

UAV Autopilot Controllers Test Platform Using Matlab/Simulink and X-Plane

Lucio R. Ribeiro and Neusa Maria F. Oliveira
Instituto Tecnológico de Aeronáutica, lregis@ita.br, neusa@ita.br

Abstract - Presently there is vast interest in UAV (Unmanned Aerial Vehicle) development given its civilian and military applications. One of the main UAV components is the autopilot system. Its development invariably demands several lab simulations and field tests. Generally after an UAV crash few parts remain usable. Thus, before embedding an autopilot system, it has to be exhaustively lab tested. With educational and research purposes in autopilot control systems development area, a test platform is herein proposed. It employs Matlab/Simulink to run the autopilot controller under test, the flight simulator X-Plane with the aircraft to be commanded, a microcontroller to command model aircraft flight control surfaces and a servo to drive these control surfaces. These resources are interconnected through data communication buses. So that, the autopilot controller designed on Matlab/Simulink is tested by controlling an aircraft on X-Plane. The inputs given to the aircraft flight control surfaces in the X-Plane are simultaneously sent to the microcontroller which translates these commands into effective servo movement control. Through this platform, designed autopilot systems can be applied into models similar to real aircraft minimizing risks and increasing flexibility for design changes. As study case, tests results from a roll attitude autopilot system are presented.

Index Terms - Autopilot, X-Plane.

INTRODUCTION

The fast advancement of modern aircraft design has required the development of many technologies such as aerodynamics, structures, materials, propulsion and flight controls.

Currently, the aircraft design strongly relies on automatic control systems to monitor and control many of the aircraft subsystems. Those control systems can also provide artificial stability to improve the flying qualities of an aircraft.

The autopilot systems were firstly conceived to primarily stabilize the aircraft after being disturbed from its wing-level equilibrium flight attitude [1].

The modern autopilot systems are much more complex and are essential to flight aiding crew in navigation, flight

management, stability augmentation and landing operations, for example.

With the development of modern aircraft new problems arose due to their dynamic stability. For instance, some oscillations inherent to the aircraft can be adequately damped or controlled by the pilot if their period is around 10 seconds or more. On the other hand, if the period of oscillation is 4 seconds or less, the pilot's reaction time is not short enough. Thus, such oscillations have to be artificially well damped [2].

The oscillations named "Short period" in pitch for longitudinal motion, and "Dutch roll" for lateral-directional motion fall into the category of a 4-second oscillation. In more conventional aircraft that kind of oscillation may be damped by handling their constructive characteristics. However, for modern jets artificial damping is required. It is provided by an automatic system [2].

The above concepts are also applicable to UAVs (Unmanned Aerial Vehicle) whose development are presently being object of vast interest given the increasing possibilities of its civilian and military applications. These applications include reconnaissance, surveillance, search and rescue, remote sensing, traffic monitoring missions, etc [3].

One of the main components of UAVs is the autopilot system. The autopilot system development invariably demands several lab simulations and field tests. The later ones are of high risk particularly when conducted on UAVs. Generally after an UAV crash few parts remain usable. Thus, before embedding an autopilot system, it has to be exhaustively lab tested.

Based on this necessity, with educational and research purposes in autopilot control systems development area, a test platform is herein proposed.

OBJECTIVE

This present article intends to describe a test platform developed to aid in the autopilot design process.

The test platform herein proposed provides an environment where the designed autopilot system can be applied into models very similar to real aircraft. Several flight situations can be simulated. Parameters of flight as well as aircraft responses can be monitored and easily analyzed. These features increase the flexibility to implement changes and immediately check out the results.

Basically, this test platform employs the Matlab/Simulink to run the autopilot controller to be tested, the flight simulator X-Plane with the aircraft to be commanded, a microcontroller to command model aircraft flight control surfaces and a servo to drive these control surfaces. All this resources are interconnected through data buses in order to exchange information.

As study case, this article presents the results obtained from tests conducted over an autopilot control system designed for lateral movement, specifically for roll mode control. This system was conceived for Boeing 747-400 aircraft.

AIRCRAFT TRANSLATIONAL MOTIONS

An airplane in free flight has three translational motions (vertical, horizontal and transverse), three rotational motions (pitch, yaw and roll) and numerous elastic degrees of freedom.

In order to reduce the complexity of this mathematical modeling problem, some simplifying assumptions may be applied. First, it is assumed that the aircraft motion consists of small deviations from its equilibrium flight condition. Second, it is assumed that the motion of the airplane can be analyzed by separating the equations into two groups [1]:

- Longitudinal Equations: composed by X-force, Z-force and pitching moment equations,
- Lateral Equations: composed by Y-force and yawing and rolling moments equations.

Figure 1 shows the above mentioned aircraft reference axes, forces and moments.

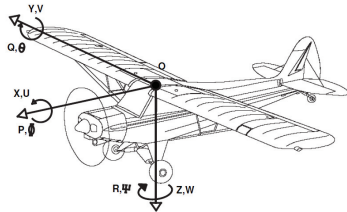


FIGURE 1
AIRCRAFT AXES, FORCES AND MOMENTS

I. Lateral-directional motion

The lateral-directional motion of an aircraft disturbed from its equilibrium state is a combination of rolling, yawing and sideslipping motions. This interaction between roll and yaw produces a coupled motion [1].

The lateral-directional motion, specifically the roll motion, is the movement intended to be simulated on the proposed test platform as demonstration example. To simplify the studied case, it was chosen to address specifically the roll motion in this work. Even though this simplifies the dynamic model equations, the platform functionality and characteristics are very well demonstrated.

II. Roll motion closed-loop control system

The aircraft roll motion can be controlled by an autopilot system block diagram as shown in the Figure 2 [1].

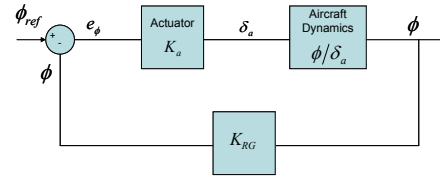


FIGURE 2
A ROLL ATTITUDE AUTOPILOT SYSTEM

Through this simplified system, the roll angle (ϕ) can be controlled by a reference angle applied as input. Primarily, the roll attitude autopilot is designed to maintain the aircraft with wings leveled, or with $\phi = 0$. Usually, this simplified system does not fulfill the control design performance requirements in relation to damping ratio, overshoot and undamped natural frequency.

So that, in the study case herein presented, a more efficient control system is employed [1]. It is shown in Figure 3.

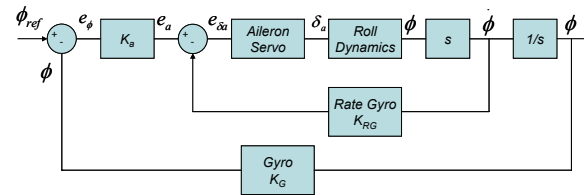


FIGURE 3
ROLL ATTITUDE AUTOPILOT SYSTEM

In this case, an accurate aircraft roll to aileron transfer function for Boeing 747-400 aircraft is employed in the roll autopilot system design. This transfer function was calculated based on aircraft physical characteristics and stability derivatives as detailed in [5]. Moreover, a typical aileron actuator transfer function is employed.

The roll attitude autopilot system gains K_G , K_a and K_{RG} are calculated through the root locus method considering the above rationale.

THE TEST PLATFORM

The main components of the test platform are as follows:

- Matlab/Simulink containing the autopilot control system,
- Flight Simulator X-Plane containing the aircraft model to be controlled,
- Microcontroller to command model aircraft flight control surfaces,

- Model aircraft digital servo to drive these control surfaces.

The test platform concept is based on the block diagram presented in Figure 4. In the test platform, the block “Controller” is replaced by the designed autopilot system model. This model runs into the Matlab/Simulink environment.

Similarly, the block “Aircraft Dynamics” is replaced by the flight simulator X-Plane with the aircraft to be controlled.

Thus, the basic principle of the test platform consists in establishing the communication between Matlab/Simulink, X-Plane, microcontroller and servo in the following mode:

The parameters calculated by the autopilot control system are sent to X-Plane in order to command the aircraft flight control surfaces. The X-Plane calculates the new aircraft attitude according to the inputs received from Matlab/Simulink. The X-Plane sends those new aircraft attitude parameters back to Matlab/Simulink closing the loop. Matlab/Simulink restarts the process by providing updated commands to X-Plane aircraft.

The inputs given to the aircraft flight control surfaces in the X-Plane are simultaneously sent to a microcontroller which translates these commands received from Matlab/Simulink into effective servo movement control. The microcontroller calculates the deflection angle to be imposed to the flight control surfaces and generates a proportional PWM signal to command the servo. The model aircraft flight control surfaces reproduce the same deflection observed on the X-Plane aircraft.

The communication between Matlab/Simulink and X-Plane is made through UDP (User Datagram Protocol).

Between Matlab/Simulink and microcontroller it used RS-232 serial communication.

Block diagram shown by Figure 4 summarizes the platform concept.

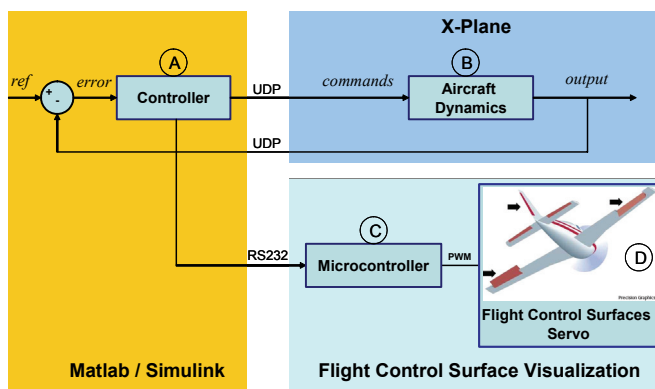


FIGURE 4
TEST PLATFORM BLOCK DIAGRAM

The flight simulator X-Plane provides very accurate aircraft models and has a very important feature: the possibility to exchange data with external systems [6]. Giving its realistic simulations capability, X-Plane is also

FAA (Federal Aviation Administration) approved for pilot training. The aircraft models simulated in X-Plane are built based on their real physical dimensions, power and weight among other characteristics. X-Plane is not considered a game, but can be categorized as an engineering tool that can be used to predict the flying qualities of fixed and rotary wing aircraft [6].

Most other flight simulators use the stability derivatives method to compute how an airplane flies. This technique involves simply forcing the aircraft nose to return to a centered position along the flight path with certain acceleration for each degree of offset from straight-ahead flight of the airplane. This is too simplistic to be used across the entire flight envelope of the airplane. Stability derivatives will not normally take into proper account the asymmetric effects of engine failures, the chaotic effects of turbulence, stalls, spins and the myriad of dynamic effects that airplane flight generates. In other words, the commonly used stability derivatives are gross over simplifications of how an airplane flies. In summary, those simulators can not predict how the airplane will fly. Basically, the airplane designer teaches the simulator how the airplane should fly, and the simulator reproduces that information right back to the user [6].

X-Plane instead, assimilates the geometric shape of any aircraft and then figures out how that aircraft will fly. It does this by an engineering process called “blade element theory”, which involves breaking the aircraft down into many small elements and then finding the forces on each little element many times per second. These forces are then converted into accelerations which are then integrated to velocities and positions. This method of computing the forces on the airplane is much more detailed, flexible, and advanced than the flight model that is used by most other flight simulators. By doing this process, X-Plane accurately predicts what will be the performance and handling qualities of an airplane of a given geometry [6].

The microcontroller represented by block C on Figure 4 symbolizes the experimenter board MSP-EXP430F5438 manufactured by Texas Instruments and shown in Figure 5.

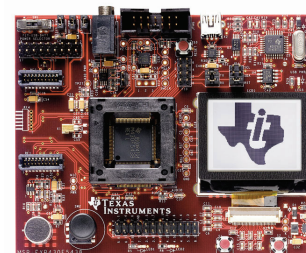


FIGURE 5
MICROCONTROLLER UNIT

This is a development kit based on MSP430 microprocessor family with clock of 18 MHz, low power consumption, featured with a variety of I/O (digital, PWM), besides RAM and flash memory space. The choice for this

device is justified by its high processing capability as well as easy integration with digital servos through PWM I/O ports.

On block D of Figure 4 it is represented the digital servo that commands model aircraft flight control surfaces. A typical servo is shown in Figure 6.



FIGURE 6
MODEL AIRCRAFT DIGITAL SERVO

It receives PWM signals from the microcontroller unit and converts them into proportional axis movement.

X-PLANE DATA INTERFACE

Flight Simulator X-Plane has an important feature that is essential to this test platform development. It is its capacity of sending and receiving data to and from other devices. One way to implement this communication is by employing protocol UDP. Data packets are sent and received through the microcomputer Ethernet port. Figure 4 shows UDP protocol being used to establish the communication between Matlab/Simulink (Controller) and X-Plane (Aircraft Dynamics).

UDP uses a simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and data packets may arrive out of order, appear duplicated or go missing without notice. Error checking and correction are considered either not necessary or performed in the application. This way, UDP avoids the overhead of such processing at the network interface level being extremely fast [4]. Time-sensitive applications such as this test platform often use UDP once dropping packets is preferable to waiting for delayed packets, which it is not an option in real-time systems.

Hence UDP speed constitutes a key point in the test platform once the communication between Matlab/Simulink and X-Plane must be fast enough to synchronize commands, data processing and system responses.

X-Plane is able to send or receive up to 99.9 data packets per second via UDP. Each data package may be configured to carry aircraft parameters data that are selected on X-Plane Data Input and Output interface.

For example, in the case of lateral motion, parameters such as roll, yaw, pitch, altitude and speed shall be selected for transmission. Each parameter receives a numeric label for proper identification.

Figure 7 shows the interface that serves to select data for input and output from X-Plane. Besides parameters selection for UDP data transmission or receiving, it also serves for selecting parameters to be shown during aircraft flight simulation, file recording and graphs construction.

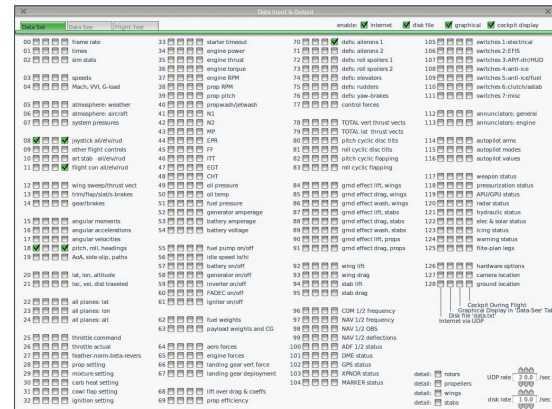


FIGURE 7
X-PLANE PARAMETERS USER INTERFACE

Once the parameters are selected on X-Plane, through UDP it is possible to receive them at Matlab/Simulink environment and after processing, to send commands back to X-Plane aircraft.

In the case of roll attitude autopilot proposed example, Matlab/Simulink generates the roll angle reference and control signal sending them to X-Plane. In its turn, X-Plane commands the simulated aircraft according to the inputs received. The new aircraft roll angle position is sent from X-Plane back to Matlab/Simulink, restarting the process.

The X-Plane data package follows a pattern shown on Figure 8.

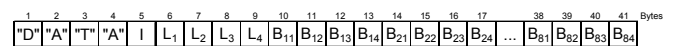


FIGURE 8
X-PLANE DATA PACKAGE PATTERN

Basically it is composed by a sequence of bytes that need to be properly interpreted. X-Plane uses what are known as single precision floating point variables for just about everything sent over the network. This means that the numbers can be stored using four bytes.

The first four bytes of the packet shown on Figure 7 represents the characters “DATA” used to indicate that this is a data packet. The fifth byte is an internal code (I). The next four bytes represent the parameter label (L_1, L_2, L_3, L_4). Following there are 8 sets of 4 bytes ($B_{11}, B_{12}, B_{13}, B_{14}$ to $B_{81}, B_{82}, B_{83}, B_{84}$) representing the data itself in single precision floating point. Taking each set of 4 bytes, the first bit is the sign bit. It tells whether the number is positive or negative. The next 8 bits are the biased exponent. The remaining 23 bits represents the mantissa. The next seven sets of 4 bytes complete the data.

So, the Matlab/Simulink model has to decode this data packet accordingly. It also has to properly encode the data to be sent to X-Plane. Most of the blocks of the test platform implemented on Matlab/Simulink environment are dedicated to decode and encode data packets from and to X-Plane.

IMPLEMENTATION

Figure 9 illustrates the test platform implemented.

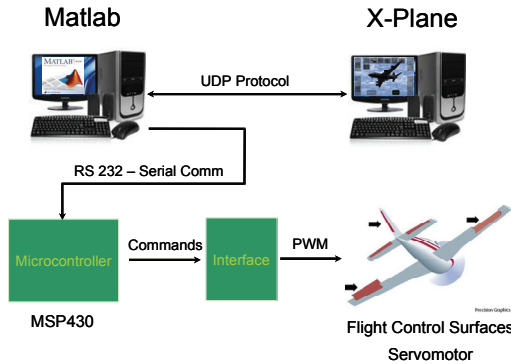


FIGURE 9
AUTOPILOT TEST PLATFORM IMPLEMENTATION

In one computer Matlab/Simulink runs a model containing the autopilot control system and other blocks responsible for UDP data packets decode and encode process as well as serial communication with microcontroller. In the other computer X-Plane simulates the aircraft to be controlled. Both computers communicate to each other through their Ethernet port using UDP protocol.

Through serial bus the microcontroller receives the same commands Matlab/Simulink sent to X-Plane. It decodes this data converting them into PWM signal. An interface amplifies this signal to apply it to the servo that commands the model aircraft flight control surfaces.

APPLICATION EXAMPLE – TEST RESULTS

In order to demonstrate how the test platform could be applied in laboratory classes and also help autopilot systems development, the roll attitude autopilot system was designed and integrated into the platform. The dynamic system used as the case to be studied and tested using the proposed platform can be represented by Figure 3. The gains K_G , K_a and K_{RG} of that system were calculated through the root locus method using the systems dynamics as in (1) and (2), which are the Boeing 747-400 roll to aileron and typical aileron servo transfer functions.

Roll Dynamics:

$$\frac{N_{\dot{\alpha}}^{\phi}}{\Delta_{Lat}} = \frac{s(149.42s^2 + 33.66s + 133.07)}{s(870.78s^4 + 628.63s^3 + 791.37s^2 + 438.00s + 5.47)} \quad (1)$$

Aileron Servo:

$$S(s) = \frac{10}{s + 10} \quad (2)$$

The designed system result is shown on Figure 10. It is also shown which blocks will be simulated by X-Plane and which ones will be integrated into the test platform model at Matlab/Simulink.

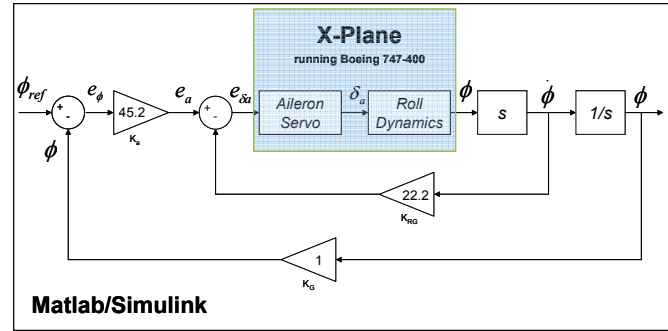


FIGURE 10
ROLL ATTITUDE AUTOPILOT SYSTEM FOR BOEING 747-700

The roll to aileron and servo transfer functions are simulated by the aircraft Boeing 747-400 at X-Plane. The other autopilot blocks are distributed into the test platform model. Figure 11 shows the entire test platform model implemented at Matlab/Simulink environment.

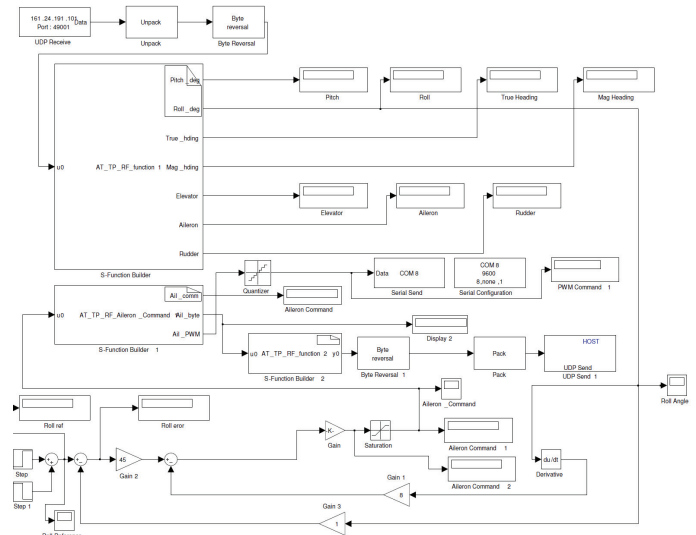


FIGURE 11
AUTOPILOT TEST PLATFORM MODEL AT MATLAB/SIMULINK

In a typical lab class, the student should find the test platform ready to be used with the communication interfaces appropriately configured. The gains K_G , K_a and K_{RG} shall then be fed into the controller implemented at Matlab/Simulink. The gains calculation may be performed at the lab or be previously done, being the lab classes preferably used for test at the platform.

To initialize the simulation, X-Plane is loaded with aircraft Boeing 747-400 in a cruise flight at 40,000 ft altitude. As soon as the test platform model shown on Figure 11 runs, the designed autopilot system takes the aircraft control. In principle, for any roll angle applied as reference into the autopilot system, the aircraft at X-Plane shall respond following that reference.

The reference signals applied as system input, the system response and also the commands sent to the aircraft flight control surfaces actuators can be monitored through real time graphs at Matlab/Simulink. In the proposed study case, the input is the reference roll angle, the system response is the new aircraft roll angle attitude and the commands are the deflections to be imposed to the ailerons. Figure 12 shows the results for the proposed roll attitude autopilot system implemented at the test platform.

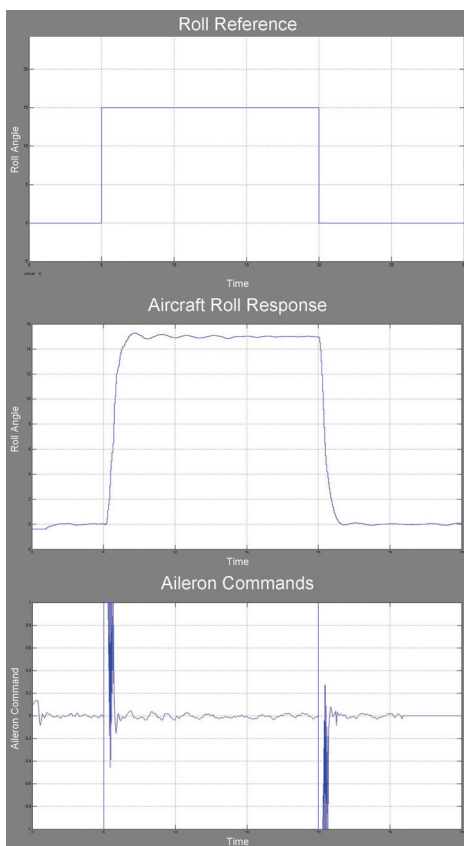


FIGURE 12
ROLL REFERENCE AND RESPONSE AND AILERON COMMANDS

The top graph on Figure 12 presents the autopilot roll angle reference applied into the system. Center graph shows the aircraft roll response. Bottom graph presents the commands sent from the autopilot system to Boeing 747-400 ailerons at X-Plane. As expected, the commands to ailerons become more intense when the roll angle reference changes from 0 to 15 degrees and from 15 to 0 degrees. During simulation it is possible to see the Boeing 747-400 assuming 15 degrees bank angle attitude and performing a slight turn

to the right. In parallel it is possible to verify the servo movements in response to these commands too. Interesting to notice that even with the aircraft leveled and keeping a certain roll attitude there are small commands to the flight control surfaces. This is something that only was possible to observe due to the realism provided by X-Plane simulation which introduces small perturbations to aircraft flight reproducing a real atmosphere with wind, turbulences, etc.

One exercise that can also be performed on the designed autopilot system is to vary the loop gains and check it out the aircraft responses. This can be used to properly tune the loop gains to optimize system performance.

CONCLUSION

The development of this test platform resulted in a valuable tool to aid autopilot systems study and design. It allows monitoring the aircraft responses for a designed autopilot system with high degree of realism. It is possible to change control system parameters very easily and check the results out in a friendly environment. This possibility eases design task as well minimizes risks of embedding the system for field tests.

The test platform also permits the study of longitudinal and lateral-directional movement of a specific aircraft or ultimately the model aircraft for which the autopilot system is being designed.

A possible extension of this study is to simulate in test platform the design of a full lateral-directional autopilot system that encompasses roll and yaw angles control once in practice these two motions work coupled.

REFERENCES

- [1] NELSON, R. C. "Flight Stability and Automatic Control" McGraw-Hill Second Edition, 1998
- [2] BLACKLOCK J. H. "Automatic Control of Aircraft and Missiles" Wiley Interscience Publication, 1991
- [3] ADANG SUWANDI AHMADI, JAKA SEMBIRING, "Hardware in the Loop Simulation for Simple Low Cost Autonomous UAV (Unmanned Aerial Vehicle) Autopilot System Research and Development" Institut Teknologi Bandung, Indonesia, 2007
- [4] DANIEL ERNST, "Development of Research Platform for Unmanned Vehicle Controller Design, Evaluation, and Implementation System: From MATLAB to Hardware Based Embedded System", Department of Computer Science and Engineering College of Engineering University of South Florida, Florida, 2007
- [5] ROSKAM, J. "Airplane Flight Dynamics and Automatic Flight Controls" PART I. Roskam Aviation and Engineering Corporation, 1982
- [6] LAMINAR RESEARCH, "X-Plane Description", X-Plane Manual, 2009