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

Platform for Quadrirotors: Analysis and applications

Helosman Figueiredo[1], Adriano Bittar[2] and Osamu Saotome[3]

Abstract— **This paper describes the effort of modeling and simulating the quadrirotor ITA-001 aircraft developed by a quadrirotors study group of the Instituto Tecnolgico de Aeronutica (-ITA). We developed this simulation platform to support research and training of new researchers in the area of unmanned aerial vehicles, especially quadrirotor. This paper describes the implementation of a Software-In-the-Loop (SIL), which will be used by the study group for the testing platform, using the commercial flight simulator X-Plane, the dynamics of aircraft and the environment, and the software Matlab/Simulink, which simulates where the control laws are implemented. The main result of this work was the development of a tool to aid the research and development of new aircraft of the type quadrirotor type and evaluate the use of flight simulator X-Plane as a simulation tool, for this purpose a comparison was made between the responses of X-Plane and the mathematical model based on equations of motion of the quadrirotor. Another result was an example of application of this platform for laboratory classes, using two aircraft and laws guiding.**

I. INTRODUCTION

Currently the area of unmanned aerial vehicles (UAVs) has been gaining interest. Advances in data processing system of low consumption systems, miniaturization of sensors, use of MEMS devices, engine optimization, and development of control theory especially for autonomous vehicles has facilitated the development of reduced-size aircraft.

UAVs have attracted the attention of military and civilian community, because of the range of applications that can be employed, for example, surveillance, logistics, rescue operations, and inspections of hostile environments [1]. Together with their evolution, UAVs have also become more complex. In general, the aircraft can be divided into two broad categories: heavier than air and lighter than air vehicles [2]. Fig. 1 shows the main groups of UAVs and highlights the position of quadrirotors.

A. Quadrirotors

Quadrirotors is earning prominence among UAVs because they have unique characteristics of flight such as hovering, flying at low speed, simplified mechanics, as well as vertical take-off and landing. All these features allow the quadrirotor

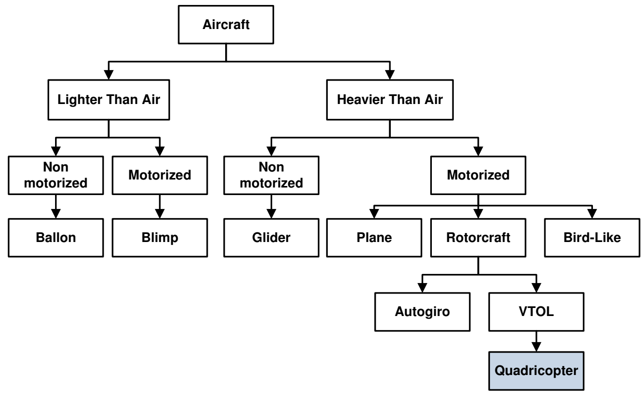

Fig. 1: UAVs groups.

to perform tasks that are not possible with other aircraft. Quadrirotor is an aircraft composed of four independent rotors disposed in + or X, thus making its mechanics and construction simple. This paper addresses a specific quadrirotor ITA-001 shown in Fig. 2 developed by the quadrirotor study group at ITA.

Fig. 2: Quadrirotor ITA-001.

Simplified mechanics facilitating miniaturization of the quadrirotor, along with its good maneuverability makes it an ideal aircraft to conduct missions in confined spaces [3]. The main research groups dealing with subjects involving quadrirotors looked at topics, such as cooperative work between other vehicles [4], interactive learning, and aggressive maneuvers [5].

B. Simulation Platforms

The growth of interest in UAVs has also generated growth in research and development of simulators for these vehicles. Simulation is an important phase in the design of aircraft, because it is used to predict the characteristics of the system,

*This work was supported by Instituto Tecnólogico de Aeronáutica - ITA
[1]Helosman Figueiredo at the Instituto Tecnólogico de Aeronáutica - ITA, Division of Mechanical Engineering, Pça. Mal. Eduardo Gomes, 50 Vila das Acácias, CEP: 12.228-900, São José dos Campos-SP, Brasil. helosman@gmail.com
[2]Adriano Bittar with the University of Denver, Unmanned Systems Research Institute Denver, CO, USA adrianobittar@gmail.com
[3]Osamu Saotome at the Instituto Tecnólogico de Aeronáutica - ITA, Division of Electrical Engineering, Pça. Mal. Eduardo Gomes, 50 Vila das Acácias, CEP: 12.228-900, São José dos Campos-SP, Brasil. osaotome@gmail.com

and is a good way to research interaction with users or even investigate certification, licensing and accidents [6].

The development of new aircraft concepts for UAVs, such as ducted fan aircraft, octorotors, quadrirotors, the simulation activities have become essential to study and development of new technologies. Due to this, demand for simulation, several tools have been developed and adapted for this task, such as robotic simulators: Gazebo and Microsoft Robotic Studio, and flight simulators: Flight Gear, Microsoft Flight Simulator, and X-Plane.

Traditionally, flight simulators are utilized in phases of development, certification, operator training, and for entertainment. Current simulators integrate accurate dynamic models with highly realistic 3D graphics. Using identical interfaces for the cockpit of the aircraft, simulators allow the user to experiment safely with various flight scenarios [7].

Most researchers choose open source or commercial flight simulators, such as Flight Gear, X-Plane, and Microsoft Flight Simulator, as interfaces for graphic display. Simulators are also used in conjunction with numerical computation software MATLAB. This software is generally used to test control algorithms and navigation; however, some studies use this software to simulate the aircraft dynamics.

Commercial and open source simulators are widely used because they provide integrated modeling software that allows one to edit or create aircraft and the simulation environment. They also have some aircraft models already available for use, facilitating the development of experiments. Using this software, studies may focus on different areas of modeling and simulation, such as the design and evaluation of control laws.

A good example of open source simulators is Flight Gear simulator. This simulator can run run several instance, for example an instance responsible for Graphical 3D and 2D map by another simulation. A good example of this is the Atlas that provides the 2D map positions of an aircraft. Another good tool is the integrated map editor, which provides a way to generate map data, including the placement of objects in 3D and use landscape photos. These characteristics are exploited in the following works: [8], [9], [10], and [11].

For commercial simulators a good example is X-Plane simulator, which has been used by the world's leading defense contractors, air forces, aircraft manufacturers, and space agencies for applications ranging from flight training to new aircraft design concepts . Its use has grown because of the way the simulator generates the dynamic model of the aircraft. In addition, it can be run at high frame rate and is certified with the Federal Aviation Administration (FAA) to be used in pilot training [12].

Another advantage of using the X-Plane simulator is the Plane Maker module. This module can be used to create or edit aircraft,which allows the modeling and testing of diverse concepts of an aircraft. Other strong point of this simulator is the method of simulation called "*Blade Elements*" that calculates the interaction with the environment, speed, and attitude of the aircraft [6].

Some researchers prefer to develop their own simulation environment, for example, the work reported in [7], where they used an environment developed based on a Java platform, using the sub-models in Fortran to simulate the dynamic of aircraft propulsion systems and atmospheric turbulence. The rendering engine was developed based on libraries in C/C++. The problem with this kind of simulator is the time spent in preparation of the simulation whereas X-Plane is ready to use.

The high level of realism of simulators is an essential tool for developing UAVs, facilitating the understanding of the flight mechanics of these aircraft, and identification and correction of design errors before the first flight. For this purpose, three dimensional simulators have been used.

To confirm the choice of X-Plane, a table was created to compare the existing simulation tools. Among the points assessed include: preparation for flight, configurations, documentation, and certification. Table I shows the comparison.

TABLE I: Simulators Comparison.

Characteristics	Flight Gear	X-Plane	Robotic	Custom
Ready to fly	Yes	Yes	No	No
Configurations	Easy	Easy	Easy	Hard
Documentation	Good	Good	Good	Good
Certification	No	Yes	No	No

The Robotic group represents leading 3D robotic simulators in the market, such as Gazebo, Microsoft Robotic Simulator, and Morse. The Custom simulators use resources of 3D interfaces for computer games such as the SimplySim.

This work describes a simulation platform for quadrirotors using the flight simulator X-Plane 9 and the Matlab Simulink software for modeling and controlling the aircraft. This work includes a comparison between the model used in X-Plane software and a mathematical model developed in Matlab, and an example of using this platform as an educational tool.

C. *Structure of Work*

This paper is organized as follows. The concept of simulation platform is in Section II. The quadrirotor ITA-001 motion equantions are in Section III. Section IV presents the implementation process of software-in-the-loop. The results and an example of utilization are presented in Section V. The conclusions and Future works are in Section VI.

II. Simulation Platform for quadrirotor ITA-001

A. *The Proposal Platform*

This work proposes a simulation platform consisting of two computers connected by an Ethernet network, using UDP protocol. The tools used in this work were the flight simulator X-Plane 9 for simulation and Matlab/Simulink for control. This concept is illustrated in Fig. 3.

One PC computer runs flight simulator X-Plane that is responsible for simulating the quadrirotor motion dynamics and 3D visualization of simulation. The other PC with Matlab is responsible for running the autopilot and storage

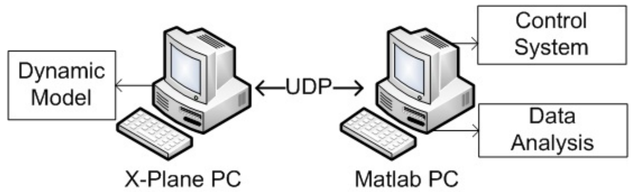

Fig. 3: Simulation Platform Concept.

as well as for logging and analysis of simulation data. The simulation loop is shown in Fig. 4.

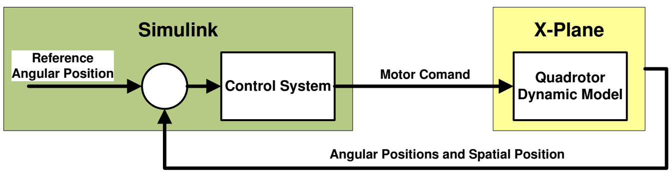

Fig. 4: Simulation loop.

The Matlab PC generates the error signal from the reference signal and data feedback provided by X-Plane. Then it uses the error signal as input of the control law, thus generating a command signal for the engines that will be sent to X-Plane. Next, the X-Plane PC receives data for engine commands, performs the interactions, and sends the new data positions to the Matlab PC.

III. Quadrirotors

The concept of quadrirotors occurred in 1907 with the Breguet brothers and Professor Richet. Their concept was a large and heavy aircraft, and it was not able to make flights with large payload for large distances [13].

Quadrirotors are robust and simple because they have no *swashplates* and complex linkages found in conventional helicopters. Most four-rotor aerial robots are built from remote control aircraft components, as a result, these robots have good reliability and performance needed to serve as experimental platforms [14] and it is easy to find spare replacement parts. Fig. 5 illustrates a conventional quadrirotor frame with rotors arranged in a cross shape.

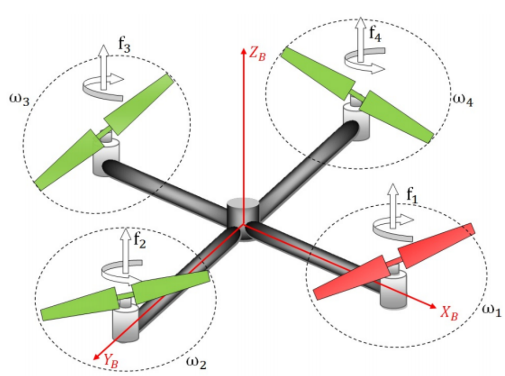

Fig. 5: Configuration of conventional quadrirotor.

In this type of aircraft, the movements of translation and rotation are performed by varying the speed of the rotors. Fig. 6 show combinations of rotor speed for the main movements, for a plus shape quadrirotor.

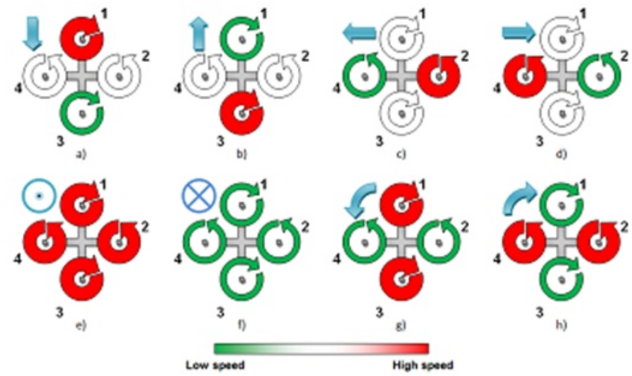

Fig. 6: Movements of a quadrirotor [15].

A. Flight dynamics

A quadrirotor's flight dynamics can be represented by a system in which the outputs are the angular and spatial positions, and the voltage of the four motors are inputs. The mechanical equations, which regulate the movement in six degrees of freedom, are presented below. Fig. 5 represents the positioning of axes and nomenclature of engines, used in the creation of the mathematical model. The modeling was based on the work of [16] and [17].

The control torques can be expressed by (1), (2), and (3), where Ω_i is the speed of the motor, b is the coefficient of impulse, l is the distance between the quadrirotor geometric center and the center of propeller, and d is the coefficient of drag.

$$\tau_x = bl\left(\Omega_4^2 - \Omega_2^2\right) \tag{1}$$

$$\tau_y = bl\left(\Omega_4^2 - \Omega_1^2\right) \tag{2}$$

$$\tau_z = d\left(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2\right) \tag{3}$$

By having four rotors spinning in different axes, the gyroscopic effect adds resistible torques to the movements of the X and Y axes. These torques generated by gyroscopic effect can be expressed by (4) and (5).

$$\tau_x = I_{rotor}\omega_y\left(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4\right) \tag{4}$$

$$\tau_y = I_{rotor}\omega_x\left(-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4\right) \tag{5}$$

Where I_{rotor} is the moment of inertia of the rotor, ω_x and ω_y represent the angular velocities in X and Y respectively, and Ω_i represent the rotation speed of the engine 1, 2, 3, and 4.

Adding aerodynamic torques, the gyroscopic effect of the rotor and the effects of inertia, one can describe the motion of quadrirotor using (6), (7), and (8) presented below, where $\ddot{\Phi}$, $\ddot{\Theta}$, and $\ddot{\Psi}$ are the angular accelerations:

$$\ddot{\Phi} = \dot{\Theta}\dot{\Psi}\frac{(I_y - I_z)}{I_x} + bl\frac{(\Omega_4^2 - \Omega_3^2)}{I_x} + \dot{\Theta}I_{rotor}\frac{(\Omega_1 - \Omega_3 + \Omega_2 - \Omega_4)}{I_x} \quad (6)$$

$$\ddot{\Theta} = \dot{\Psi}\dot{\Phi}\frac{(I_z - I_x)}{I_y} + bl\frac{(\Omega_2^2 - \Omega_1^2)}{I_y} + \dot{\Phi}I_{rotor}\frac{(\Omega_1 - \Omega_3 + \Omega_2 - \Omega_4)}{I_y} \quad (7)$$

$$\ddot{\Psi} = \dot{\Phi}\dot{\Theta}\frac{(I_x - I_y)}{I_z} + d\frac{(-\Omega_1^2 + \Omega_3^2 - \Omega_2^2 + \Omega_4^2)}{I_z} \quad (8)$$

The spatial accelerations are proportional to the forces generated by the engine. The forces of the motors can be represented by $T_i = b\omega_i^2$ and the sum of the forces of the motors can be expressed by $\vec{F} = \sum_{i=1}^{4} T_i$. Accelerations in three-dimensional space are represented by (9), (10), and (11).

$$\ddot{x} = \frac{(\cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi)}{m}\sum_{i=1}^{4} T_i \quad (9)$$

$$\ddot{y} = \frac{(\sin\Psi\sin\Theta\cos\Phi - \cos\Psi\sin\Phi)}{m}\sum_{i=1}^{4} T_i \quad (10)$$

$$\ddot{z} = -g + \frac{(\cos\theta\cos\Phi)}{m}\sum_{i=1}^{4} T_i \quad (11)$$

B. Quadrirotor ITA-001

The quadrirotor ITA-001 was based on an experimental structure, assembled from parts of commercial RC (*Remote Controlled*) model airplanes. The structure is made from aluminum rods and a set of acrylic parts. It has four brushless motors placed at the ends of stems, four motor brushless drivers, and a controller for stabilization and receiving commands. Table II are characteristics of quadrirotor ITA-001.

TABLE II: Parameter list.

Parameters	Value
Half wingspan	0.32m
Payload	0.8Kg
Mass	1.2Kg
Flight Time	15min

IV. Implementation

To implement this platform we used the work of [18], [19], [6], and also other internal manuals and reports prepared by students of the ITA. These studies use X-Plane and Matlab simulation of unmanned aerial vehicles (airplanes, helicopters and ductedfans).

To implement the concept shown in Fig. 3, the work was divided into three stages: modeling, communication and development of system control, and data analysis in Matlab/Simulink.

A. Plane-Maker Modeling

Plane-Maker is a program that comes with X-Plane, which allows users to create their own planes. Using this software, any imaginable aircraft can be built. Once all the physical specifications of the plane were inserted (eg, weight, wingspan, deflection control, engine power, airfoil sections, etc.), the X-Plane simulator will predict how the plane flies in the real world [20].

This section is very important because the X-Plane uses the simulation method called *Blade Elements*. With this method, X-Plane assimilates the geometry of the aircraft and simulates the flight dynamics. The flight simulator separates aircraft into small sections, calculates the velocity vector of each element, and then the vectors are added together, and the resulting velocity vector is generated for the aircraft. Dividing by the mass forces of the aircraft, linear accelerations and moments of inertia are used for the angular accelerations. These interactions occur about 240 times per second.

The model in Plane-Maker was created with the following steps:

1) Sketching on paper.
2) Create fuselage in Plane-Maker.
3) Create secondary objects, such as landing gear and engine nacelles.
4) Configure the systems and internal properties, such as engines, electrical systems, weight and balance.
5) Configure additional features of the aircraft, such as weapons or special controls.
6) Create the instrument panel.
7) Perform flight test for fine tuning.
8) Add textures, 3-D objects, paintings, and extras.

Following the recommendations of the manual and in possession of the real aircraft, the modeling process was successful. Fig. 7 presents the modeling phase of the aircraft and Figs. 8 and 9 illustrate the first flight, for fine tuning of parameters.

B. Interaction between X-Plane and Matlab

The native method of communication (import and export data) in X-Plane is the UDP protocol. This protocol is an unwarranted protocol, in other words, it does not guarantee that the packets arrive, so it may present a potential problem resulting from data corruption [19]. However this problem did not occur, because of short transmisions distances in a dedicated network. This simulator has the capacity to export various data types, such as various forces acting on the

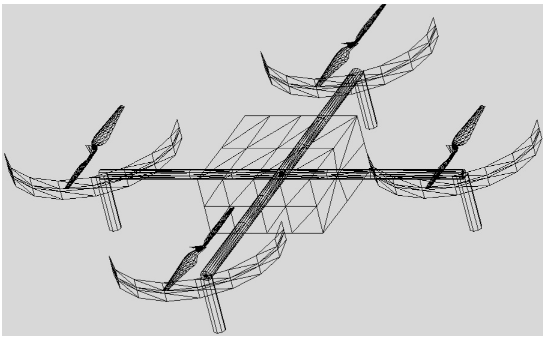

Fig. 7: ITA-001 Plane Maker Model.

Fig. 8: ITA-001 Cockpit.

aircraft. The path also allows the insertion of failures during flight.

The main differences between the TCP and UDP protocols is that TCP has mechanisms to initiate, maintain and terminate the communication, detect and correct errors, and allows the re-transmission of corrupted packets; UDP does not have complex control structures. The UDP data packet is basically composed of data, which makes packing and unpacking of data faster, thus providing a quicker time response for the quadrirotor.

To establish communication between the software's Simulink and X-Plane requires the use an Ethernet network with IP addresses defined for each computer. Then, both software must be configured for sending packets and X-Plane must be configured for sending data. Communication ports and what data should be sent in Simulink must be configured in the IP simulator. The port sending and the port that should receive the packet must be configured to send the shipping address and port. The works [18], [21], [22] and [23] describe in detail the communication between X-Plane and Matlab.

C. Control System

The quadrirotor is a naturally unstable system, making it very difficult to control manually. Diverse control techniques have been used such as PID controllers, PD, Fuzzy, and neural networks. The main focus of this work is to develop an

Fig. 9: X-Plane Flight.

educational platform and major efforts have been dedicated to modeling and simulation. We did not perform detailed studies on the control system: instead we opted to use PI-D loops for the attitude and altitude, with gains adjusted empirically.

A PI-D controller was added in an attitude and altitude loop. The basic architecture of the controller implemented is shown in Fig. 10.

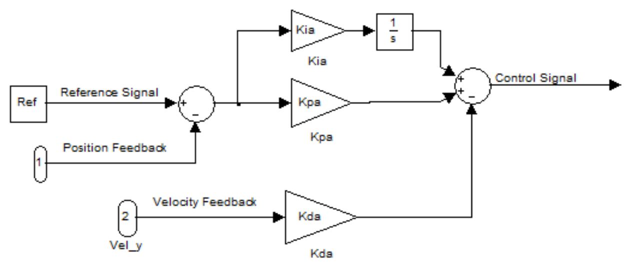

Fig. 10: PI-D Architecture.

D. Transformation of Coordinate System in X-Plane System for North-East-Down (NED)

Coordinate systems: the coordinate systems are used to refer to a particular geographical body or vehicle. Thus, to set the position of an object, one must specify the coordinate system in which the object is inserted. The coordinate systems used in inertial navigation are three mutually orthogonal axes [24].

A system commonly used in aircraft systems is the system NED (North-East-Down), also known as local navigation system. Its origin can be established at any point of the globe. Its X axis points to true north, its Z axis points to the center of the Earth, and its Y axis is referenced according to the right-hand rule [24]. A framework of this kind enables easy and intuitive reading of coordinates in a neighborhood of the origin due to the local concept of north, east and below. It can be seen in Fig. 11.

The flight simulator X-Plane uses a different coordinate system for navigation, which is commonly used in the

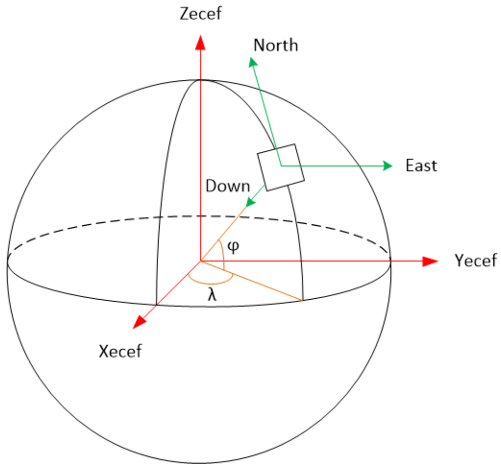

Fig. 11: NED coordinate system.

aeronautical systems. The simulator generates a virtual environment tangent to the circumference of the earth, with the coordinate axes arranged as follows: X-axis eastward, Y-axis up, Z-axis to the South.

To make the transformation from the coordinate system of the flight simulator X-Plane to the NED coordinate systems, took two rotations: the first rotation was performed on the Y-axis in a counterclockwise direction, the second rotation was performed on the X-axis clockwise. The sequences of rotations are illustrated in Fig. 12.

This sequence of rotations shown above may be described as (12) presented below:

$$R_{Sim-NED} = R_x * R_y =$$
$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -\cos\phi & -\sin\phi \\ 0 & \sin\phi & -\cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$
$$= \begin{bmatrix} -\cos\theta & 0 & \sin\theta \\ -\sin\phi\sin\theta & -\cos\phi & -\sin\phi\cos\theta \\ -\cos\phi\sin\theta & \sin\phi & -\cos\phi\cos\theta \end{bmatrix} \quad (12)$$

V. Results

This work obtained results in various areas such as: modeling, design, control systems, education. This work provides a basis for several other future works. This platform can be used to research areas such as hardware-in-the-loop, software-in-the-loop, and interaction between multiple aircraft.

The main result of the work is the simulation platform for quadrirotor ITA-001, presented in the previous sections. Another important result was the creation of the mathematical model and the model for the flight simulator. This platform can be used as a design tool for aircraft being used to test the concepts of flight mechanics even before building a prototype. It can also be used to assist in the development of control laws and navigation of this type of aircraft.

Another important result obtained by this work was the educational of putting together a 3D graphic interface and models of refined flights in X-Plane flight simulator platform with Matlab/Simulink, wich is used widely in academia as one of the main design tools for control systems, enabled the Instituto Tecnolgico de Aeronutica to facilitate the training of researchers in this area. This facility can be seen when new students were able to make the first contact with the aeronautical environment, without it being necessary to understand all the concepts. This platform has become a tool for consolidation of knowledge and realization of new discoveries for these students.

Sections V-B present a laboratory activity in which students could use this platform as a learning tool. This example is used in order to elucidate this educational outcome.

A. Analysis of X-Plane Simulations

To analyze the quality of simulation in X-Plane, a mathematical model was fitted to (6), (7), (8), (9), (10), and (11). Then using the same control law with the same reference signs, we compared the responses.

The conditions were chosen in order to compare takeoff and hover to 1m altitude. The response graphs are shown in Fig. 13 and 14.

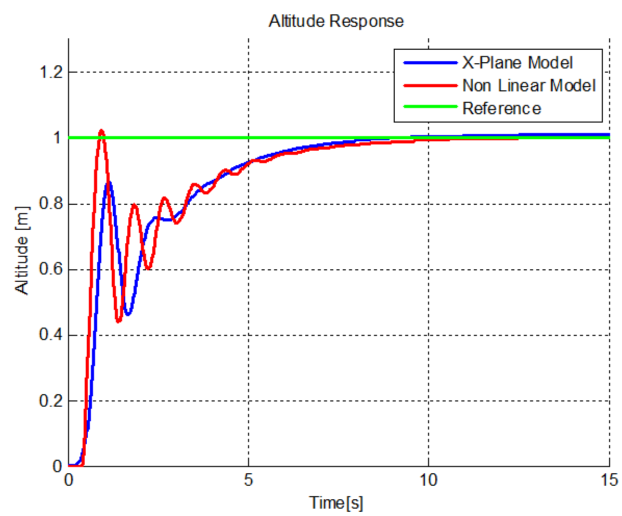

Fig. 14: Altitude response for comparison of the models (Takeoff and hovering on 1 meter).

Both models showed similar responses. The nonlinear model implemented in Simulink showed more oscillatory response, because this takes no account of the atmosphere in which the quadrirotor entered as show in Fig. 13. The aircraft in X-Plane does not start with the angles totally zeroed causing an error to follow the trajectory. In the non-linear model, perturbations have not been added so the answer of this model is well below the reference signal. However, a comparison with the real quadrirotor is needed to confirm the results produced by this work.

B. Application Example

In order to provide example of the use of the simulation platform developed to simulation quadrirotor ITA-001. The application discusses interaction between aircraft and guidance algorithms.

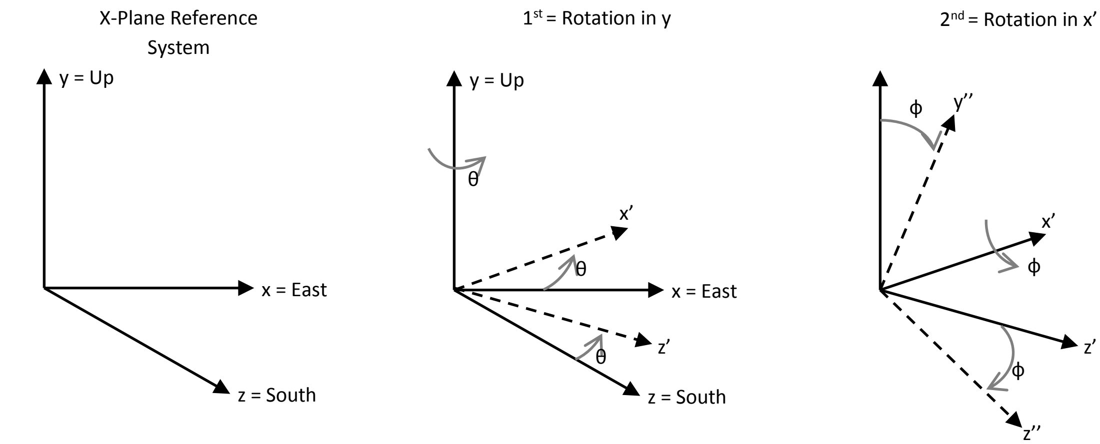

Fig. 12: Sequence of rotations.

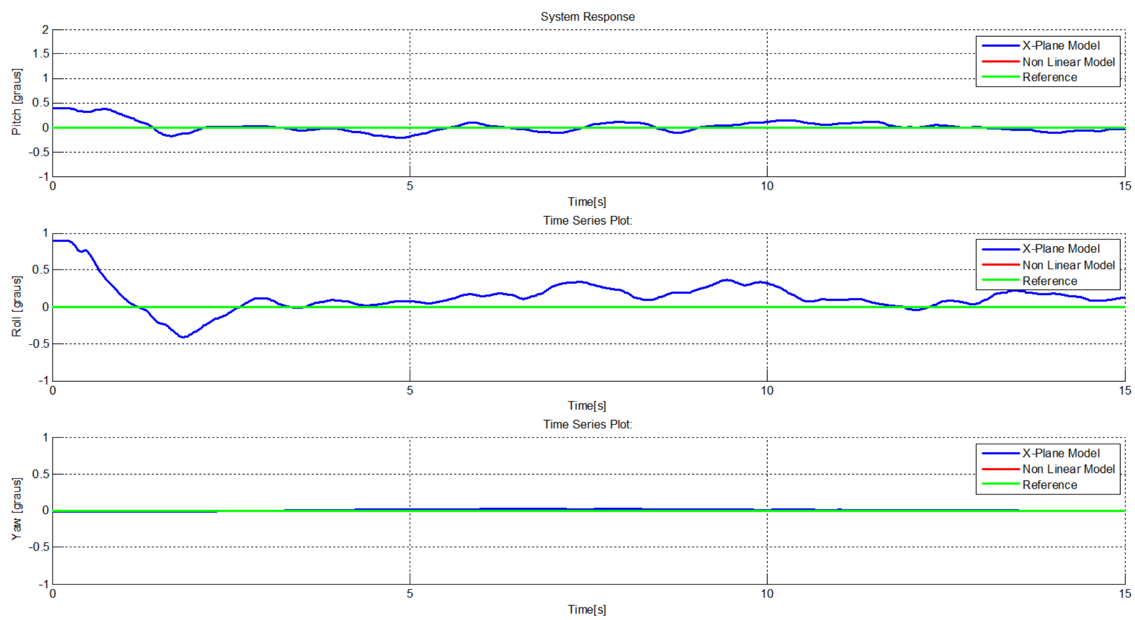

Fig. 13: Attitude response for comparison of the models (Takeoff and hovering on 1 meter height). The reference signal overlaps the signal of the Non-linear model, in the responses of pitch, roll and yaw.

This application example consists of a manually controlled target aircraft (Quadrirotor 1) and an aircraft pursuer (Quadrirotor 2), which made the follow through a Line of Sight (LOS) guidance law. The guidance logic used in this work is through waypoints using the angle LOS. The desired LOS angle can be calculated by (13).

$$\Psi = \tan^{-1}\left[\frac{y_d(k) - y(t)}{x_d(k) - x(t)}\right] \quad (13)$$

In this work, the waypoint will be the target vehicle controlled manually with a USB joystick. Fig. 15 presents the moment of pursuit.

Finally, the view of the path traveled by the aircraft in NED system is presented for a journey undertaken with the aid of the Joystick. Fig. 16 shows the route of the aircraft in 3D, and Fig. 17 presents a 2D graphic.

With this application example, it is clear that many areas of

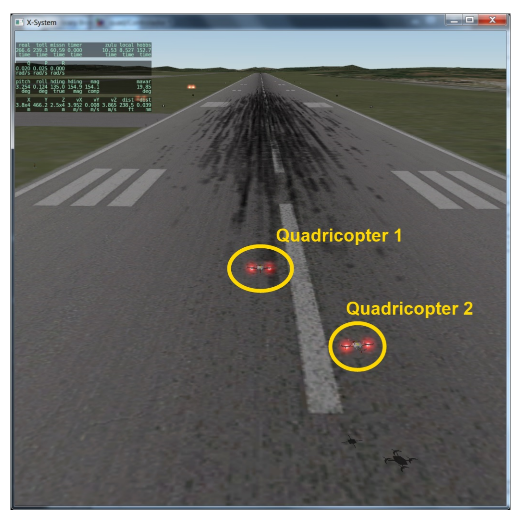

Fig. 15: Persecution in X-Plane.

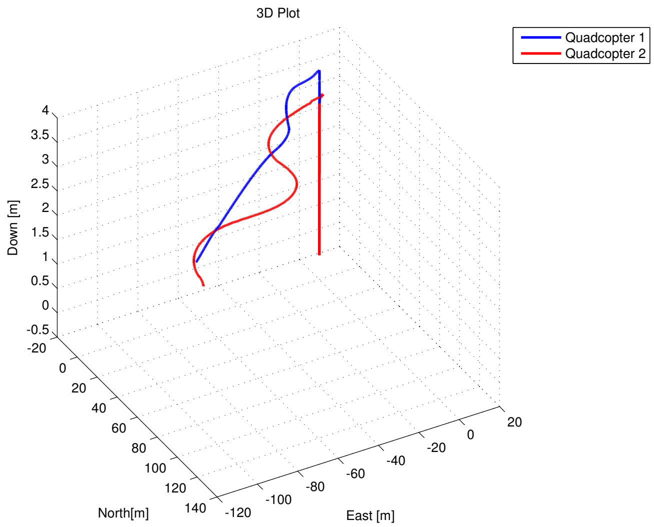

Fig. 16: Coordinates the simulation of persecution 3D Plot.

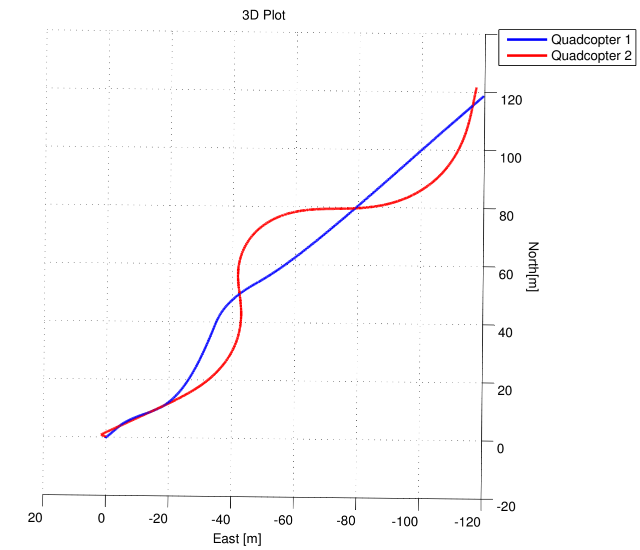

Fig. 17: Coordinates the simulation of persecution (North X East).

knowledge can be explored with this platform. In instruction, the teacher could use this platform as a test bench for students to evaluate their control laws and guidance, as has been done in ITA.

VI. Conclusion and Future Work

This test platform is a great tool for the study and design of automatic control systems for UAV type quadrirotor. It is also possible to evaluate the response of the aircraft in various flight conditions.

The flight simulator X-Plane along with the tool Plane Maker is shown to be a sophisticated tool that can be used to model various types of aircraft, including quadrirotor ITA-001.

This environment allows simulation of the quadrirotor with a high degree of realism. This capability provides some benefits such as: reducing the risks associated with the experimental flights, reducing design time and easing the adjustment of the autopilot. In addition, with a 3D visualization of the aircraft during the simulation, the movements of the aircraft are easily understood.

This simulation platform opens up numerous possibilities for applications, enabling also the testing of several techniques for control and navigation algorithms, as well as testing with software-in-the-loop and hardware-in-the-loop. This architecture also allows the simulation of a fleet of aircraft that can interact with each other and with other vehicles.

As a continuation of this work, a flight with real aircraft becomes necessary to compare the results. Another activity is to create experiments in the laboratory, exploring areas that a quadrirotor could be used for rescue, surveillance, transportation or interactions with other aircraft.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

References

[1] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 29 2007-nov. 2 2007, pp. 153 –158.

[2] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, Lausanne Polytechnic University, Lausanne, 2007. [Online]. Available: http://biblion.epfl.ch/EPFL/theses/2007/3727/EPFL/TH3727.pdf

[3] B. Nourghassemi, "Development of the control algorithms for autonomous landing of unmanned aerial vehicles," 2009.

[4] A. Benini, A. Mancini, R. Minutolo, S. Longhi, and M. Montanari, "A Modular Framework for Fast Prototyping of Cooperative Unmanned Aerial Vehicle," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 507–520, Aug. 2011. [Online]. Available: http://link.springer.com/10.1007/s10846-011-9577-1

[5] G. Hoffmann, S. Waslander, and C. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," *2009 IEEE International Conference on Robotics and Automation*, pp. 3277–3282, May 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152561

[6] T. Indriyanto, Y. Jenie, and Ibrahim, "Modeling and simulation of a ducted fan unmanned aerial vehicle (uav) using x-plane simulation software," 2010.

[7] E. Capello, G. Guglieri, and F. B. Quagliotti, "UAVs and simulation: an experience on MAVs," *Aircraft Engineering and Aerospace Technology: An International Journal*, vol. 81, no. 1, pp. 38–50, 2009.

[8] J. Kim, A. Budiyono, D.-M. Kim, H.-G. Song, and D.-H. Kim, "A TMO-based flight program of an unmanned helicopter," *Aircraft Engineering and Aerospace Technology*, vol. 83, no. 6, pp. 353–362, 2011.

[9] B. Cervin, C. Mills, and B. Wünsche, "A 3d interface for an unmanned aerial vehicle," *SE Stage*, vol. 4, 2004.

[10] B. R. Trilaksono, R. Triadhitama, W. Adiprawita, A. Wibowo, and A. Sreenatha, "Hardware-in-the-loop simulation for visual target tracking of octorotor UAV," *Aircraft Engineering and Aerospace Technology*, vol. 83, no. 6, pp. 407–419, 2011. [Online]. Available: http://www.emeraldinsight.com/10.1108/00022661111173289

[11] Atlas, "About atlas," 2013, [Online; accessed 06-September-2013]. [Online]. Available: http://atlas.sourceforge.net/

[12] L. Research, "Introduction," 2014. [Online]. Available: http://www.x-plane.com/desktop

[13] J. G. Leishman, "A History of Helicopter Flight," 2000. [Online]. Available: http://terpconnect.umd.edu/ leishman/Aero/history.html

[14] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a quad-rotor robot," in *Proceedings Australasian Conference on Robotics and Automation 2006*, no. November, 2006, pp. 27–29. [Online]. Available: http://www.araa.asn.au/acra/acra2002/Papers/Pounds-Mahony-Hynes-Roberts.pdf

[15] J. Domingues, "Quadrotor prototype," MASTER THESIS, Universidade Técnica de Lisboa, 2009.
[16] P. Beugnet, R. Polonowski, and K. A. N. Mohamed, "Modélisation et interface de contrôle d'un quadri-rotor," in *Projet I4 SE*, Paris, 2008, p. 72.
[17] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," MASTER THESIS, Lund University, 2008.
[18] L. R. Ribeiro, "PLAFORMA DE TESTES PARA SISTEMAS DE PILOTO AUTOMÁTICO UTILIZANDO MATLAB/SIMULINK E SIMULADOR DE VÔO X-PLANE," Mestre em Ciências, Instituto Tecnológico de Aeronáutica, 2011.
[19] R. Garcia and L. Barnes, "Multi-UAV Simulator Utilizing X-Plane," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 393–406, 2009.
[20] L. Research, *Plane Maker Manual*, 2011. [Online]. Available: http://www.x-plane.com/files/manuals/Plane_Maker_manual.pdf
[21] A. Bittar, *Piloto Automático para VANT 's TUTORIAL Comunicação X-Plane Simulink*, 2011. [Online]. Available: https://pilotoautomaticovant.wikispaces.com/Tutorial+X-Plane+%26+Simulink
[22] F. H. V. and O. Saotome, "Simulation Platform for Quadricopter: Using Matlab/Simulink and X-Plane," *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, pp. 51–55, Oct. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6363318
[23] A. Bittar, N. Oliveira, and H. Figueiredo, "Hardware-in-the-loop simulation with x-plane of attitude control of a suav exploring atmospheric conditions," *J Intell Robot Syst*, pp. 51–55, Oct.
[24] S. Mohinder and P. A. Lawrence R. W.and Andrews, *Global Positioning Systems, Inertial Navigation and Integration.* John Wiley and Sons, 2001.