24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-1764
UAV CONTROL LOOPS DESIGN TESTED IN AN HIL PLATAFORM

Marcelo Henrique dos Santos
Neusa Maria Franco Oliveira
Instituto Tecnológico de Aeronáutica, Departamento de Engenharia Eletrônica e de Computação. Praça Marechal Eduardo Gomes, 50. Vila das Acácias. São José dos Campos, SP – Brazil
marcelo.h.santtos@gmail.com, neusa@ita.br

Benedito Carlos de Oliveira Maciel
FT Sistemas. Rodovia Presidente Dutra, Km 138. Parque Tecnológico. São José dos Campos, SP – Brazil
bcomaciel@flighttech.com.br

Abstract. *Unmanned Aerial Vehicles (UAVs) were initially idealized to perform military applications, however the subject is gradually gaining force as the applications begin to gain notoriety not only in the academic institutions, but also in the industry and in military operations. One of the main control systems in a UAV is the autopilot system (AP). The development and validation of this kind of system requires numerous simulations, and the field tests involve risks, because the damages after the fall of the airplane are often irreversible. In the scenario presented, this article has the objective to develop an embedded system for the autonomous control of a small fixed wing unmanned aerial vehicle. For this work the Piper J-Cub 1/4 scale aircraft was used and its non-linear mathematical model that describes the equations of motion was simulated in a computational environment using the MATrix LABtory software (MATLAB®). The flight control system proposed in this article has the objective of allowing an autonomous flight and was designed using classic control techniques. For the navigation system the direction cosine matrix method was used to determine the attitude of the aircraft. The Line of Sight (LoS) algorithm responsible for calculating the reference directional angle was implemented as the guidance system. This article also proposes the communication between the embedded AP system and the dynamics of the aircraft creating a hardware-in-the-loop to simulate a flight trajectory and the results were effective for a set of waypoints trajectory.*

Keywords: *unmanned aircraft systems, airspace control, hardware-in-the-loop*

1. INTRODUCTION

Unmanned Aerial Vehicle (UAV) is a type of aircraft controlled without a on-board pilot human (Barnhart *et al.*, 2011). Although these aircraft can be controlled remotely, the use of autonomous control systems has been widely used (Kemp, 2010). One of the main control systems in a UAV is the autopilot system (McManus *et al.*, 2003).

Autopilot (AP) can be defined as a system designed to control vehicles without human interference. This system can be coupled to several types of terrestrial, aquatic, or aerial autonomous vehicle (Nelson, 1998). In general, an autopilot is composed of navigation, guidance, control subsystems and a control ground station (Bittar, 2012).

The development and validation of AP system requires numerous simulations, and the field tests involve risks because, the damages are irreversible after a fall of the aircraft (Adiprawita *et al.*, 2007). In order to avoid losses, AP systems have been developed and tested in the laboratory before being embedded for real testing with the model aircraft. The development of test platforms to simulate the flight dynamics of an aircraft and its behavior according to the autopilot system has been commonly studied. In this context it is possible to find in the literature several concepts of Hardware-In-the-Loop (HIL) and Software-In-the-Loop (SIL) which are indispensable tools for the rapid certification of autopilot, avionics and control software with minimal cost and effort (Jung and Tsiotras, 2007; Ribeiro, 2011).

This paper will present the development of the embedded system of an AP capable of performing a flight mission autonomously for the Piper J-Cub 1/4. The subsystem designed presented and mainly discussed was the Control subsystem. The tests and results presented were done using a HIL (Hardware-in-the-Loop) simulation. The remaining text is organized as follows: Section 2 presents the problem and the plant model with the nonlinear equations of the aircraft used in simulations. Section 3 presents the design of longitudinal and lateral-directional controllers of the aircraft. Section 4 presents the simulation results and some considerations. Finally, Section 5 concludes the paper and presents future works.

2. AIRCRAFT MODEL

This section describes the development of the nonlinear mathematical model of flight dynamics of a fixed wing unmanned aerial vehicle. The mathematical model of the aircraft was implemented on Matlab/Simulink environment through `sfunctions`. Figure 1 shows a simplified block diagram layout of the nonlinear simulation model for a six-degree of freedom aircraft (6-DOF).

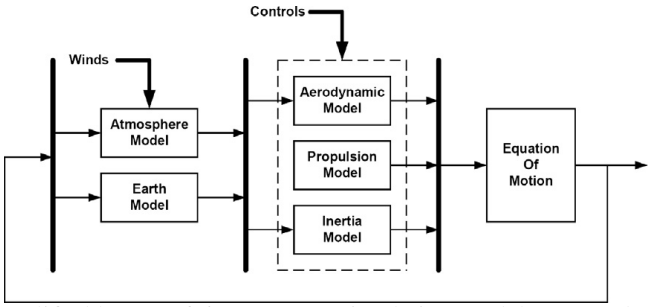


Figure 1: Simplified layout of 6-DOF UAV simulation model (Paw and Balas, 2011).

In this article the flat-Earth model was used, where the Earth is assumed as an inertial frame and its accelerations are considered insignificant.The atmospheric model as well as the Earth model is independent of the UAV platform used. The aerodynamic and propulsive models will not be detailed and the deformations that the aircraft suffers when subjected to aerodynamics efforts will not be considered.

The motion equations for a (6-DOF) aircraft can be derived from Newton's Second Law, which demonstrates that, the sum of the external forces acting on a body must be equal to the temporal variation rate of its linear momentum , and the sum of the external moments acting on the body must be equal to the temporal variation rate of its angular momentum (Maciel, 2008).

The simplifications used, such as not considering Centrifuge and Coriolis accelerations, or deformation in aircraft can be detailed in (Nelson, 1998; Stevens and Lewis, 2003). In this work we will use two coordinate system to describe the complete set of equation: the first coordinate system can be thought of as a local tangent plane or navigation system North-East-Down (NED); and the second coordinate system is attached to a moving aircraft that is at an arbitrary orientation relative to the tangent plane, referred to as body coordinate system. For a better understanding of the proposed problem, the complete set of equations describing the model used is organized as state space form, as shown below:

Force Equations:

$$\dot{U} = -qW + rV - g\sin\theta + {}^{(X_A+X_T)}/_{m} \tag{1}$$

$$\dot{V} = -rU + pW + g\sin\phi\cos\theta + {}^{(Y_A+Y_T)}/_{m} \tag{2}$$

$$\dot{W} = -pV + qU + g\cos\phi\cos\theta + {}^{(Z_A+Z_T)}/_{m} \tag{3}$$

Moment Equations:

$$\dot{p} = \frac{-J_{zz}L^{ext} - J_{xz}N^{ext} + J_{xz}\left(-J_{xx} + J_{yy} - J_{zz}\right)pq + \left(J_{xz}^2 + J_{zz}^2 - J_{yy}J_{zz}\right)qr}{J_{xz}^2 - J_{xx}J_{zz}} \tag{4}$$

$$\dot{q} = \frac{M^{ext}\left(J_{zz} - J_{xx}\right)pr + J_{xz}\left(r^2 - p^2\right)}{J_{yy}} \tag{5}$$

$$\dot{r} = \frac{-J_{xz}L^{ext} - J_{xx}N^{ext} + \left(J_{xx}J_{yy} - J_{xx}^2 - J_{xz}^2\right)pq + J_{xz}\left(J_{xx} - J_{yy} - J_{yy}J_{zz}\right)qr}{J_{xz}^2 - J_{xx}J_{zz}} \tag{6}$$

Kinematic Equations:

$$\dot{\phi} = p + \tan\theta(q\sin\phi + r\cos\phi) \tag{7}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{8}$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta} \tag{9}$$

Navigation Equations:

$$\dot{x}_N = U\cos\theta\cos\psi + V(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) + W(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \tag{10}$$

$$\dot{y}_E = U\cos\theta\sin\psi + V(\sin\phi\sin\theta\sin\psi - \cos\phi\cos\psi) + W(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \tag{11}$$

$$\dot{z}_D = -U\sin\theta + V\sin\phi\cos\theta + W\cos\phi\cos\theta \tag{12}$$

where U (m/s), V (m/s) and W (m/s) are the body axes linear velocities; p (rad/s), q (rad/s) and r (rad/s) are the body axes angular rates; ϕ (rad), θ (rad) and ψ (rad) are the body attitude angles; and x_N (m), y_E (m) and z_D (m) are the coordinates North, East and Down $(\text{altitude}\,(h))$ in the navigation coordination system. When necessary, it is important change the input and output variables scale in order to adjust them to the units that you want to work with.

Finally, the aircraft used as a case study in this work is the model Piper J-3 Cub 1/4 scale. More information about the aircraft, manufacturer, geometric aspects and moving control surfaces are described in (Jensen *et al.*, 2004). The mathematical modeling of this aircraft has already been the subject of previous research, and the aerodynamic and momentum coefficients used in the non-linear mathematical model of simulation implemented were estimated in (Du, 2011).

2.1 LINEARIZATION

For the development of the control project of an autopilot using the classical control theory, it is necessary to linearize the nonlinear mathematical model that describes the set of equations of the motion. For the linearization of the model it is necessary to place it in a equilibrium region. The search for these equilibrium values in the aeronautical context is also known as trimming, “tuning" the control surfaces and the torque applied to the engine for wings-level flight.

In this paper, the states and the equilibrium inputs used were obtained using MATLAB’s `fsolve` function to solve the nonlinear equations of the mathematical model. Applying the trimming conditions, it is possible to calculate a set of linear equations that approximate the dynamics of motion near the equilibrium point, as in Eq. (13). Normally the Δ notation is discarded, meaning $\mathbf{x}$ and $\mathbf{u}$ refer to deviations from the equilibrium (Franklin *et al.*, 2013).

$$\begin{cases} \Delta\dot{\mathbf{x}} = [A]\,\Delta\mathbf{x} + [B]\,\Delta\mathbf{u} \\ \mathbf{y} = [C]\,\Delta\mathbf{x} + [D]\,\Delta\mathbf{u} \end{cases} \tag{13}$$

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x_{trim}}$, $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u_{trim}}$ and matrices $[A]$, $[B]$, $[C]$ and $[D]$ are obtained from the Taylor series first order aproximation of the non-linear model written as a function $\mathbf{x}$ and $\mathbf{u}$.

After the correct triming of the non-linear mathematical model, the system operates in steady-turn, in other words, it is possible to divide the study of the motion of the aircraft in the longitudinal and lateral-directional motions, since the coupling of these dynamics is minimal and can be disregarded (Stevens and Lewis, 2003).

2.1.1 Longitudinal Dynamics

The linear longitudinal model obtained according to the described trimming is represented by matrices $[A]$, $[B]$, $[C]$ and $[D]$ that describe the longitudinal equations of motion for Piper J-3 Cub 1/4 in the state space, defined according to Eq. (14).

$$[A] = \begin{bmatrix} -0.1068 & 0.4047 & -0.7378 & -9.8016 & 0.0000 \\ -0.6310 & -6.9080 & 22.9883 & -0.3147 & -0.0009 \\ 0.3972 & -12.3859 & -26.2910 & 0 & -0.0000 \\ 0 & 0 & 1.0000 & 0 & 0 \\ -0.0321 & 0.9995 & 0 & -23.0000 & 0 \end{bmatrix} \quad [B] = \begin{bmatrix} 11.6874 & 0 \\ 0 & 0 \\ 0 & -168.0094 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$[C] = \begin{bmatrix} 0.9995 & 0.0321 & 0 & 0 & 0 \\ -0.0014 & 0.0435 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & -1.0000 \end{bmatrix} \quad [D] = \begin{bmatrix} 0 \end{bmatrix}_{5\times 2} \tag{14}$$

where, the state, control and output vectors are presented respectively by Eq. (15).

$$\begin{aligned} \mathbf{x} &= [U \quad W \quad q \quad \theta \quad h]^T \qquad \mathbf{u} = [\delta_T \quad \delta_e]^T \\ \mathbf{y} &= [V_T \quad \alpha \quad q \quad \theta \quad h]^T \end{aligned} \tag{15}$$

2.1.2 Lateral-Directional Dynamics

The linear lateral-directional model obtained according to the described trimming is represented by matrices $[A]$, $[B]$, $[C]$ and $[D]$ that describe the lateral-directional equations of motion for Piper J-3 Cub 1/4 in the state space, defined according to Eq. (16).

$$
\begin{aligned}
[A] &= \begin{bmatrix} -0.2397 & 1.1379 & -22.6223 & 9.8016 & 0 \\ -3.6259 & -45.9261 & 17.6378 & 0 & 0 \\ 1.6642 & -2.9615 & -3.9523 & 0 & 0 \\ 0 & 1.0000 & 0.0321 & 0 & 0 \\ 0 & 0 & 1.0005 & 0 & 0 \end{bmatrix} \quad [B] = \begin{bmatrix} 0 & -3.0921 \\ -368.7231 & 0.6738 \\ -11.6682 & 31.0398 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\
[C] &= \begin{bmatrix} 0.0435 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix} \quad [D] = \begin{bmatrix} 0 \end{bmatrix}_{5\times 2}
\end{aligned}
\tag{16}
$$

where, the state, control and output vectors are presented respectively by Eq. (17).

$$
\begin{aligned}
\mathbf{x} &= [V \quad p \quad r \quad \phi \quad \psi]^T \qquad \mathbf{u} = [\delta_a \quad \delta_r]^T \\
\mathbf{y} &= [\beta \quad p \quad r \quad \phi \quad \psi]^T
\end{aligned}
\tag{17}
$$

3. CONTROLLER DESIGN

In this section, we present the development of the controllers for the proposed PA. In order to better understand the proposed problem, the model of the plant was divided into two systems that represent the longitudinal and lateral-directional dynamics of the aircraft.

The operation of the proposed autopilot can be divided into flight phases. In a typical cruise flight phase, the UAV will be flying in a desired cruising flight condition and will conduct waypoint navigation with yaw angle tracking with the speed plus pitch-attitude hold and altitude hold modes alternating between each other.

To simplify, the dynamics of the aircraft actuators have not been modeled and therefore will be considered ideal. For the designed dynamic compensators, proportional-integral (PI) controllers were added to eliminate the steady-state error. The option for a PI controller is due to the fact that proportional action is indispensable for a good transient control system response, combined with the integral action that is responsible for a small steady-state error (Franklin *et al.*, 2013). The initial tuning for PI controller gains was done using the Ziegler and Nichols (1942) method and the detailing of each autopilot mode implemented for lateral-directional and longitudinal dynamics will be presented in the following subsections.

3.1 Longitudinal Controller

The longitudinal control system is composed by the combination of the altitude hold and speed plus pitch-attitude hold modes presented in subsections 3.1.1 and 3.1.2, respectively. The controllers used in the longitudinal control loops ensure that the aircraft does not perform abrupt motions returning to the steady-state after reaching the desired reference values.

The controllers designed in the longitudinal control loops act in parallel with each other. If the reference altitude (h_r) is above or below the aircraft by up to 5 meters, the PA altitude hold is activated, otherwise the PA speed plus pitch-attitude hold is activated until the altitude of the aircraft is 5 meters reference, this time the autopilot mode is switched to the altitude controller.

3.1.1 Speed plus Pitch-Attitude Hold

Figure 2 shows the block diagram of the speed plus pitch-attitude hold autopilot control loops. This control design considers the two control inputs simultaneously (δ_{T}, δ_{e}), the combination of pitch-attitude autopilot for elevator control (δ_e), along with speed-hold autopilot for the power throttle control (δ_T).

The pitch-attitude hold autopilot is normally only used when the aircraft is in wings-level flight, where the controlled variable is θ ($\theta = \gamma + \alpha$). The controller does not hold the flight-path angle (γ), constant because the angle of attack (α) changes with flight conditions. Because of these characteristics only the pitch-attitude-hold autopilot is not very important in an flight control project, however, the same configuration of this AP can be used as inner loops of other autopilots, such as altitude hold and automatic landing (Stevens and Lewis, 2003).

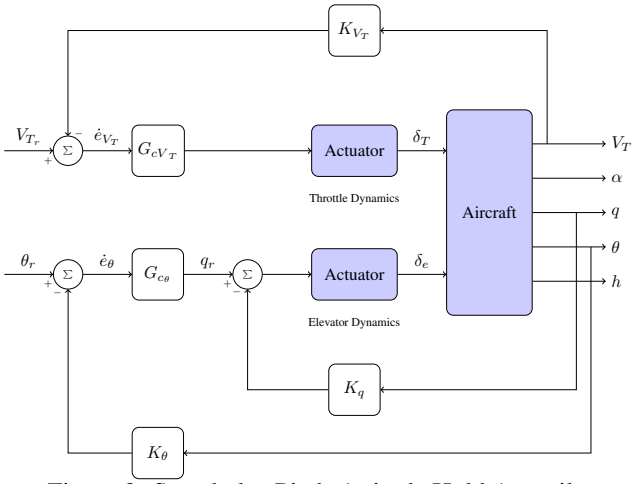


Figure 2: Speed plus Pitch-Attitude Hold Autopilot

This controller architecture is composed for q and θ feedbacks together with a dynamic compensator G_{c_θ}. The gain K_q equal 0.0228 was determined by applying the root locus method (Franklin *et al.*, 2013) as a function of the values obtained corresponding to the damping ratio and the natural frequency for the short-period mode. The gain K_θ of the pitch angle feedback is unitary. As the Ziegler-Nichols method provided approximate gains, the system response to the calculated parameters showed a very high overshoot, so the controller parameters K_p and T_i were adjusted with the respective gains (-0.7434, 0.7632).

The closed loop transfer function relating the pitch angle (θ) and the input reference (θ_r) is described by Eq.(18). It is observed that the transfer function has a zero and a pole near the origin that practically cancel out.

$$\frac{\theta(s)}{\theta_r(s)} = \frac{124.9(s+6.87)(s+1.31)(s+0.1445)(s+0.0001003)}{(s+0.1309)(s+0.0001003)(s^2+1.871s+2.431)(s^2+27.47s+510.8)} \tag{18}$$

The speed hold controller architecture is composed of the airspeed feedback of the aircraft (V_T) with a dynamic compensator G_{cV_T} for the throttle. The gain K_{V_T} of the feedback is unitary and the gains K_p and T_i of the adopted PI controller have been adjusted with the respective gains (0.05056, 6.3275).

The closed-loop transfer function between the aircraft velocity (V_T) and the input reference (V_{T_r}) is described by Eq.(19). It is important to note that the state of the system has been increased due to the introduction of integrator in the two controllers (G_{c_θ}, G_{cV_T}).

$$\frac{V_T(s)}{V_{T_r}(s)} = \frac{0.59061(s+0.158)(s+0.0001379)(s^2+1.878s+2.207)(s^2+27.47s+510.8)}{(s+0.4742)(s+0.171)(s+0.0001379)(s^2+1.948s+2.54)(s^2+27.47s+510.8)} \tag{19}$$

For the parameters used, the closed loop system presented a satisfactory response, as shown by Fig.(3). For θ_r a unitary step with amplitude equal to 5 degrees was applied at the instant of time equal to 20 seconds, while, the reference velocity aerodynamic of the aircraft V_{T_r} was maintained in the region of equilibrium.

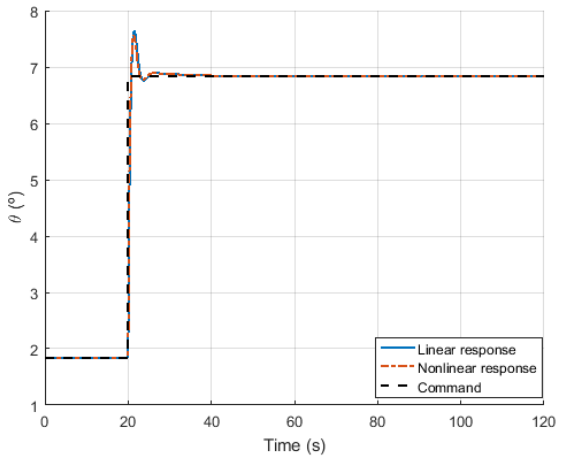
It is observed that for the time response of the pitch angle (θ) shown by Fig.3a the settling time is about 5.29 seconds with an overshoot of approximately 1.77%. Figure 3b shows that due to the altitude increasing an airspeed decreasing occurs. However the designed controller was able to return the aircraft to its initial airspeed.

3.1.2 Altitude-Hold

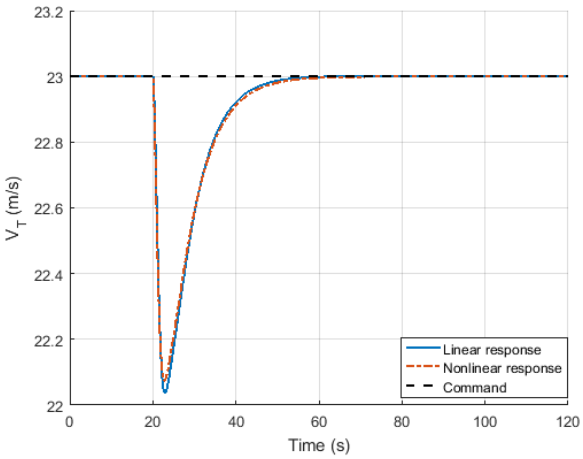
The altitude hold mode implemented in this paper is proposed by Stevens and Lewis (2003). The pitch-attitude hold AP detailed earlier is used as an inner loop of the project, and the dynamic compensator G_{c_h} is designed so that a good altitude control is obtained.

Complementing the pitch-attitude hold controller design, the aircraft altitude (h) is feedback and the controller G_{c_h}, is added between the external and inner loop to eliminate the steady-state error. The gain K_h of the feedback is unitary, so the only gains to be defined are the controller parameters K_p and T_i. The system response to the gains using the Ziegler-Nichols method was unstable, so the gains (0.0405, 51.8520) were adopted to obtain a stable response with small overshoot and larger settling time, resulting in a smoother response to the attitude control.

The closed-loop transfer function relating the altitude (h) and the input reference (h_r) is described by Eq.(20).



(a) Pitch angle response in the time domain

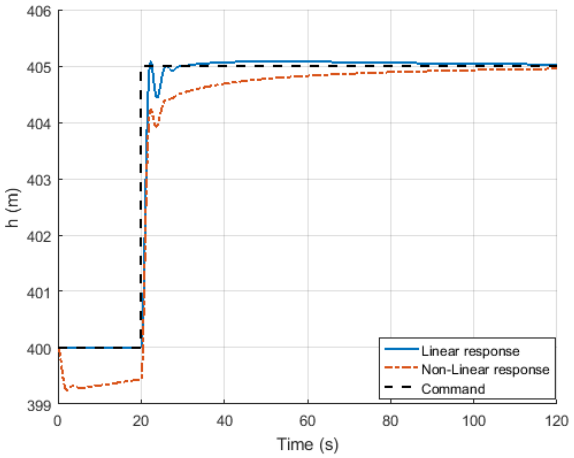


(b) Aerodynamic Velocity response in the time domain

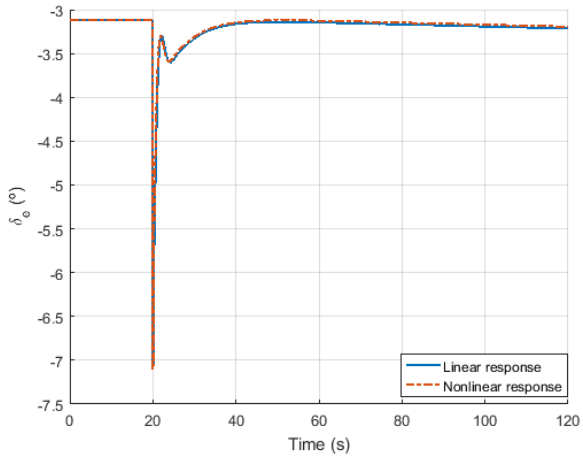
Figure 3: Response of the Aircraft for a Reference in the Pitch Angle and Aerodynamic Velocity

$$\frac{h(s)}{h_r(s)} = \frac{-0.0013473\,(s - 1.83 \cdot 10^6)\,(s + 1.287)\,(s + 0.09151)}{(s + 32.65)\,(s + 0.391)\,(s + 0.08175)\,(s^2 + 2.436s + 4.842)\,(s^2 + 7.744s + 57.48)} \tag{20}$$

The closed-loop system thus presented a satisfactory response as shown in Fig.4. For the altitude reference value (h_r) a step with amplitude of 5 meters was applied at the instant of time equal to 20 seconds.



(a) Altitude response in the time domain



(b) Elevator control surface response in the time domain

Figure 4: Response of the Aircraft for a Reference in the Altitude

For the time altitude response shown by Fig.4a the altitude settling time is about 5.29 seconds with an overshoot of approximately 1.77%. Figure 4b shows that to produce the expected altitude response an approximate deflection of 4 degrees on the control surface (δ_e) is required. It is emphasized that in the altitude autopilot mode, all states of the system return to the initial condition as desired, as the aircraft returns to the equilibrium condition in wings-level flight.

3.2 Lateral-Directional Controller

Automatic navigation is an important autopilot function for military and civil aircraft. Heading-hold autopilot is designed to hold the aircraft on a given compass heading. The conventional method of implementing this autopilot is to add additional feedback to the yaw angle (ψ) using the roll angle hold mode to control the bank angle (ϕ) as an internal loop of the controller design, the diagram of the proposed autopilot heading hold is presented by Fig. 5.

The roll angle hold loop and the turn coordinate designed are detailed in the next subsections.

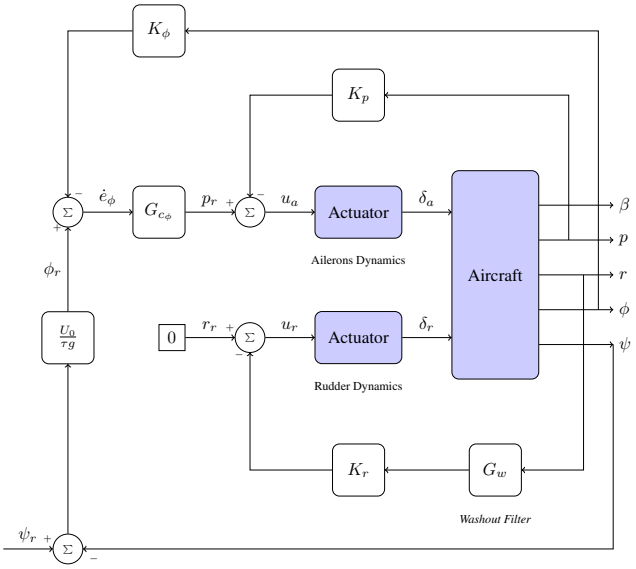


Figure 5: Heading Hold Autopilot

3.2.1 Roll-Angle Hold

This autopilot in its simplest form acts as a wing leveler. If the aircraft is held at some attitude other than wing level, additional control systems must be used to control the aircraft's sideslip and pitch rate, so that a coordinated turning motion is produced. If a means of varying the roll reference is provided, the aircraft may be steered in any direction by a single control. This mode can be provided as the inner loops for other automatic pilots that allow an aircraft to fly on a fixed compass heading or follow a radio navigation beam in the presence of cross-winds (Stevens and Lewis, 2003).

The Roll-Angle Hold controller architecture as shown in Fig. 5 is composed for p and ϕ feedbacks, together with a dynamic compensator G_{c_ϕ}. The gain K_p equal 0.0876 was determined by applying the root locus method, and the gain was chosen as a function of the values obtained corresponding to the damping ratio and the natural frequency not damped for the dutch roll mode . The roll angle is feedback, where a PI (proportional-integral) controller was added to eliminate the regime error. The gain K_ϕ of the feedback is unitary and the initial tuning for controller gains was done using the Ziegler and Nichols (1942) method. Due to the method provide approximate gains, the response of the system with the determined parameters presented a very high overshoot. In this way, the parameters K_p and T_i of the controller were adjusted with the respective gains (-0.1284, 2.2170) to obtain a response with small overshoot.

The closed-loop transfer function that relates the roll angle (ϕ) and the input reference (ϕ_r) is described by Eq.(21).

$$\frac{\phi(s)}{\phi_r(s)} = \frac{47.392(s+0.4511)(s^2+4.697s+41.34)}{(s+5.977)(s+2.837)(s+0.6541)(s^2+8.349s+79.66)} \tag{21}$$

The closed loop system presented a satisfactory response as shown in Fig.6. For the reference value of the roll angle (ϕ_r) a step with amplitude of 30 degrees was applied at the instant of time equal to 20 seconds.

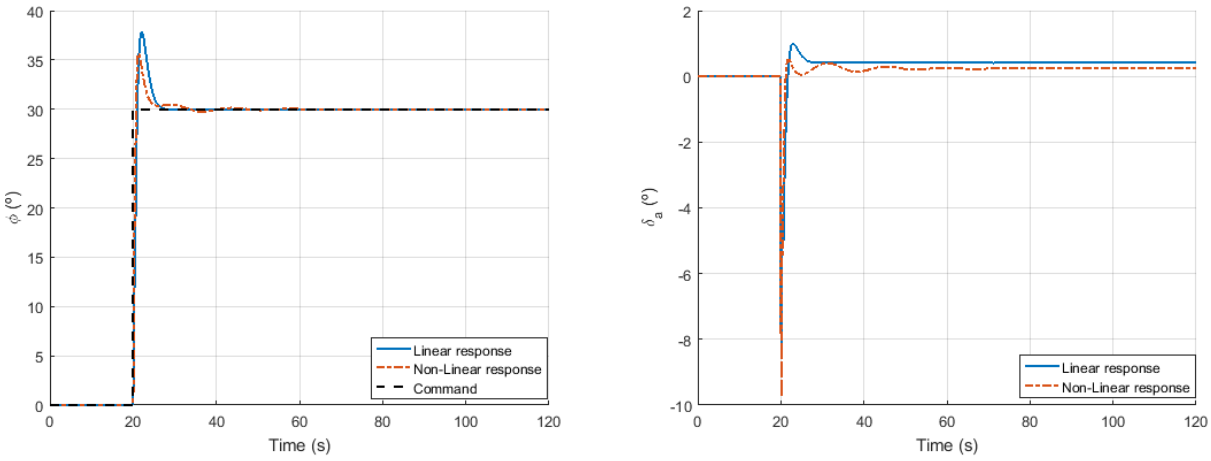
For the response in the time shown by Fig.6a the settling time of ϕ is about 5.34 seconds with an overshoot of approximately 20%. The Figure 6b shows that to produce the expected bank angle response an approximate deflection of 8 degrees on the control surface (δ_a) is required.

3.2.2 Yaw Damper and Turn Compensation

The Heading Hold autopilot control external loop is a change from the conventional autopilot proposed by Stevens and Lewis (2003), by feedback the yaw angle (ψ) and including a turn compensation to generate the roll angle reference for the roll angle hold autopilot described above.

The feedback of the angular rate r, functions as a yaw damper, the purpose of which is to use the rudder to generate a yaw moment that opposes any angular rate from the dutch roll mode. However, for this implementation of the yaw damper there is a problem, since there are several flight modes that require a constant yaw rate, such as coordinate turn.

As a solution to the problem we used a control loop with feedback of yaw damper transient, in which the feedback signal is differentiated so that it disappears during the steady-state conditions. The approximate differentiation can be performed with a simple order high-pass filter, known as washout filter, with transfer function according to Eq.(22).



(a) Roll angle response in the time domain (b) Ailerons control surfaces response in the time domain

Figure 6: Response of the Aircraft for a Reference in the Roll Angle

$$G_w = \frac{s\tau_{\mathrm{w}}}{1 + s\tau_{\mathrm{w}}} \tag{22}$$

Where the time constant τ_{w}, should be chosen according to the compromise relationship between a good damping for the dutch roll mode and a satisfactory turn input. Typically this time constant is in the range of 0.5 to 1 second. The value used in this project was τ_{w} equal 1 second. The gain K_r equal 0.021 was determined using the root locus method so that during a transient maneuver the yaw rate feedback loop works to increase the damping ratio of the dutch roll mode.

A coordinate turn is defined as the null lateral acceleration of the aircraft in its center of gravity. In a coordinate turn, the aircraft maintains the same attitude of inclination and bank angle relative to the reference coordinate system, but its direction changes continuously at a speed of observation. Therefore, the Euler angle rates $\dot{\psi}$ and $\dot{\theta}$ are practically null. Equation (23) provides a “turn compensation" that allows the aircraft to be maneuvered through coordinate turns by applying commands to the roll angle control system. When the aircraft is tilted to the angle ϕ, so that the vector sum of (mg e $m\, U_0\, \dot{\phi}$) is along the z-axis of the aircraft, the desired roll angle depending on the yaw rate.

$$\phi_r = \frac{U_0}{g}\, \dot{\psi} = \frac{U_0}{\tau\, g}\, (\psi_r - \psi) \tag{23}$$

Since ϕ_r tends to be a noisy signal, then it is necessary to generate a smoother signal by filtering it. Considering that the reference yaw angle (ψ_r) is known, and that the yaw angle (ψ) will track (ψ_r) relatively slowly, a low pass filter eliminating the higher frequency noise was added to the dynamics presented by Eq.(23). Where, the time constant τ depends on the aircraft and its design requirements for lateral-directional flight.

4. RESULTS AND DISCUSSION

In this section the developed autopilot composed of the control, navigation and guidance subsystems will be presented. The autopilot will be simulated with the Piper J-3 Cub 1/4 nonlinear plant developed in Matlab-Simulink environment. Finally, the results obtained in Hardware-In-the-Loop will be presented, with the control subsystem embedded in a microcontroller and the nonlinear plant along with the guidance and navigation in Matlab-Simulink environment.

As already described in this paper, a typical autopilot consists of three main subsystems: control, navigation and guidance. The autopilot is simulated with the aircraft block, where non-linear equations describing the dynamics of motion of the aircraft are implemented, scope of the Section 2. The implementation of the control subsystem design has been detailed in Section 3.

4.1 Hardware-In-the-Loop Plataform

In this section, the Hardware-In-the-Loop platform, developed to validate and test the automatic control designed in this paper, will be presented. In particular, hardware-in-the-loop (HIL) simulation tests should be used whenever possible to validate hardware and software under realistic conditions. HIL can be defined as a proposed test platform to provide an environment in which similar models can be applied to the actual aircraft on the designed autopilot. One of the plataform features is the flexibility to immediately implement and check the results (Ribeiro and Oliveira, 2010).

In the built platform, it is proposed that the control to be validate is embedded into the ARM Cortex M3 v7M microcontroller, which communicates directly with a computer, where the non-linear model of the aircraft is implemented along with the navigation and guidance algorithms. Communication between the two systems is done through the Ethernet interface of the microcontroller via User Datagram Protocol (UDP). Figure 7 shows the autopilot block diagram with the implemented subsystems. The guidance and navigation subsystems, as well as the nonlinear model of the aircraft, are simulated in the computer and the control subsystem embedded in the microcontroller.

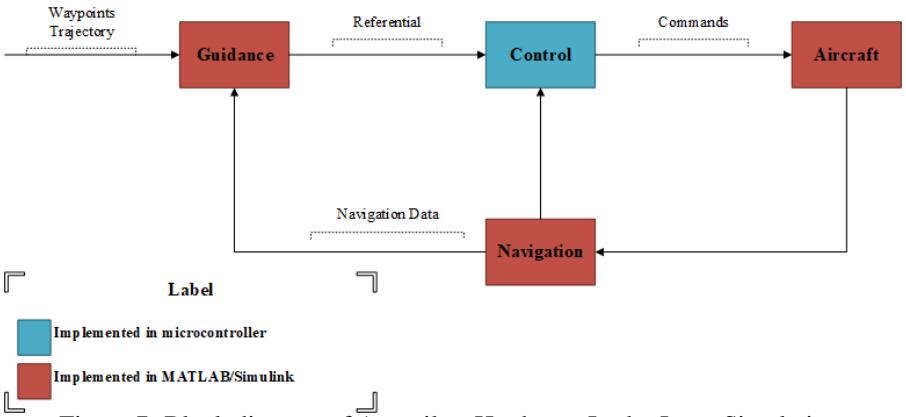


Figure 7: Block diagram of Autopilot: Hardware-In-the-Loop Simulation.

The guidance subsystem is generally responsible for generating the references to the control subsystem from a sequence of waypoints provided by the user. These references are calculated based on the trajectory to be tracked and the aircraft information, coming from the navigation subsystem, allowing the aircraft to maneuver in a consistent way in order to perform a predefined mission.

The implemented guidance algorithm is responsible for monitoring the aircraft position and determining, from a database of waypoints, how to steer the airplane to the desired point, generating references to the yaw angles, aerodynamic velocity and altitude. The guidance algorithm is also responsible for the switching of the predefined waypoints and the calculation of the reference directional angle known as Line of Sight (LOS), obtained according to Bittar (2012).

The navigation is a subsystem capable of providing orientation, position and velocity of an aircraft relative to a coordinate system. As the study of the navigation subsystem is not the scope of this paper, for the sake of simplification, this subsystem will be considered initially as ideal, in other words, the aircraft outputs are obtained without considering some errors, usually found in practice. The implemented navigation system is responsible for obtaining the outputs of the aircraft through the output vector ($\mathbf{y}$) of the aircraft block. The output equations of nonlinear model that characterize the aicraft dynamics of motion will be used regardless of the errors inherent in the reading of this type of sensors.

The controller embedded into the microcontroller receives as input the aircraft navigation data and the reference inputs calculated by the guidance algorithm. Once the calculations are performed, the controller outputs the command deflections (δ_T, δ_e, δ_a, δ_r) that control the aircraft, represented by the nonlinear model.

Before embedding the control, it is necessary to preliminarily discrete control laws, since the microcontroller works with a sampling time and, therefore, does not perform continuous calculations. Thus, in order to obtain a real-time response fast enough to act on the aircraft model, and at the same time respect the sampling limitations of the microcontroller, the sampling time of the microcontroller should be synchronized with Simulink, which also works with a fixed sampling time to solve the nonlinear equations of the mathematical model.

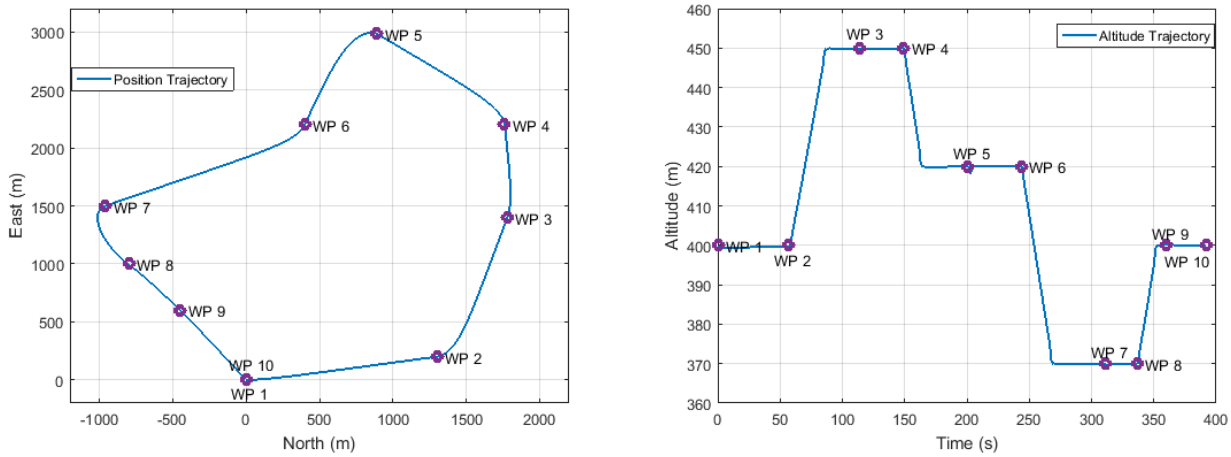
4.2 Mission Simulation

In this section, the simulations will be presented with complete autopilot in a computational environment. The waypoints to be reached by the aircraft controlled by the AP designed is presented in the $[\text{North East Down VT}]^{\text{T}}$ waypoint vector of Eq.(24).

$$\mathbf{waypoints} = \begin{bmatrix} 0 & 1300 & 1780 & 1760 & 890 & 400 & -960 & -800 & -450 & 0 \\ 0 & 200 & 1400 & 2200 & 2984 & 2200 & 1500 & 1000 & 600 & 0 \\ 400 & 400 & 450 & 450 & 420 & 420 & 370 & 370 & 400 & 400 \\ 23 & 23 & 23 & 23 & 23 & 23 & 23 & 23 & 23 & 23 \end{bmatrix}^{\text{T}} \tag{24}$$

The autopilot runs in Simulink environment, using the ODE4 simulation algorithm (Runge-Kutta method), with sampling period of 0.01 seconds. The Runge-Kutta method was adopted by the complexity of the differential equations to be solved and by the ease of its use on the Matlab-Simulink environment.

The system is simulated to execute the presented trajectory, in ideal conditions. By ideal conditions, it is considered: that the atmospheric model is simplified, that is, without bursts of wind in the three directions; there are no variations in the plant, that is, the mathematical modeling and the inertial and geometric parameters of the aircraft do not present errors; and there is no measurement noise, i.e., the sensors do not exhibit the typical errors mentioned above. For these conditions, the the controlled flight result is shown by Fig. 8.



(a) North and East [2-Dimensional] trajectory response

(b) Altitude response

Figure 8: Traveled trajectory

Figure 8a shows the path traveled to reach the waypoints predetermined by Eq.(24) is visualized in two dimensions, as a function of North and East. Figure 8b shows the temporal variation of altitude, in which it is possible to perceive that even the aircraft moving away from the equilibrium region the waypoints were reached in a satisfactory way.

These figures allow to conclude that the designed control, inserted in the constructed autopilot, presents satisfactory answers to the ideal conditions, causing the aircraft to reach the waypoints inserted by the user, according to the rules presented in the guidance and navigation block.

5. CONCLUSIONS AND FUTURE WORKS

The design of the control loop developed using the lineaized models and tested in the HIL platform was sucessful. There are some diferences between the responses, as it was expected. However, the proposed mission, follow several predetermined WP was satisfactory accomplished.

As for future works, some improvements are in scope. Firstly, it is to improve the implementation of the Line of Sight (LoS) guidance algorithm, which currently only tracks the waypoint through the yaw angle. A more robust implementation is required, for example using the Dubins Path Algorithm (Beard and McLain, 2012; McLean, 1990), to start projects with path optimization and obstacle traversal. Secondly, the modeling and inclusion of errors inherent to low-cost inertial sensors in the nonlinear simulation model that describe the equations of motion of an aircraft. This allows the implementation and analysis of more efficient attitude determination algorithms such as the direction cosine matrix method (DCMM) implemented in Pereira (2013), or using direct and indirect Kalman filters (Alves dos Santos, 2015).

6. ACKNOWLEDGEMENTS

The autors acknowledge the support of Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and the Instituto Tecnológico de Aeronáutica (ITA) by the necessary support to realize the present work.

7. REFERENCES

Adiprawita, W., Suwandi Ahmad, A. and Sembiring, J., 2007. "Hardware in the loop simulation for simple low cost autonomous uav (unmanned aerial vehicle) autopilot system research and development". In *International Conference on Electrical Engineering and Informatics*. Institut Teknologi Bandung, Indonesia.

Alves dos Santos, L., 2015. *Estudo de Algoritmos de Determinação de Atitutde para Aplicações Utilizando Sensores de Baixo Custo*. Mestrado em engenharia eletrônica e computação, Instituto Tecnológico de Aeronáutica.

Barnhart, R.K., Hoffman, S.B., Marshall, D.B. and Shappee, E., 2011. *Introduction to Unmanned Aircraft Systems*. CRC Press.

Beard, R.W. and McLain, T.W., 2012. *Small unmanned aircraft: Theory and practice*. Princeton University Press.

Bittar, A., 2012. *Arquitetura da Unidade Central de Processamento do Pegasus Autopilot : da concepção à implementação de um sistema de tempo real em Hardware-In-the-Loop*. Mestrado em engenharia eletrônica e computação, Instituto Tecnológico de Aeronáutica.

Du, Y., 2011. *Development of real-time flight control system for low-cost vehicle*. Msc by research thesis, Cranfield University.

Franklin, G.F., Powell, J.D. and Emami-Naeini, A., 2013. *Sistemas de Controle para Engenharia*. Bookman Editora LTDA, Porto Alegre, Brasil, 6th edition.

Jensen, D., Roberts, M., Alifano, J., Szumczyk, T. and Cheung, W., 2004. "Development of polyuav: polytechnic university's unmanned aerial vehicle". *Polytechnic University. AUVSI Undergrad Competition. Brooklyn, NY*.

Jung, D. and Tsiotras, P., 2007. "Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle". *AIAA Infotech at Aerospace, AIAA*, pp. 07–2763.

Kemp, I., 2010. *Unmanned Vehicles Handbook*. The Shephard Press Ltd.

Maciel, B.C.d.O., 2008. *Modelagem e Identificação de Aeronaves via Abordagem de Sistemas Lineares com Parâmetros Variantes*. Ph.D. thesis, Instituto Tecnológico de Aeronáutica.

McLean, D., 1990. "Automatic flight control systems(book)". *Englewood Cliffs, NJ, Prentice Hall, 1990, 606*.

McManus, I.A., Greer, D.G. and Walker, R.A., 2003. "Uav avionics "hardware in the loop" simulator". In *10th Australian International Aerospace Congress*. Brisbane, Queensland, Australia. URL `https://eprints.qut.edu.au/1161/`. (Date last accessed on Sep. 09, 2017).

Nelson, R.C., 1998. *Flight stability and automatic control*. WCB/McGraw Hill, 2nd edition.

Paw, Y.C. and Balas, G.J., 2011. "Development and application of an integrated framework for small uav flight control development". *Mechatronics*, Vol. 21, No. 5, pp. 789–802.

Pereira, M.O., 2013. *Sistema de Navegação e Determinação de Atitude Inercial para Aeronaves Não Tripuladas com Base em Sensores de Baixo Custo*. Mestrado em engenharia eletrônica e computação, Instituto Tecnológico de Aeronáutica.

Ribeiro, L.R., 2011. *Plataforma de Teste para Sistemas de Piloto Automático utilizando Matlab/Simulink e Simulador de Vôo X-Plane*. Mestrado em engenharia eletrônica e computação, Instituto Tecnológico de Aeronáutica.

Ribeiro, L.R. and Oliveira, N.M.F., 2010. "Uav autopilot controllers test platform using matlab/simulink and x-plane". In *Frontiers in Education Conference (FIE), 2010 IEEE*. IEEE, pp. S2H–1.

Stevens, B.L. and Lewis, F.L., 2003. *Aircraft Control and Simulation*. John Wiley & Sons, New Jersey, EUA, 2nd edition.

Ziegler, J.G. and Nichols, N.B., 1942. "Optimum settings for automatic controllers". *trans. ASME*, Vol. 64, No. 11.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.