Simulation Platform for Quadricopter: Using Matlab/Simulink and X-Plane

Helosman V. Figueiredo and Osamu Saotome
Instituto de Tecnológico de Aeronáutica, Praça Mal. Eduardo Gomes, 50 - Vila das Acácias - São José dos Campos - SP – Brasil

Abstract — **The unmanned aerial vehicles (UAVs) has grown in military and civilian areas of application. Several industries (automotive, military, factories, space, etc.) use robots for dangerous and repetitive tasks. This paper is dedicated to a special type of aerial platforms, the quadricopter. UAVs that has been highlighted by having flight characteristics and construction only, for example, hovering flight, vertical takeoff and landing, high maneuverability, low speed flight, and simple mechanics. This work proposes a tool for simulation and visualization of an aerial air-type quadricopter robots, using the flight simulator X-Plane 9, Matlab and Simulink. The aircraft under study is the so-called ITA-001, developed by the quadricopter study group the Aeronautical Institute of Technology – ITA.**

Keywords — **Quadricopter, Simulation, Matlab, X-Plane.**

I. Introduction

Currently, the use of unmanned aerial vehicles (UAVs) has increased both in the military and civil areas, especially in missions where human operation is unnecessary, repetitive or dangerous. These are employed in diverse areas such as surveillance, photography, traffic monitoring, identification of pests in agriculture and educational platforms [1].

UAVs can be classified into two groups: fixed-wing and rotorcraft. Fixed wing flight vehicles are suitable for outdoor use and can cover large area. The rotary-wing vehicles have features such as: increased maneuverability, vertical takeoff and landing, and hovering flight, enabling flight at low altitudes and indoor application [2]. For the group of rotorcrafts, a configuration that is gaining prominence is the quadricopter, which consists of four independent propellers arranged in a cross shape.

The concept of quadricopter appears in 1907 with Breguet brothers and professor Richet work [3]. Their aircraft concept was large and heavy, and it was not possible to make a flight with large payloads or large distance flights. Since then, rotorcraft research had been developed, but the concept of quadricopter was forgotten. In recent years, the development of lightweight building materials, the improvement of the relation between power and weight, and the miniaturization of the engine and control systems, have been factors that turned the quadricopter feasible. Since then, several civil and military institutions are researching and developing this type of aircraft [4].

The autopilot is a major component of a UAV. The design of automatic pilot systems requires many simulations and tests on real aircraft. The real aircraft tests are high risk ones and accident prones. Therefore, before being embedded, autopilot systems must be thoroughly tested in the laboratory [5].

Several platforms have been developed for UAV simulation, but they are focused on complex vehicles and integration of their system. Great effort and development time are required to build a custom configuration for specific integration of sensors and normally complex software is required to operate. Many of them do not support modifications of hardware or software, and to update or to modify it, one spends almost the same time necessary to develop a new UAV. Besides the simulation platform related problems, it is also difficult to simulate the actual meteorological conditions along the route [6].

Some authors has been addressed the topic of simulation platform for UAVs. The reference [6] treats the problems of small fixed wing UAVs, analyzing the control and navigation algorithms. The reference [7] treats the problem of the ducted type fan UAVs, and proposes a concept of simulation platform based on the commercial flight simulator X-Plane 9 and Matlab \ Simulink. The reference [8] is a simulation platform for multi-UAV helicopters, and proposes a simulator for test and evaluation of control algorithms for stabilization of the aircraft and formation of the swarm. As our knowledge, none of the recent works presents a simulation platform for quadricopter control and navigation and prepares a suitable environment for quadricopter swarm research, at the point of view of its embedded system.

For the reasons mentioned above and for not having found a platform to quadricopters simulations, with support for autopilot tests, ability to customize features of the aircraft and to simulate real conditions (such as winds, turbulences, etc.), we propose in this article a platform for simulation and visualization of quadricopter type aircraft using the commercial simulator X-Plane 9 and Matlab Simulink. The main objective of this platform is to test and to help the development process of autopilot systems for quadricopter type aircraft. The quadricopter ITA-001 will be used as an object under test of this study.

This article is organized as follows: Section II describes the quadricopter ITA-001, Section III presents the simulation platform, Section IV describes the data interface for the simulation, Section V describes the implementation platform, Section VII is about platform application and Section VIII contains conclusions and suggestions for future works.

II. Quadricopter ITA-001

The quadricopter is an aircraft with four independent rotors, fixed at the end of each axle as shown in Fig. 1 [4]. In order to cancel the moment generated by the rotating blades, a pair of propellers rotates clockwise and the second pair in a counterclockwise direction.

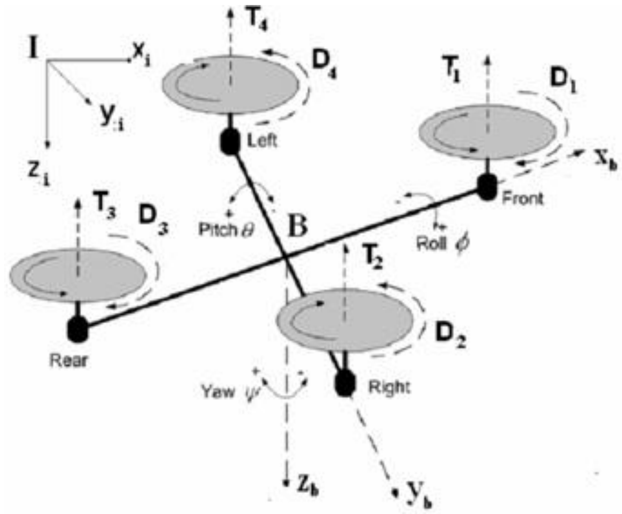

Fig. 1. Quadricopter Configuration [4]

A. ITA-001 Description

The model ITA-001 quadricopter was based on an experimental structure, assembled from parts of commercial RC (Remote Controlled) airplanes, as shown in Fig. 2. Its structure is made of aluminum rods, together with pieces of acrylic. It contains four brushless motors located on the ends of the rods, four engine controllers and a controller for stabilization.

Fig. 2. ITA-001

B. ITA-001 Control System

The quadricopter ITA-001 features two control problems: glide control and path control. The first control system is used to obtain a stable hovering operation. The second control system is to perform missions that require to travel certain paths. Fig. 3 shows the control system for a quadricopter.

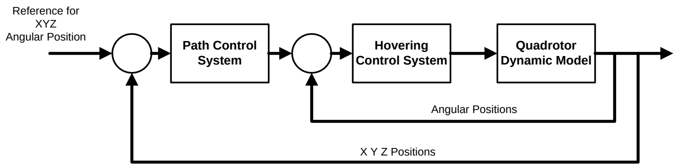

Fig. 3. Control System

III. Simulation Platform

This platform has been implemented using the flight simulator X-Plane 9 and Matlab. The chosen means of communication between software is UDP (user datagram protocol). The Matlab/Simulink runs the autopilot and processes data flight for analysis. The X-Plane simulates the dynamics of the aircraft and displays the 3D view of the Simulation. The Simulation Platform concept is presented in Fig. 4.

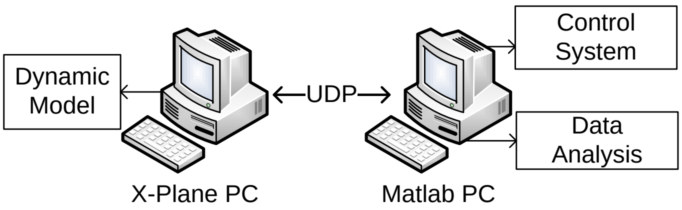

Fig. 4. Simulation Platform Concept

A. X-Plane flight simulator

This simulator was chosen for the following characteristics:

- Certification by Federal Aviation Administration (FAA).
- Plane Maker: tool for drawing and modifying aircraft.
- Has been used as the basis for designing and testing control algorithms.
- Has a good interface for data injection and data extraction.
- Accurate flight models.

The X-Plane uses the geometric shape modeled in Plane Maker to see how the aircraft will fly. It uses an engineering method called blade element theory, which means to divide the aircraft into small elements and then find the forces acting on each element several times per second [9].

X-Plane native method of communication (import and export data) is the UDP protocol. This protocol is a non-guaranteed protocol and gives no assurance that data packets arrived in order or didn't arrive at all, so it may present a possible problem resulting from data corruption. [8]

This simulator has configurations where it is possible to view the various forces acting on the aircraft, the path traveled by the aircraft and, also, allows the introduction of in-flight failures.

B. Matlab/Simulink

Matlab/Simulink® is an environment for multi-domain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing [10].

Simulink is integrated with MATLAB®, providing immediate access to an extensive range of tools that let you develop algorithms, analyze and visualize simulations, create batch processing scripts, customize the modeling environment, and define signal, parameter, and test data [10].

In the simulation platform, the software Matlab / Simulink is responsible for processing the UDP data, control system for glide and post processing of data simulation.

IV. Data Interface

The UDP protocol is chosen because it is the standard system for sending and receiving data from the X-plane and, also, it is compatible with Matlab / Simulink.

The speed of communication is a point of great importance, because the control commands must be synchronized with the simulation in X-plane. The X-Plane has capacity to send 99.9 packets per second, but most of the inertial measurement systems operate at a rate of 40 to 60 times per second. Therefore, we chose to work with 40 packets per second.

The type and size of the UDP packet depends on the amount and types of data to be exported. The X-Plane has a interface to set the output and receiving data, which is shown in Fig. 5.

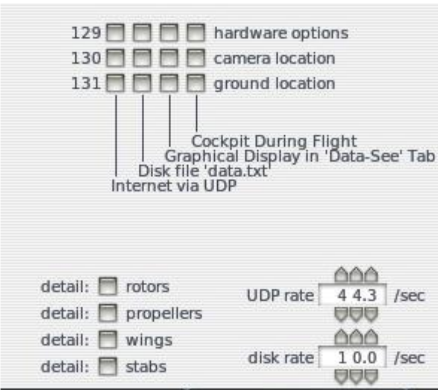

Fig. 5. UDP Configuration X-Plane9

The data packet sent by the X-plane follows this pattern: the first four bytes represent the type of packet, the fifth byte is an internal policy, the next four bytes indicate what type of parameter is being sent, the following four bytes represent the value in single-precision float. An example of packet data is shown in Fig. 6.

	DATA				I	ROTULO				Dado 1				...			
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
						Speeds				Vind(Kias)				...			

Fig. 6. Data Packet

In this work the data sent from the X-plane for the Matlab / Simulink are:

- Angular Position ;
- Scalar Position;
- Angular Velocity;
- Scalar Velocity.

V. Platform Implementation

This platform has been implemented using two desktop computers. The first computer runs the flight simulator X-Plane and the second one runs control system and processing, and also analysis of simulation data. The computers communicate each other through Ethernet port via UDP protocol.

Flight simulator X-Plane package contains the simulated environment, some aircrafts models, such as airplanes Cessna 172, Piper Malibu, helicopters Bell 47, Bell 206, and many others, and a software environment to develop new airplane models. The quadricopter is not mentioned. In this work, we present the development of a X-plane quadricopter model, using the existing (but not documented) airplane and helicopter models.

For this implementation it had been necessary to build the quadricopter ITA-001 model for X-Plane 9, using the tool Plane-Maker. For the modeling process, we made use of the Plane-Maker Manual [11], and Tutorial Plane-Maker [12].

Making ingenious use of Plane-Maker special options, and also taking advantage of symmetry existing in the system, the modeling was performed successfully, as illustrated in Figs. 7 and 8. It is worth to note that the quadricopter is an UAV (Unmanned Aerial Vehicle), so it doesn’t have cockpit. But for simulation, the cockpit vision of Fig. 8 is very useful for pilot training purposes, because the front view facilitates control training of the aircraft and the instruments view helps in maneuvers training.

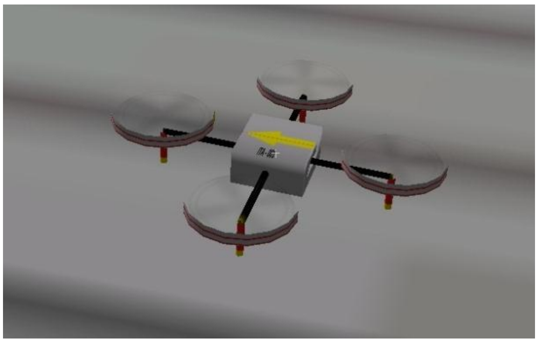

Fig. 7. ITA-001 X-Plane model

Fig. 8. ITA-001 Cockpit Vision

VI. X-PLANE SIMULATION ANALYSIS

To analyze the quality of the response generated by the flight simulator X-Plane, this analysis was performed following these steps: a mathematical model was fitted non-linear quadricopter with the same characteristics used for modeling the X-Plane. Then simulations were made with similar reference signals for the two models, using the same control system. The mathematical model was developed based on the work of Pierre Beugnet [13]. The equations governing the dynamics of the quadricopter, taken from [13] are presented below:

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\frac{(I_y-I_z)}{I_x} + bl\frac{(\Omega_4^2-\Omega_2^2)}{I_x} + \dot{\theta}I_{rotor}\frac{(\Omega_1-\Omega_2+\Omega_3-\Omega_4)}{I_x} \quad (1)$$

$$\ddot{\theta} = \dot{\psi}\dot{\phi}\frac{(I_z-I_x)}{I_y} + bl\frac{(\Omega_3^2-\Omega_1^2)}{I_y} + \dot{\phi}I_{rotor}\frac{(\Omega_1-\Omega_2+\Omega_3-\Omega_4)}{I_y} \quad (2)$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\frac{(I_x-I_y)}{I_z} + d\frac{(-\Omega_1^2+\Omega_2^2-\Omega_3^2+\Omega_4^2)}{I_z} \quad (3)$$

$$\ddot{x} = \frac{(\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi)}{m}\sum_{i=1}^{4} T_i \quad (4)$$

$$\ddot{y} = \frac{(\sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi)}{m}\sum_{i=1}^{4} T_i \quad (5)$$

$$\ddot{z} = -g + \frac{(\cos\theta\cos\phi)}{m}\sum_{i=1}^{4} T_i \quad (6)$$

Equations 1, 2 and 3 represent the angular accelerations and the 4, 5 and 6 the accelerations in spatial axes. Ix, Iy and Iz are the inertia on axes x, y, z, b is the thrust factor, l represents the quadricopter half scale, d is the drag coefficient and Ti represents the engine torque in the motors 1, 2, 3 and 4.

As described in Section V, there is no quadricopter model in X-Plane, so we had been constructed it, based on helicopter model. So it is important to verify if this model matches with the mathematical model given by equations (1) to (6) and represented in Matlab/Simulink. To perform this comparison, the two models were submitted to a closed-loop control to hover at the constant altitude of 4 meters. The result, presented in Fig. 9, shows that both control systems converge in an almost similar way.

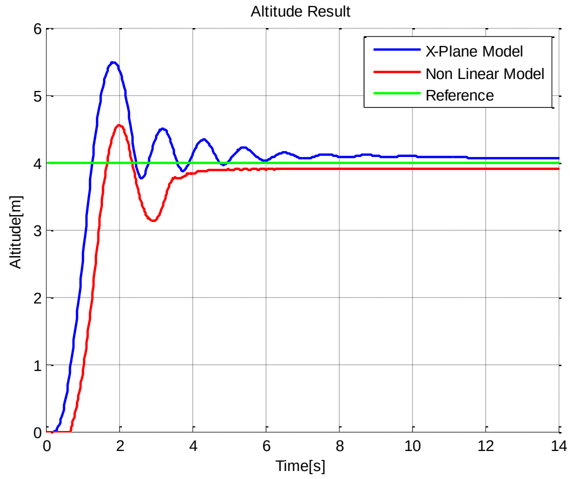

Fig. 9. Comparison of the models

The comparison between the mathematical model and the X-Plane model shows a less than 2% error after convergence, and the X-Plane model presents a bigger maximum peak due to initial conditions of the simulator. The control system is adjusted based on the mathematical model. For a better adequacy of the results, the control system must also be adjusted in the model of the X-Plane.

As a result of this comparison, we verify that the model developed in the flight simulator obeys the dynamics of a real quadricopter described in the literature.

VII. APPLICATION

A. Development tool for autopilot

To demonstrate the use of this simulator as a development tool for autopilot systems, a control system for hovering was designed and implemented in simulation platform. The high level hovering control system block diagram is shown in Fig. 10.

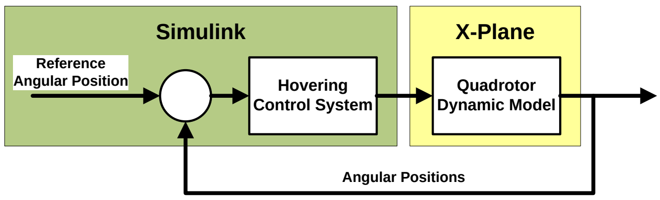

Fig. 10. High Level Hovering Control

The quadricopter dynamics is simulated at X-Plane, and the autopilot blocks are inserted in the block control system. Fig. 11 presents a detailed vision of the control system block.

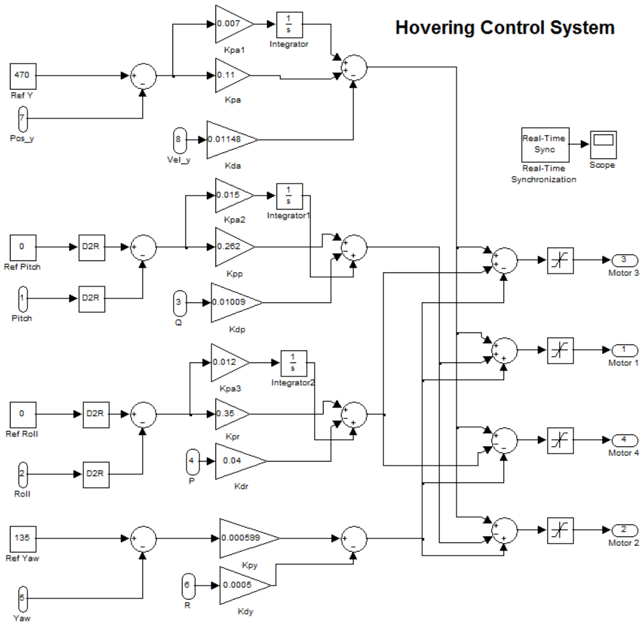

Fig. 11. Detailed view of the control system

The reference signal applied to the system and the responses to this stimulus are presented in Fig. 12. For hovering, the reference signals of the angular positions were set to zero, and the reference altitude is set to 476 meters. Altitude response is shown in Fig. 13.

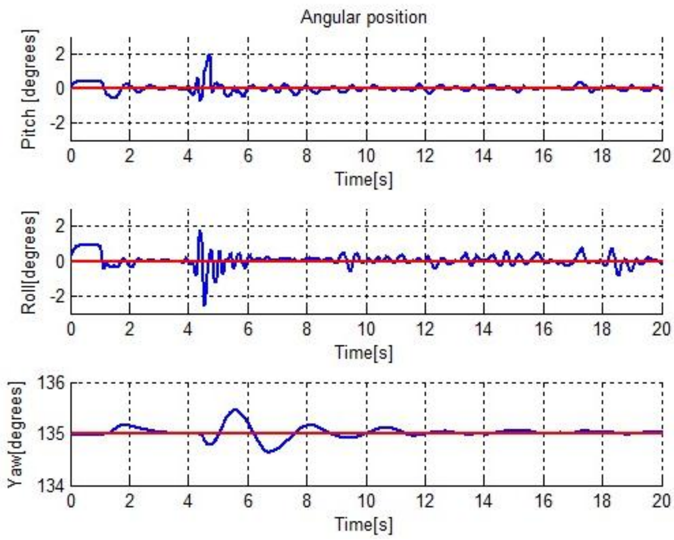

Fig. 12. Angular Positions response

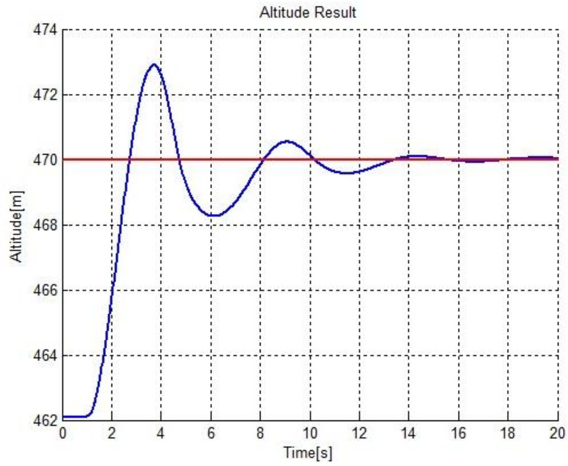

Fig. 13. Altitude Response

For this simulation, the quadricopter starts from the ground, and then establishes the reference signals to hover in 470 meters. The control loops gains were tuned empirically. In order to ease this work, we designed a graphical user interface shown in Fig. 14.

The Fig. 12 results show that pitch and roll values lower oscillation and approaches to convergence after 6 seconds, and yaw values take approximately 10 seconds to convergence. Furthermore, Fig. 13 shows altitude stabilization after 14 seconds.

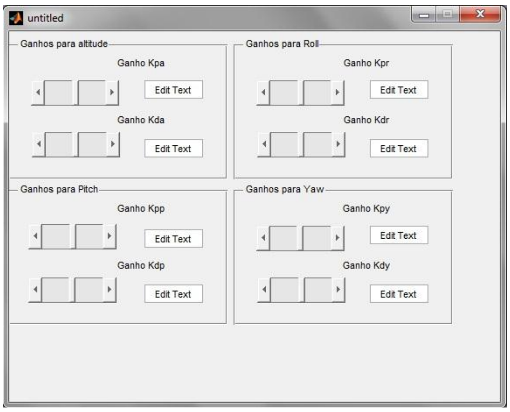

Fig. 14. Adjust interface

Better responses can be obtained, doing more simulations to make fine adjustments in the controller gains.

B. Design tool for embedded system

An example of the utilization of this platform as a design tool for an autopilot system in context of an embedded autonomous system, is the analysis of the control loop frequency. This analysis is important because for small aircrafts, such are the case of quadricopters, energy supply by battery and payload are very restricted, making critical the choice of sensors and microcontrollers that will be embedded. The control loop frequency can be limited by the speed of data acquisition for feedback or the processing capacity of the microcontroller where the control system was embedded.

As a design tool, we could use this platform and test the control system in various frequencies. As an example we use the hovering control system previously proposed, and the response of the system at various frequencies, generated by the computer. Fig. 15 presents the system response to altitude.

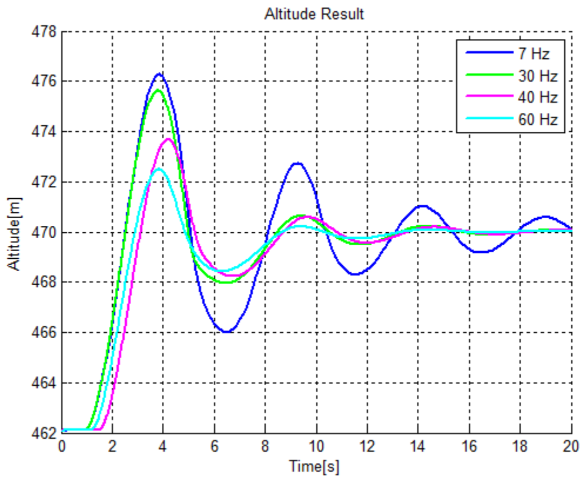

Fig. 15. Altitude Response

Analyzing the resultant curves of Fig. 15, we find that for low frequencies the system shows a larger overshoot and longer accommodation. Increasing the frequency of operation, we can observe that the maximum peak and the settling time are both lowering with the increase of frequency. This experiment is an example of how this simulation platform can generate the minimum requirements for the design of an embedded system prototype, given appropriate conditions.

This platform is also a good tool for design and analysis of algorithms for navigation, because it allows the test of these algorithms without the utilization of real aircraft, thus reducing the development time and system costs.

VIII. Conclusion

This test platform is a great tool for design and study systems for quadricopter type UAV autopilot. Also it is possible to monitor the response of the aircraft in a variety of flight conditions.

This simulation tool is useful also to try easily various techniques of control and navigation algorithms for hardware-in-the-loop tests. This architecture also enables simulating quadricopter swarms and interactions with vehicles on the ground.

To continue this work, the simulation results will be compared with the real quadricopter, thus increasing the reliability of the results obtained in the simulation.

References

[1] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," *2007 IEEERSJ International Conference on Intelligent Robots and Systems*, no. 1, pp. 153-158, 2007.

[2] P. McKerrow, "Modelling the Draganflyer four-rotor helicopter," *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, no. May, pp. 3596-3601 Vol.4, 2004.

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

[4] R. Goel and S. Shah, "Modeling, Simulation and Flight Testing of an Autonomous Quadrotor," *of the IISc*, 2009.

[5] L. R. Ribeiro and N. M. F. Oliveira, "UAV autopilot controllers test platform using Matlab/Simulink and X-Plane," *2010 IEEE Frontiers in Education Conference (FIE)*, p. S2H-1-S2H-6, Oct. 2010.

[6] S. Santos, C. Nascimento, S. Givigi, A. Bittar, and N. Oliveira, "EXPERIMENTAL FRAMEWORK FOR EVALUATION OF GUIDANCE," *COBEM*, 2011.

[7] T. Indriyanto and Y. I. Jenie, "Modeling and Simulation of a Ducted Fan Unmanned Aerial Vehicle (UAV) Using X-Plane Simulation Software," 2010.

[8] 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.

[9] L. Research, "X-Plane Operation Manual," 2011. [Online]. Available: http://www.x-plane.com/files/manuals/X-Plane_Desktop_manual.pdf.

[10] Mathworks, "Simulink Overview," 2012. [Online]. Available: http://www.mathworks.com/products/simulink/.

[11] L. Research, "Plane Maker Manual," 2011. [Online]. Available: http://www.x-plane.com/files/manuals/Plane_Maker_manual.pdf.

[12] A. Bittar, "Tutorial do Plane-Maker," São José dos Campos, 2011.

[13] P. Beugnet, "Modélisation et interface de contrôle d ' un quadri-rotor," 2008.