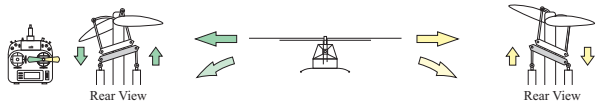


Fig. 10. Roll command diagram.

III. Flight Instrumentation

In order to guide autonomous flights of the aircraft, any control system will demand sensorial data. When flying in free-space a helicopter has six degrees of freedom, which allow it to translate along and to rotate around the three spatial axes X,Y,and$Z$. While performing such movements the aircraft is subject to position, velocity and acceleration changes, both linear and angular, which form a set of 18 variables. Thus, the goal of the sensory system is to provide information on the position and orientation of the vehicle relative to the global reference. Besides the sensory information, it is necessary to develop an actuation system to drive the servomotors, replacing the radio control, or in parallel with it (originally the only way to control the miniature helicopter used in this work was the radio control operated by a human being). Thus, a certain number of COTS subsystems were installed onboard the aircraft, together with an electronic board designed to integrate such COTS subsystems and the platform being developed, to allow the autonomous accomplishment of flight missions. Some of such subsystems are now described.

A. Low Level Stabilizer

There are some models of aircraft stabilizers available in the market. They are not intended to autonomously control the vehicle, but just to stabilize the miniature helicopter when hovering, and to keep it in a certain position if the radio control is in the neutral position (or the communication channel is lost). In this project it is used the Helicommand 3D®- HC, from the German manufacturer Captron, which recommends its use by novice pilots, because they do not have the expertise to guarantee a safe flight. This module was installed onboard the helicopter, along with the high-level controller and embedded system in order to increase the safety of the equipment and researchers, not only in the case of an automatic control, but also in the case of a manual control, if necessary.

The acceleration channel go directly to the helicopter ESC (main rotor speed controller), because to move the aircraft smoothly it is not necessary to control its speed (it is enough to use the other four control variables). Another fact is that the gyroscope of the miniature helicopter is connected in series with the stabilizer, which also has an internal gyroscope. Therefore, it is needed to adjust the gain of the two sensors, or to remove the gyroscope of the helicopter and to connect the servo directly into the rudder channel in the RC Module.

B. Sensors

The IMU (Inertial Measurement Unit) is the sensor responsible for providing the UAV attitude, i.e., the orientation variables (roll, pitch and yaw angles). However, it is necessary to filter the data coming from the accelerometers, gyroscope and magnetometer, because of the noisy measures they provide. In addition, by integrating the velocities one can get the position of the vehicle. To determine the altitude of the aircraft, it is used a ultrasound sensor pointing down, installed in the bottom of the vehicle, which measures the distance to the ground. In high altitudes, when the ultrasound is out of scale, it is used a barometric sensor, which measures the atmospheric pressure, whose value is converted to altitude through simple calculation. Such sensing modules also includes a GPS module, which is responsible for delivering the global position of the vehicle in outdoor flights.

C. Embedded System AuRoRa Board

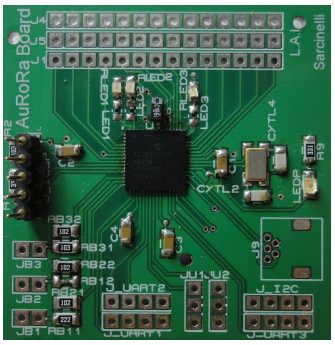


Fig. 11. Image of the electronic board *AuRoRa Board* (the connectors were omitted for better visualization).

In order to control the helicopter a specific circuit board was developed, called *AuRoRa Board*. It is part of the *AuRoRa Platform*, the platform dealt with here, and is responsible for capturing the data stemming from the radio control receiver and to send the actuation signals to the RC Module, which sends them to the servomotors. It also communicates with a high level embedded circuit and receives the sensorial data as well.

The Figure 12 shows the circuit schematic. The central object is a micro-controller, while the contacts on the top of the board are the input control signals, at the left, and outputs for the HeliCommand, at the right. At the bottom it has the sensor inputs and the communication channels, as UART and I^2C interfaces. Not shown in the schematic, such board has a JTAG connector at the left, to allow programming it.

Aiming at controlling the helicopter, it is necessary a micro-controller capable of reading at least 5 PWM

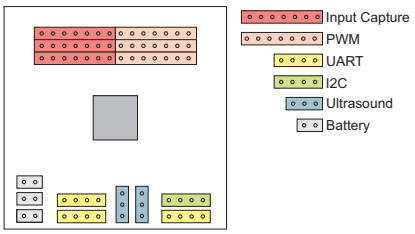


Fig. 12. Schematic of the *AuRoRa Board*.

inputs and to create the same amount. The RC Module sends to the board 7 PWM signals. Five of them correspond to the commands of roll, pitch, yaw, collective, and acceleration, and the others correspond to gyro and gear commands. The gyro one is used to adjust the gain of the gyroscope of the helicopter, and the gear one was modified to allow switching between the automatic, semi-automatic and manual control modes. In the automatic control mode the controller sends all the first 4 commands to the helicopter, while keeping the acceleration constant, set up by the human operator. In the semi-automatic control mode only the collective is automatically controlled, for safety reasons during the tests of controllers designed to autonomously guide the aircraft. Finally, in the manual control mode the helicopter is entirely controlled by the Radio Control, which is how the RC miniature helicopter used in this work was designed to operate.

After installing the *AuRoRa Board* in the helicopter, the full schematic of the connections is shown in Figure 13. Such a board is in between the RC Receiver and the Helicommand. Thus, it can read the data from the Radio Control and generate the data that will be sent to the servos as well, in manual or automatic control mode. Notice that as it reads all channels, including acceleration, the *AuRoRa Board* can fully control all channels of the miniature helicopter, not demanding any human action through a Radio Control, unless it is of interest.

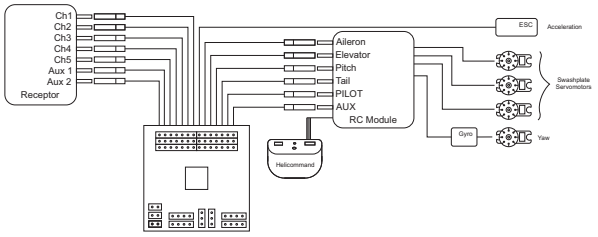


Fig. 13. Illustration of the connection of the *AuRoRa Board* to the signal receptor unit.

D. Data Transmission

It is worth mentioning that the developed system has different ways to communicate, which gives it versatility for use in various situations.

The first one is the UDP channel, which is a communication protocol that can be used between computers. It is a simple protocol, encapsulated in a IPv4 or IPv6 packet, but there is no guarantee that the packet will get its destination, i.e., there is no destination response confirming the message, what makes it a non-secure but fast protocol. In tasks where timing is important, as when running controllers, this protocol is recommended, due to its high velocity, compared to the TCP/IP protocol for instance. To guarantee some system security, however, some precautions were implemented in connection with the UDP protocol, such as the repetition of the previous message, if one message is lost, checksum and a exclusive network link between two computers for this communication channel, to guarantee that the channel will certainly be free. Also, there is a Zigbee connection, which has a good communication rate and can reach long distances, even more than $1000m$, depending on the model. Completing the communication possibilities, there is a Bluetooth channel, which has high velocity but short range (less than $10m$), used for indoor experiments or short-distance flights.

IV. The Structure of the *AuRoRa Plataform*

First of all, it is necessary to establish the conceptual differences between an online system and a real-time system. A system is said to be online with another one when the state change of one directly affects the other one, being them physically wired or not. In turn, a real-time system is one that guarantees actions and responses in well defined time intervals. For instance, consider the occurrence of a control action: any new control action can take place only after a fixed interval of time has elapsed, time interval that is constant along the system operation. Based on such characterization, one can conclude that the platform developed here is close to a real time system.

Algorithm 1 presents the structure of the developed platform. Notice that all system actions require a permission to run. *Execution permission* is given only when the sampling period of each simulated or tested UAV is reached. For instance, for a quad-rotor and a helicopter simulated simultaneously, the sampling periods are $t_{sq} = 1/30s$ and $t_{sh} = 1/50s$, respectively, so that the data reading/sending from/to each vehicle are independently performed, respecting those sampling times. This approach minimizes the computational effort by avoiding the execution of a series of instructions at each iteration, and avoids sending multiple control signals to vehicles in the same sampling period.

Observing Algorithm 1, one can notice that the platform is split into several modules, which can be activated or not. For example, one can enable or disable the generation of graphics, enable a joystick operation (for safety reasons, since its actuation overrides the implemented controller) and disregard some sensing subsystems (GPS or vision system, for instance), without compromising the task. Figure 14 shows how the data flows inside the platform, when operating in the *simulation* mode (using the *Virtual UAV* module, embedded in the platform)

or in the *experiment* mode (using sensorial information sent by the aircraft and sending the control signals it generates from the sensory data and the references to the aircraft). Notice that the first option corresponds to a SIL system, whereas the second one corresponds to a HIL one.

As previously mentioned, the platform is able to simulate or run experiments with different aircrafts (real, virtual or both) flying simultaneously, guided by joysticks or automatic controllers implemented onboard them or in external computers having a communication channel to and from the aircraft.

For a description of the operation of the platform, one can take, for instance, the quad-rotor AR.Drone Parrot, which has a set of onboard sensors and is capable of establishing a wireless communication link with the proposed system. After establishing a bidirectional communication, the platform begins the process of collecting the sensory data. However, such task starts only if the *execution permission* flag is enabled. Otherwise, no action is taken. If so, the sensorial data is captured, the current reference for the navigation is gotten and the errors in terms of the attitude of the aircraft are calculated. Based on such errors, the control signals necessary to the accomplishment of the programmed task are determined and, finally, are transmitted to the vehicle.

Considering a real UAV, the control signals are transmitted to the embedded board *AuRoRa Board*, which delivers the necessary commands to the actuators onboard the aircraft. After reacting to the control actions and interacting with the environment, the sensors onboard are excited and register the values of the new flight condition. These values are transmitted to the platform and the control cycle restarts. If the UAV is simulated, the control signals are sent to the dynamic model of the aircraft, which includes the model of the actuators and the rigid body model and, eventually, the uncertainties and disturbances added to the flight maneuvers. After actuating in the model, using numerical integration it determines the future position of the aircraft, updating the state variables and filling the data of the sensory part. At this time, such data are available for a new *execution permission* of the controller. Figure 14 shows this situation.

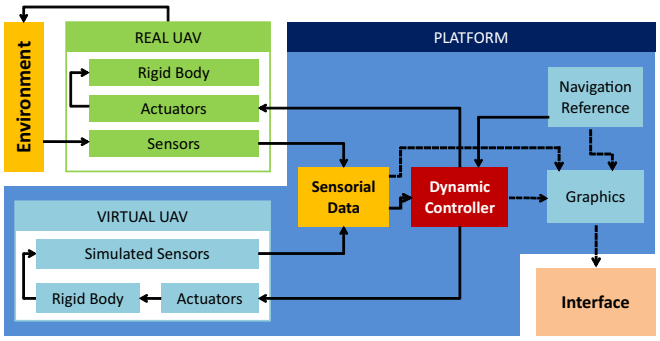


Fig. 14. Block diagram describing how the *AuRoRa Platform* works.

Algorithm 1: The structure of the proposed platform with multiple computers.

```
Initialization;
while t < t_max do
    if Execution permission then
        Read sensors;
        Read/Calculate the desired position;
        if There is a joystick then
            Read commands;
        else
            Apply controller;
        end
        Send control signals;
        Record variables;
    end
    if Graphic permission then
        if UDP transmission permission then
            Show graphics on client;
        else
            Show graphics on server;
        end
    end
end
```

In both cases, if the online flight data displaying is enabled a *graphic permission* flag is activated each and every predetermined time period (not necessarily the same period of the control cycle) and the task of illustrating the current state of the aircraft during its flight is performed. However, the online generation of the virtual environment represents a computational cost that may compromise the time-setting of the *execution permission*, and the sending of the control signals to the aircraft. Therefore, if the whole platform is running on the same machine, it is advisable to perform experiments without using the *graphic permission*. Figure 17 compare these two situations.

It is worth mentioning that along the navigation all data are stored, so that after finishing the experiment/simulation a log file correspondent to the flight is available. A possible use of such log file is to plot graphics to analyze the behavior of the aircraft during the mission after its accomplishment, for instance.

A. *Decentralized Version of the* AuRoRa Platform

The *AuRoRa Platform* has some disadvantages regarding the control execution time when displaying the graphics, as mentioned before. To illustrate online the movement of the aircraft/aircrafts through computer animation, the tasks of calculating and sending control

signals to the vehicles and receiving sensory information are compromised. After all, due to the sequential structure of the algorithm, the system must complete an action before starting the next one.

Referring to a simulation using a miniature helicopter and knowing that graphical display features has a high computational cost, Figure 17(a) shows the time values of the execution cycles (iterations), which vary around $100ms$ when graphical display is included, which clearly outweigh the sampling time of the aircraft. In this situation the UAV would not, for a period of a few cycles, receive a new control signal, which could even drive it to an unstable operation.

In order to not compromise the vehicle control involved in an experiment, one option is to choose not to display the data relative to the running task, as previously suggested. However, it is often necessary to observe the behavior of the aircraft and control signals, as well as a three-dimensional representation of the path navigated by the aircraft, to evaluate the performance of a controller, for instance. Hence, to distribute the display and the control activities between two (or more) interconnected computers becomes an interesting solution.

To solve this problem, the *AuRoRa Platform* allows the use of a ground station, where the actual experiment or simulation is performed and, via UDP protocol, to transmit the data to a second ground station, as shown in Figure 15, to distribute the computational effort. Notice that in such a figure the ground station A is responsible for the stabilization and navigation of the aircraft, while the ground station B only displays its current state.

In order to compare the two versions and therefore justify the implementation of a multi-machine system, Figure 17 illustrates the iteration time spent using the platform in its centralized and decentralized versions. One can check that in the decentralized version the control routine, sending and receiving data are performed in a time lower than $t_{rx} = \frac{1}{45}s$, correspondent to the sampling time of the miniature helicopters T-REX 450 and T-REX 600, which does not occur in the centralized version.

It is important to mention that, for the tests run so far, the data transmission packets for graphical displaying in a second machine using UDP communication was

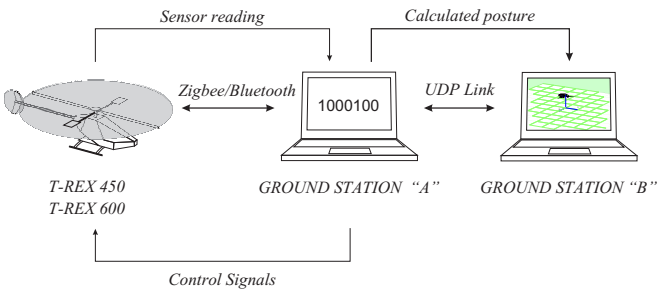


Fig. 15. Structure of the *AuRoRa Platform* with task distribution.

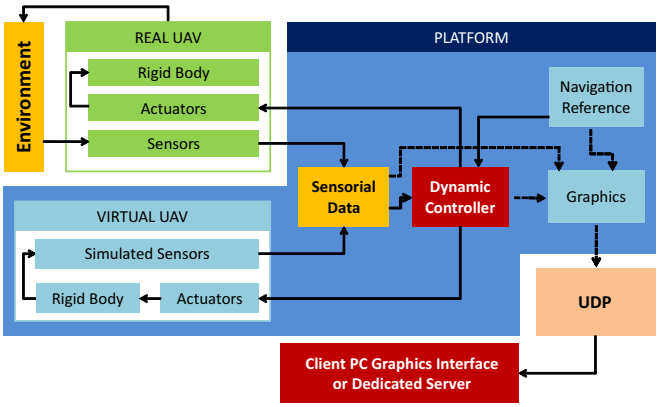


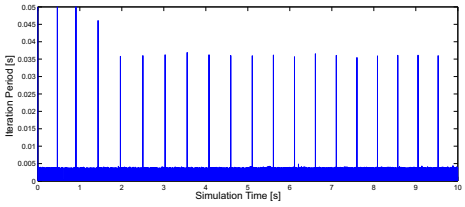
Fig. 16. Block diagram of the *AuRoRa Platform*.

performed over a time period lower than $1ms$, thus not compromising the sampling time of the vehicles. A specific protocol was created to pack and unpack such data. In case of any packet loss, the system operation is not affected, because the client station simply displays the current state of navigation through the data entry of the UDP communication channel.

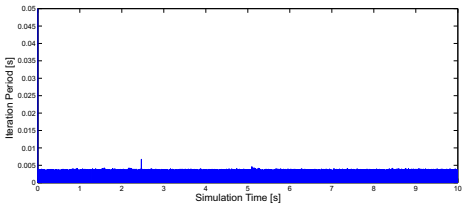
B. *Multi-Vehicle Version of the* AuRoRa Platform

Regarding the tests of models and controllers, it is interesting the capability to simulate several vehicles at the same time, in order to allow comparing their performances, for instance. Including, this analysis can be performed along an experiment, using the graphical interface included in the *AuRoRa Platform*.

It is worth emphasizing the fact that the same graphical interface is presented for the real experiments. Thus, the 3D projection of the vehicle is shown according to the received sensor data. The simulation can be performed with the amount of vehicles that the computer being used supports without compromising the sample period. Considering the idea of UDP communication and



(a) Centralized control.



(b) Decentralized control.

Fig. 17. Duration of the iterations of the *AuRoRa Platform*.

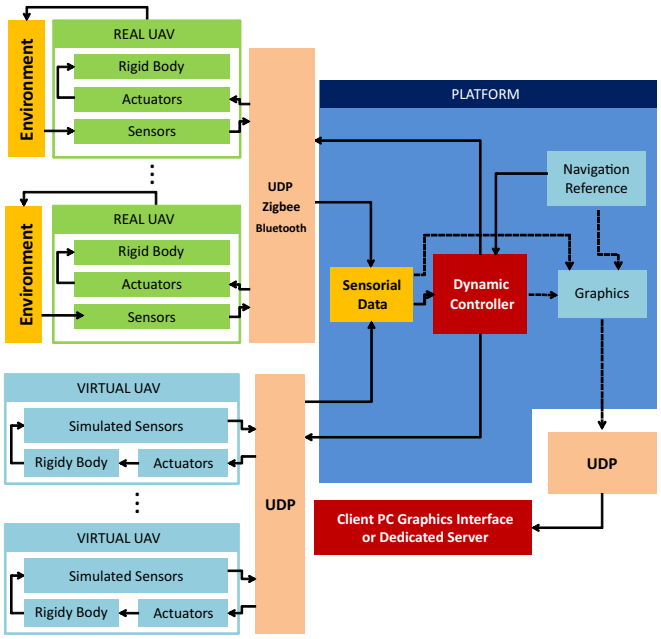


Fig. 18. Block diagram of the version of the *AuRoRa Platform* developed to deal with multiple vehicles.

decentralization of the *AuRoRa Platform*, it can not only distribute the graphics generation, but also distribute the vehicle simulation, i.e., the central computer will only perform the calculations for the controller and send data to other computers, which will be responsible for generating the dynamics of the UAV and deliver their sensory data, respecting the sampling period of each vehicle. The resulting structure is shown in Figure 18.

Notice that one can also control multiple real vehicles using UDP protocol, if the helicopter has an embedded computer. If it is not provided with such equipment, because of low load capacity or because the user chose it, the communication between the *AuRoRa Board* and the *AuRoRa Platform* can be implemented via Bluetooth or Zigbee modules.

V. Examples Using the Proposed Platform

In this section some examples of simulated and real flights run supported by the *AuRoRa Platform* and the *AuRoRa Board* are presented, which validate the proposed system as a useful tool to manage such kind of application.

A. Three Dimensional Flight Simulation

In this subsection a simulation of a helicopter flying in 3D space, i.e., exciting all its degrees of freedom, is presented. In the graphs shown, the dashed line represents the desired values, while the continuous line indicates the values actually obtained from the vehicle sensors (for an experiment) or by numerical integration (in the case of a simulation).

First, the vehicle will take off vertically, and then will track an 8-shaped trajectory in an inclined plane, defined by $\{x_d = 2\sin(\frac{3}{40}t), y_d = 2\sin(\frac{6}{40}t), z_d = \frac{3}{2} + \frac{1}{2}\sin(\frac{3}{40}t)\}$, with the reference of orientation given by $\psi_d = 0, \forall\, t$.

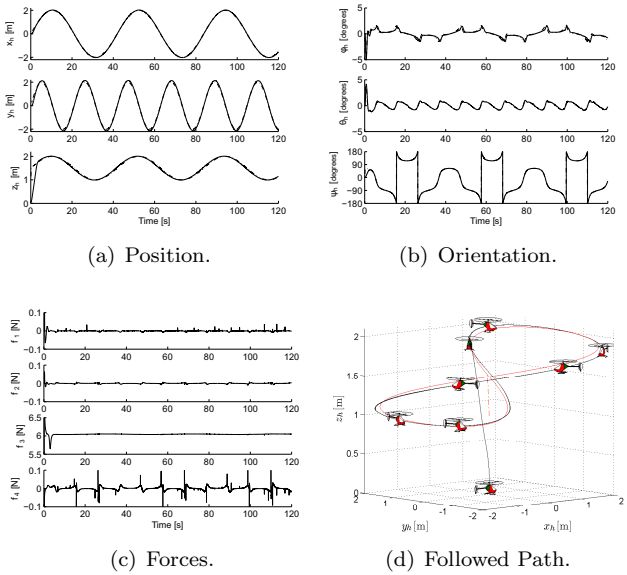
In this simulation a Gaussian noise with zero mean and standard deviation equal to $0.5N$ is introduced, along all the simulation, as well as a polarized constant disorder of intensity $[0.15, -0.05, 0.02]^T N$, acting just in the time interval $20s < t < 40s$ along the simulation.

The result of this simulation is shown in Figure 19. Notice that before starting the movement described above, the miniature helicopter performs a takeoff maneuver and then perform the above movements. In Figure 19(a) the evolution of the displacement variables is presented. The pitch and roll angles, necessary to achieve the displacements in the horizontal plane, are shown in Figure 19(b), as well as the temporal evolution of the yaw angle. Notice that the vehicle has its orientation tangential to the path, i.e., $\psi_d = 0$. Figure 19(c) highlights the forces that are applied to the miniature helicopter along the simulation. Such forces are f_1, associated to pitch movement, f_2, associated to roll movement, f_3, associated to the aircraft elevation, and f_4, associated to the yaw movement. A 3-D view of the desired and performed trajectories is shown in Figure 19(d), to allow checking the task evolution.

As one can see from this example, the flight controller being tested is robust enough to compensate for the disturbances introduced in the simulation, besides guiding the aircraft in the task accomplishment.

B. Multiple Vehicles Simulation

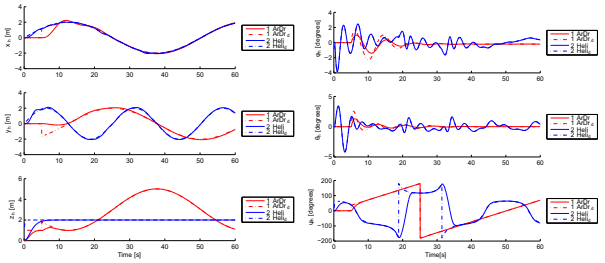
In order to test the capability of the proposed platform deal with multiple vehicles, this subsection presents a simulation involving two UAVs, a quadrotor and a helicopter, represented by red and blue lines, respectively, in the graphs of Figure 20. The first UAV tracks a circular



(a) Position. (b) Orientation.

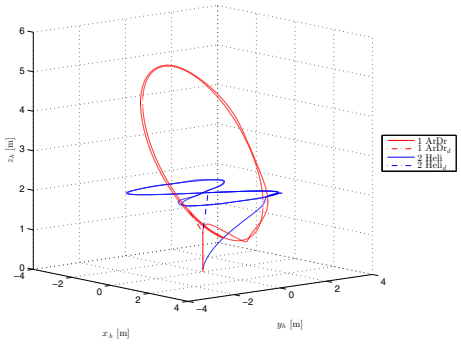
(c) Forces. (d) Followed Path.

Fig. 19. Simulation: Tracking an 8-shaped trajectory without movement restriction.



(a) Temporal evolution of the position variables.

(b) Temporal evolution of the attitude of the aircrafts.



(c) The 3D trajectory tracked by the UAVs.

Fig. 20. Simulation of trajectory tracking with two UAVs.

path in an inclined plane, whereas the second one tracks an 8-shaped trajectory in an horizontal plane, with the vertical reference fixed in $2m$ (the dotted lines represent the desired trajectories). As one can see, both vehicles accomplish their flight missions without oscillations or delays. The behavior of the aircrafts can be checked through the navigated path (Figure 20(c)) as well as by analyzing the evolution of the variables of position and attitude (Figures 20(a) and 20(b)).

C. Flight Data Log

The *AuRoRa Platform* is also capable of reading all flight data from the Radio Control. Thus, it is possible to make an analysis of the operating points of the vehicle. This is important to know the behavior of the aircraft with the applied inputs. Thus, it is possible to regulate the operation levels of the controller for a smoother or more abrupt response, increasing or decreasing the operation area, from the flight data analysis. Figure 21 shows the data captured during a manually controlled flight with the T-REX 450 helicopter. Analyzing the green curve, which represents the speed of the main rotor, one can see that it grows along with the collective channel, as expected in a radio controlled operation. Notice that the speed grows faster than the collective, and stabilizes at 70%. This is commonly done in radio control, where the acceleration chart is usually limited at a value defined by the user (in this case 70%). This limit is important because for each value of speed there is a different blade attack angle, which will cause the vehicle to sustain itself in the air and perform their movements in different values.

When high speed maneuvers are necessary, one should choose a higher threshold value. Otherwise, if just smooth movements should be performed, a lower value should be chosen, high enough for guaranteeing the lift of the aircraft (this is why we have chosen the value of 70%). Remember that in previous sections the velocity of the main rotor and tail rotor were considered constant: actually, regardless of how high or low this limit is, it should be constant from the time the helicopter takes off on. The lower this limit is, the less aggressive the maneuvers will be, compared to those performed at high speeds.

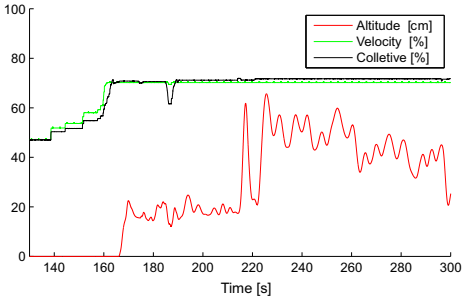


Fig. 21. Data readings coming from the radio control of the T-REX 450 miniature helicopter.

D. Automatic Height Control

Figure 22 presents the results of a real autonomous flight in which only the altitude is being controlled. The rotorcraft used in such an experiment is the TREX-600 helicopter. The objective is to keep the miniature helicopter in a fixed height, for a time interval. The curve in the figure shows that although oscillating about the reference value, the miniature helicopter could follow the established profile. Notice that in this experiment a low height was selected, which is harmful, because at low altitudes the vehicle suffers of ground effect, an aerodynamic effect in which the air-flow generated by the main rotor blades is "reflected"by the ground, increasing thrust on the vehicle in this region, making it more susceptible to abrupt movements [24].

As previously mentioned, for security reasons the Radio Control is responsible for enabling or not the commands from the automatic controller. Thus, after starting the experiment, the operator should choose a

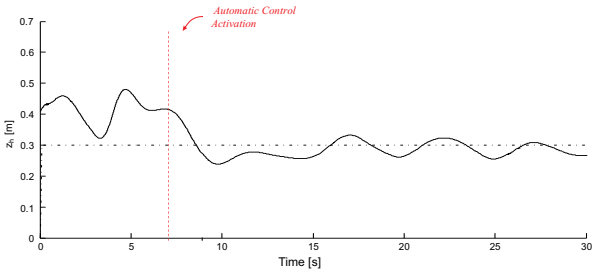


Fig. 22. Automatic height control.

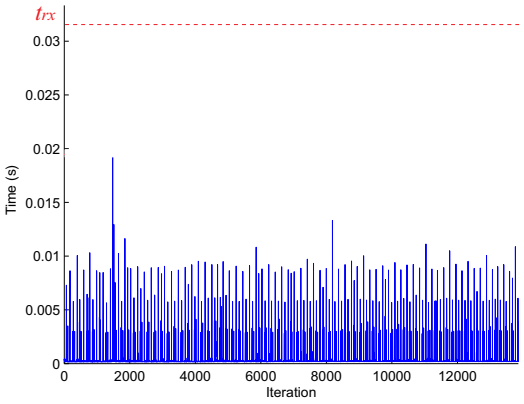


Fig. 23. Autonomous height control.

time instant to activate the autonomous controller. If necessary, he/she can disable the autonomous control, and the aircraft will be back to manual control. The time instant in which this operation mode is enabled or disabled are also shown in the figure, in connection with the experiment (see Figure 22).

Regarding the time interval between iterations, it is shown in Figure 23 that all the periods were smaller than the sampling time, assuring the efficiency of the controller being tested.

VI. Concluding Remarks

This work addresses the development of a platform to support autonomous or remotely controlled (radio control) flights of helicopters and similar aircrafts, contemplating both hardware and software aspects. The electronic board called *AuRoRa Board* was designed to read all the data coming through the radio controller and interpret them, and to actuate or not to actuate on the servomotors of the aircraft as well (according to such signals). Such a board is also able to receive control signals synthesized by high-level navigation controllers, running on an computer embedded in the vehicle or on a ground computer. Then, according to a decision made by the user, the signals to be sent to the servomotors are selected, between those coming through the radio control and from the autonomous controller. Both, the reading of the signals and the actuation on the servomotors, are accomplished with a suitable resolution.

In addition to the designed board, it was also developed a software structure, to operate in connection with it, thus implementing the *AuRoRa Platform*, which allows driving the servomotors of the aircraft through signals generated by controllers running on a ground or embedded computer, transmitted via Bluetooth, Zigbee, a wireless network or a UDP link. This communication also enables the platform to receive the flight data (aircraft sensor readings) and store them for further analysis. It can also display such data online, in the same computer that generates the signals or in another one, connected to the first one via UDP protocol. In such a case, two ground stations are adopted.

After running a series of experiments and simulations regarding distinct missions and distinct rotary-wing aircrafts, the platform here described has proven to be suitable for use as a supporting tool for real and simulated flights with autonomous rotary-wing aircrafts, as well as to allow simulating multiple vehicles simultaneously, or even running real experiments with multiple vehicles, even of different types. This way, such platform is suitable to be a support for systems dealing with the coordination of groups of rotary-wing aircrafts or even groups involving aerial and ground vehicles. To adapt the platform to such cases and to run simulations and experiments with coordinated navigation of multiple vehicles is the next step of this research.

Acknowledgment

The authors thank CNPq – a Brazilian agency that supports scientific and technological development, for financing this project (grant 473185/2012-1). Dr. Sarcinelli Filho also thanks the additional financial support of FAPES - Fundação de Amparo à Pesquisa do Espírito Santo to the project. They also thank Federal Institute of Espírito Santo, Federal University of Viçosa and Federal University of Espírito Santo, respectively, for supporting their participation in this research. Dr. Brandão thanks FAPEMIG - Fundação de Amparo à Pesquisa de Minas Gerais - for supporting his participation in this work.

References

[1] C.-T. Lee and C.-C. Tsai, "Improved nonlinear trajectory tracking using rbfnn for a robotic helicopter," *International Journal of Robust and Nonlinear Control*, vol. 20, pp. 1079–1096, October 2010.

[2] A. Dzul, R. Lozano, and P. Castillo, "Adaptive altitude control for a small helicopter in a vertical flying stand," *International Journal of Adaptive Control and Signal Processing*, vol. 18, no. 5, pp. 473–485, Jun 2004.

[3] Y. Bestaoui and R. Slim, "Maneuvers for a quad- rotor autonomous helicopter," in *AIAA Conference and Exhibit*, Rohnert Park, California, May 7-10 2007.

[4] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small unmanned aerial vehicles: A survey," *International Journal of Control, Automation, and Systems*, 2010.

[5] F. Mutter, S. Gareis, B. Schatz, A. Bayha, F. Gruneis, M. Kanis, and D. Koss, "Model-driven in-the-loop validation simulation-based testing of uav software using virtual environments," in *Proceedings of the 18th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, April 2011, pp. 269–275.

[6] C. sun Yoo, Y. shin Kang, and B. jin Park, "Hardware-in-the-loop simulation test for actuator control system of smart uav," in *2010 IEEE International Conference on Control Automation and Systems (ICCAS)*, Oct. 2010, pp. 1729 –1732.

[7] G. Cai, B. Chen, T. Lee, and M. Dong, "Design and implementation of a hardware-in-the-loop simulation system for small-scale uav helicopters," in *IEEE International Conference on Automation and Logistics (ICAL)*, sept. 2008, pp. 29 –34.

[8] A. Goktogan, E. Nettleton, M. Ridley, and S. Sukkarieh, "Real time multi-uav simulator," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, sept. 2003, pp. 2720 – 2726 vol.2.

[9] C. Bonivento, M. Cacciari, A. Paoli, and M. Sartini, "Rapid prototyping of automated manufacturing systems by software-in-the-loop simulation," in *Chinese Control and Decision Conference (CCDC)*, may 2011, pp. 3968 –3973.

[10] S. Demers, P. Gopalakrishnan, and L. Kant, "A generic solution to software-in-the-loop," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, oct. 2007, pp. 1 –6.

[11] M. Frye, S. Bhandari, and R. Colgren, "The raptor 50 6-dof simulation environment for flight control research," in *Proceedings of the American Control Conference, 2006*, june 2006, p. 6 pp.

[12] A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "A nonlinear underactuated controller for 3d-trajectory tracking with a miniature helicopter," in *2010 IEEE International Conference on Industrial Technology (ICIT)*, March 2010, pp. 1421–1426.

[13] A. S. Brandão, J. A. Sarapura, E. M. C. de Oliveira, M. Sarcinelli-Fillho, and R. Carelli, "Decentralized control of a formation involving a miniature helicopter and a team of ground robots based on artificial vision," in *Proceedings of the 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting - LARS2010*. São Bernardo do Campo/SP, Brasil: IEEE, October 2010, pp. 126–131.

[14] A. S. Brandão, V. H. Andaluz, M. Sarcinelli-Filho, and R. Carelli, "3-d path-following with a miniature helicopter using a high-level nonlinear underactuated controller," in *Proccedings of the 9th IEEE International Conference on Control and Automation - ICCA'11*, Santiago, Chile, December, 19–21 2011, pp. 434–439.

[15] A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "High-level underactuated nonlinear control for rotorcraft machines," in *Proceedings of the 2013 IEEE International Conference on Mechatronics - ICM2013*, Vicenza, Italy, February-March 2013, pp. 279–285.

[16] K. Kondak, C. Deeg, G. Hommel, M. Musial, and V. RemuB, "Mechanical model and control of an autonomous small size helicopter with a stiff main rotor," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Sendai, Japan, September 28 – Octuber 2 2004.

[17] A. Budiyono and S. S. Wibowo, "Optimal tracking controller design for a small scale helicopter," *Journal of Bionic Engineering*, vol. 4, pp. 271–280, 2007.

[18] E. D. Beckmann and G. A. Borges, "Nonlinear modeling, identification and control for a simulated miniature helicopter," in *Proceedings of the Latin American Robotic Symposium*. Los Alamitos, CA, USA: IEEE Computer Society, 2008, pp. 53–58.

[19] P. Castillo, R. Lozano, and A. Dzul, *Modelling and Control of Mini-Flying Machines*. USA: Springer, 2005.

[20] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear $\mathscr{H}_\infty$ control structure for a quadrotor helicopter," *Automatica*, vol. 46, pp. 29–39, 2010.

[21] T. J. Koo and S. Sastry, "Output tracking control design of a helicopter model based on approximate linearization," in *Proceedings of the 37th Conference on Decision and Control*, Tampa, Florida USA, 1998, pp. 3635–3640.

[22] R. Pettersen, E. Mustafic, and M. Fogh, "Nonlinear control approach to helicopter autonomy," Master's thesis, Institute of Electronic Systems, Department of Control Engineering of the Aalborg University, 2005.

[23] B. Ahmed, H. R. Pota, and M. Garratt, "Flight control of a rotary wing uav using backstepping," *International Journal of Robust and Nonlinear Control*, vol. 20, pp. 639–658, January 2010.

[24] T. Roy, M. Garratt, H. R. Pota, and M. Samal, "Robust altitude control for a small helicopter by considering the ground effect compensation," in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, 2012, pp. 1796–1800.