

Computational Simulation of flight behavior of Flying Wing UAV

Fernanda Cavalcante da Silva¹, Marco A. V. M. Grinet², Helosman Valente Figueiredo³

Abstract—The aim of this work is to model and simulate a Flying Wing Aircraft throughout a combination of two software and an automated system to compare the flight data acquired from two different simulations. The framework proposed constitutes of the development of both graphical and mathematical model of the aircraft aimed at flight data collection. In order to implement the Software-in-the-Loop (SIL), we used a commercial flight simulation environment, X-plane 10, which simulates the dynamics of the aircraft through the Blade Element Theory, and the software MATLAB/Simulink, which simulates and implement the control laws. Data transfer between X-Plane and Simulink was made possible through UDP (User Datagram Protocol), which enables process-to-process communication and works in conjunction with higher level protocols to help manage data transmission services. We collected the flight data from the simulator, as well as the responses of the mathematical model based on equations of motion and the aerodynamic derivatives, which were computed via the Digital DATCOM software. Both analytical simulation and the SIL simulation were performed with the same flight conditions, validating the longitudinal dynamic of the computational model. This framework allows further studies on aircraft simulation to be developed, reinforcing the collection, storage and processing of flight data.

Index Terms—UAV, Software-in-the-Loop, X-Plane, MATLAB/Simulink, Flying Wing

I. INTRODUCTION

IN the recent years, computational simulation has been an important tool for data acquisition, and in providing essential information about the behavior of mechanical systems [1]. In the aerospace industry, simulations are used throughout the development of all aircraft, and they evaluate the control algorithm allowing for easy manipulation of the early model. This allows for data creation for the aircraft model, for faster development of the product, as well as minimizing the number of experimental flights. Thus, reducing the number of persons involved and consequently the final cost of the project [2]. However, it is necessary to develop more reliable models of aircraft.

Several commercial simulators are available and used as a tool, known as Software-In-the-Loop (SIL), for the

implementation of flight dynamics, navigation control, and for validating models in a precise fashion before field tests [2]. SIL couples partially integrated software with an environment simulation and allows a direct information-technical communication between the two, for simple data creation, collection, and processing.

This also offers the possibility to execute tests before the hardware is available. Despite simulation-based testing being a very important part of the product development, Hardware-In-the-Loop (HIL) simulation tests should be adopted whenever it is possible in order to validate together both the hardware and the software under realistic conditions [3].

In the specific case of Unmanned Aerial Vehicles (UAV) the mathematical model, as well as the implementation of SIL and HIL were describe by [4]. It not only describes the physics of the system but also the behavior of the low-level autopilot, and the state estimation routines. UAVs are of high interest for prospective military and civilian applications due to their several advantages such as maneuverability, and lack of need for direct human interaction [5]. The literature shows several works of multicopter aircraft modeling [6] [7]. However, more detailed modeling of flying wings and their integration with SIL and HIL are still a growing field of high interest [8]. Therefore, the literature does not provide many resources on this type of aircraft.

Since there is a lack of mathematical modeling that describes the behavior of flying wings because of their design complexity and aerodynamic instability, this paper presents the longitudinal dynamic modeling of a flying wing UAV through the derivatives of stability and control as well as the equations of motion, and through the Blade Element Theory. The mathematical model describes the system by its aerodynamic derivatives and equations of motion that establish the relationship among a set of variables, which provides an explicit expression of the systems behavior pattern and how it would work before building it. The behavior of each variable shows immediately in the equation, which in return allows the system to be analyzed and optimized. Whereas, through simulation modeling, it creates and analyses a digital prototype of a physical model to predict its performance in the real world with environment effects on it.

*This work was supported by Laboratório de Robótica e Veículos Autônomos - LRVA

¹Fernanda Cavalcante da Silva at the Universidade do Vale do Paraíba - UNIVAP, Division of Aerospace Engineering, Avenue Shishima Hifumi, 2911 Urbanova, CEP: 12244-000, São José dos Campos-SP, Brasil. (e-mail: fcalsilva5@gmail.com).

²Marco Antonio Vieira Macedo Grinet at the Universidade Federal de São Paulo - UNIFESP, Institute of Science and Technology, Laboratory of Signal

and Image Processing, São José dos Campos – SP, Brazil. (e-mail: mgrinet1@gmail.com).

³Helosman Figueiredo at the Instituto Tecnológico 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. (e-mail: helosman@gmail.com).

Digital Object Identifier: 10.14209/jcis.2019.3

The purpose of the present work is to, first, mathematically and graphically model the flying wing. Second, to simulate both models aiming to compare and review the dynamic response from the introduction of longitudinal oscillations during cruise flight, using the commercial flight simulator X-Plane (Laminar Research©) and the mathematical software MATLAB/Simulink (MathWorks©). Finally, perform several flight simulation tests in order to apply refined adjustments to the aircraft parameters and to improve the control of flight stability. The results of both simulation methods were analyzed and compared. This model will allow the development of future projects from what has already been accomplished in this, thus helping the advance of new research with flying wing flight data creation, its software data storage and even a tool for embedded systems.

This paper is organized as follows: Section II introduces the software X-Plane and its Plane-Maker platform, and then the procedure used for the modeling through the Blade Element Theory. Section III describes the UDP communication between the X-Plane and MATLAB/Simulink, the necessary configuration for the interface, and the implementation of the control system through Simulink for the simulations of the UAV. Section IV demonstrates mathematically the flight dynamics of a fixed wing UAV through the derivatives of stability and control. Section V presents the results obtained and discussions. The last section concludes the importance of this work and presents the research that still needs to be done as future work.

II. PLANE-MAKER MODELING

The flying wing was modeled using the platform Plane-Maker, built into X-Plane 10, which enables the creation of any aircraft. With its graphical interface, the user can have a visual feedback of the in-flight behavior of the aircraft once all the physical characteristics are applied to the model. X-Plane 10 is a realistic flight simulator certified by the Federal Aviation Administration (FAA), and a viable tool due to its aircraft creation environment and its ability to simulate the aircraft dynamics. Through the combination of both these assets, we obtained the graphical model of the UAV.

TABLE I
FLYING WING CHARACTERISTICS LIST

Properties	Value	Units
Total Mass (payload + batteries) (m)	0.652	[kg]
Aerodynamic Reference Area (S)	0.35	[m ²]
Wing Sweep Angle (Λ)	31	[deg]
Root Aerodynamic Chord (\bar{c})	0.31	[m]
Tip Aerodynamic Chord (\bar{c})	0.2	[m]
Lateral Reference Length (b)	1.20	[m]
Moment of Inertia (I_{yy})	0.0113	[kg/m ²]
Airspeed (V_T)	40	[m/s]
Flight path angle (γ)	0	[deg]
Altitude (h)	150	[m]
Wing Airfoil	MH-45	

The UAV chosen for this project was a commercial RC Flying Wing (Fig. 1), and its characteristics are shown below in Table I. This specific model did not exist in the X-Plane database; therefore, we modeled it from scratch (Fig. 2). The dimensioning of the reference model is very important because X-Plane separates the aircraft into small sections to calculate several aerodynamic factors on each, ensuring that the entire aircraft is being computed. This methodology is based on a theory called *Blade Element*, which is explained by [6]. The model was created in the Plane-Maker according to the specifications of the actual model and following the guidelines recommended by the platform's manual [9].

The UAV is powered by a brushless electric motor and its parameters of the thrust and power are approximated according to the manufacturer's datasheet [EMAX 2822 Brushless Motors datasheet]. Because X-Plane is an aircraft flight simulator on real scales, there is a difficulty in making the reaction time of the engines of a small UAV compatible with the real model. Several in-flight tests were performed to fine-tune the parameters, specifically in the engine and CG position. The suitable engine specifications are:

- 1) *Maximum allowable Power: 166W;*
- 2) *Engine RPM: 12000;*
- 3) *Propeller: 10x5;*

The static thrust of the propeller was computed via the program JavaProp. This feature describes the geometry of the airfoil to be tested, assigns the appropriate Reynolds number, generates the aerodynamic coefficients, and therefore thrust. A file is generated and then exported to Plane-Maker.



Fig. 1. RC model Flying Wing.

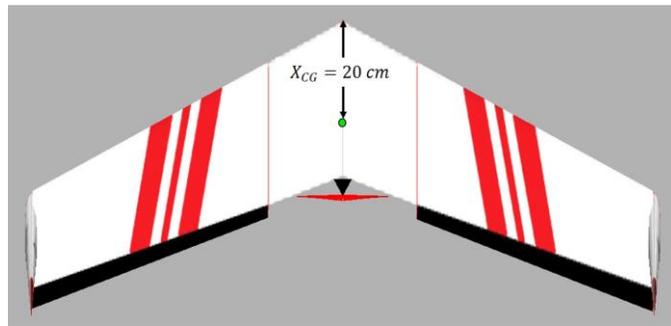


Fig. 2. RC model Flying Wing modeled on Plane-Maker.

III. SOFTWARE-IN-THE-LOOP IMPLEMENTATION

MATLAB/Simulink is an environment for simulation of dynamic systems. It provides a customizable set of block libraries that enables the user to design, simulate, and test a variety of systems time-varying. This platform presents ready blocks for performing communications with external environments, allowing data exchange. Therefore, a block diagram was created in Simulink aiming to send flight control to the model built in the platform X-Plane. The controls details for the architecture used for this project can be found at [10].

UDP is a transport layer protocol, where each output operation of a process produces exactly one UDP datagram, causing one *Internet Protocol (IP)* datagram to be sent, an in-depth description can be found in [6]. The simulation constitutes of two computers that use an Ethernet network with IP addresses defined for each, which allows data exchange between MATLAB/Simulink and X-Plane, as illustrated in Fig. 3. Both software were configured for sending and receiving packets and data, following the guidance at [2].

The MATLAB/Simulink PC generates the error signal through a reference signal and feedback data provided by X-Plane. Then, it uses the error signal as input to the control law, thus generating a command signal for the motors that will be sent back to X-Plane. Finally, X-Plane PC receives the command data for the engine, executes the interactions and sends the new data position to the MATLAB/Simulink PC again [11]. This communication procedure is explained in detail by [12] [13].

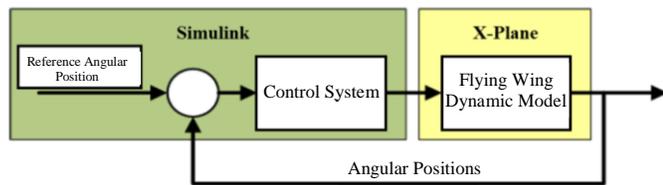


Fig. 3. Simulation loop. Modified from [11].

A. X-Plane Settings

After modeling the UAV in Plane-Maker, an extension *.acf file is generated, which is loaded into X-Plane to perform in-flight simulations. An adjustment on the number of flight-model per frame was necessary in order to establish the amount of time that the flight simulator calculates the forces in each frame. A value between 6 and 10 is recommended to ensure a better performance of the model during the simulations [2].

For the analysis of the simulations, obtaining the flight data is required. X-Plane has several ways to view this data, it can be displayed on the main screen of the simulator and sent via Ethernet using UDP. The configuration can be done in *Setting / Data Input Output*. For this work, the following output data was configured:

- 1) Angular position (ϕ, θ, ψ);
- 2) Scalar position (x, y, z);
- 3) Angular velocity (Q, P, R);
- 4) Scalar velocity (V_x, V_y, V_z).

Another important adjustment is the UDP rate/s and disk rate/s values to match the response frequency of the X-Plane with MATLAB/Simulink, which was set as 60.00 for both.

Each output generates a set of nine groups of 4 bytes each [14]. The first four bytes represent the type of packet, the fifth byte is an internal policy, and the next group of four bytes represent the parameter label that is being sent as shown in Fig. 4. For each label, X-Plane sends a packet with eight data and all of them take over the format of single-precision floating point of 32 bits [3] [6].

To send commands to X-Plane, the packet must be built with the same architecture.

	DATA				I	Label				Group 1				...	Group 8						
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	37	38	39	40

Fig. 4. Data packet from [6].

B. Control System

After modeling and adjusting all parameters of the Flying wing in Plane-Maker, following the same dimensions, weight and CG of the real aircraft, we implemented the control laws using the software MATLAB/Simulink, in order to test control algorithms and navigation of the aircraft.

The Flying Wing is a tailless fixed-wing aircraft that has no definite fuselage. Due to the lack of conventional stabilizing surfaces and the associated control surfaces, this type of aircraft suffers from the inherent disadvantages of being dynamically unstable and difficult to control. This means that the amplitude of any oscillatory motions induced by disturbances eventually increase with time infinitely relative to a steady-state flight condition as shown in Fig.5.

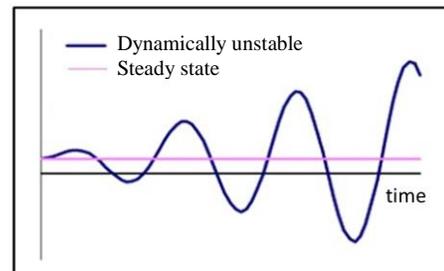


Fig. 5. Graphical example of dynamically unstable aircraft motion from [15].

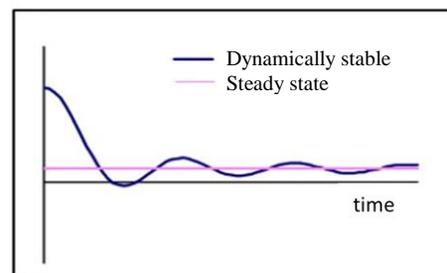


Fig. 6. Graphical example of dynamically stable aircraft motion from [15].

Dynamic instabilities can be tolerated by a human pilot or by automatic controls for feedback controllers, which is known as “closed-loop stability”. Therefore, to perform the flight tests for both analytical and graphical methods, a PID (Proportional-

Integral-Derivative) Controller system for the control loops of pitch and roll was developed in Simulink as shown in Fig. 7. It allowed us to perform the simulations at a constant velocity, oscillating only the pitch attitude degrees of freedom.

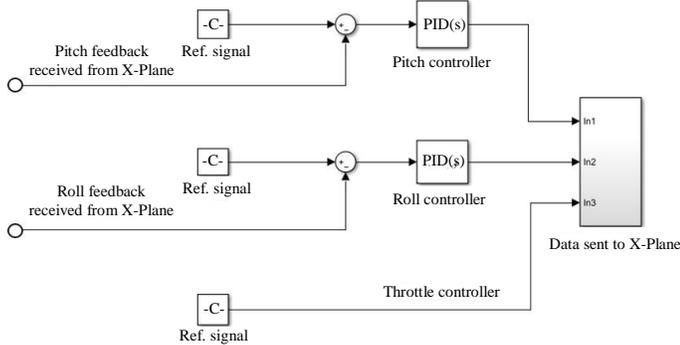


Fig. 7. PID Simulink Architecture.

C. Simulink Settings

In order to use the data received by X-Plane in the control and guidance laws implemented in Simulink, the packets need to be broken in variables. According to [6], this interface has five main blocks as illustrated below in Fig. 8. The blocks “Receiving data” and “Sending data” are responsible for closing the system control loop, and the blocks Graphical Interface and the Simulation Analyze act on every interface.

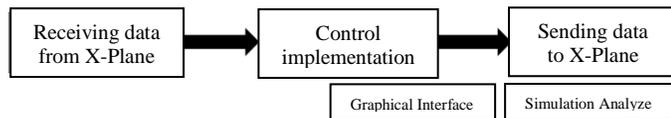


Fig. 8. MATLAB/Simulink Interface from [6].

Simulink presents some blocks that send and receive data through the UDP protocol. The first block is the UDP receiver, it receives the packets sent from X-plane, it is necessary to configure it according to the guidance at [2]. An unpacking block must be provided to extract the data, and the Byte Reversal to adapt the byte because X-Plane exports the data in big-endian format where the most significant one is the first byte, whereas MATLAB work in little-endian format, which the most significant byte comes last. Then, the control laws for a fixed wing aircraft are implemented through Simulink. The code interprets the UDP data. Fig. 9 illustrates the flowchart of the implemented program. And at the end the data is shown on the output displays.

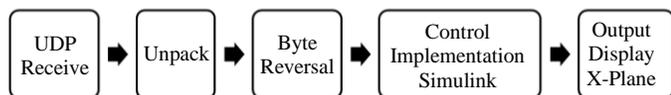


Fig. 9. Flowchart Receiving Data.

The UDP Sending is similar to the receiving process. The data packet must be mounted with the same architecture that X-Plane sends commands as illustrated in Fig. 4.

IV. MATHEMATICAL MODELING

The dynamics for fixed-wing aircraft can be approximately decomposed into longitudinal motion. Aiming to obtain the mathematical model that describes the longitudinal dynamics, a series of considerations are made to obtain the equations, some are described below:

- 1) The Earth is the inertial reference system;
- 2) The aircraft is a rigid body and its weight is a constant;
- 3) Acceleration of gravity does not change with flight altitude;
- 4) The Reynolds number and Mach number effects are approximately constant;
- 5) The Sideslip Angle (β) is considered null;
- 6) The disturbances around the equilibrium are small, with small variations to the pitch angle (θ);
- 7) The deflection of the elevator does not alter any force, only the moment of pitch.

An aircraft usually has 6 degrees of freedom (6DoF), which can be described by equations of forces and moments acting on it and presents nonlinear dynamics [16]. Using these nonlinear equations of the longitudinal dynamics, and the calculation of the forces and moments through the stability and control derivatives we can finally simulate the flight of the UAV.

A. Stability and control derivatives via USAF DATCOM

The stability and control derivatives determine the static and dynamic stability of the aircraft. The static stability refers to the direction of aerodynamic moments as the aircraft oscillates its nominal flight condition, where the tendency of restoring the initial position defines if the aircraft is statically stable. The dynamic stability refers to the dynamic behavior of the airframe in response to disturbances, where the aircraft is defined as dynamically stable if its response dampens the disturbance out over time as shown in Fig. 6.

The stability and control derivatives can be estimated or identified through wind tunnels tests or via flight tests [10]. According to the literature, the most feasible method for obtaining these derivatives in cruise flight conditions is through calculations based on empirical data since wind tunnels is not always accessible.

The Data Compendium (DATCOM) has been widely used for the estimation of the aircraft derivatives. It was developed by the United States Air Force (USAF) and is based on empirical data (wind tunnel data history and in-flight testing of certain types of airfoils and fuselages) and simplified analytical methods, this method is known as semi-empirical. Given an aircraft configuration, declared in FORTRAN language, DATCOM easily provides the estimation of aerodynamic coefficients and their stability derivatives.

All the models parameters and specifications for the analytical modeling were the same as those used for modeling in Plane-Maker. The model was created in DATCOM following the guidelines recommended by the manual [17]. Table II contains the numerical values of the longitudinal aerodynamic derivatives of the flying wing. $C_{M\alpha}$ is referred to as the longitudinal static stability derivative, C_{Mq} is referred to as the

pitch damping derivative, and $C_{M\delta_{elev}}$ is associated with the deflection of the control surface elevator.

TABLE II
DIMENSIONLESS LONGITUDINAL AERODYNAMIC DERIVATIVES GENERATED BY DATCOM.

C_{D0}	C_{L0}	C_{M0}	$C_{L\alpha}$	$C_{M\alpha}$	$C_{M\dot{\alpha}}$	C_{Mq}	$C_{M\delta_{elev}}$
0.0307	0.508	-0.219	4.235	0.132	0	0.691	-0.256

B. Aerodynamic coefficients for forces and moment

Aerodynamic forces and moments have a complex dependence on several variables, resulting in a complicated nonlinear correlation between each variable [10]. These variables, known as aerodynamic coefficients, are dimensionless and depend on the derivatives of stability and control for a flight condition, and are described below:

1) Lift Coefficient:

$$C_L = C_{L0} + C_{L\alpha} \cdot \alpha \quad (1)$$

2) Drag Coefficient:

$$C_D = C_{D0} + C_{DCL} \cdot C_L^2 \quad (2)$$

3) Pitching Moment Coefficient:

$$C_M = C_{M0} + C_{M\alpha} \cdot \alpha + C_{M\delta_{elev}} \cdot \delta_{elev} \quad (3)$$

Where: δ_{elev} is the elevator control deflection, and α is the attack angle.

C. Aerodynamic Forces and Moment

As the aircraft passes through the air, a pressure distribution, or aerodynamic force, is generated around its body. The air velocity (V_a), air density (ρ) and the shape and attitude of the aircraft are all variables of the distribution of pressure acting on the aircraft. Hence, in order to capture the effect of the pressure with a combination of forces and a moment, the dynamic pressure (\bar{q}) is given by $\frac{1}{2} \rho V_a^2$ and the variables that define the shape of the aircraft body are given by the planform area of the aircraft wing (S), and the mean chord of the wing (\bar{c}).

The aerodynamic forces and moment are obtained from the dimensionless aerodynamic coefficients at a given flight condition as follows:

1) Lift

$$L_{lift} = \bar{q} \cdot S \cdot C_L \quad (4)$$

2) Drag

$$D_{drag} = \bar{q} \cdot S \cdot C_D \quad (5)$$

3) Moment

$$M_{moment} = \bar{q} \cdot S \cdot \bar{c} \cdot C_M \quad (6)$$

A complete modeling of the engine with the propeller and its dynamic thrust are quite complicated and beyond our scope; however, a simplified model for the calculation of the propulsion force is the approximation by given as follow. Assuming that engine provides enough torque to the propeller and the thrust delivered by the propeller is entirely dependent

on the aerodynamics of the propeller, the thrust force can be calculated as described in [10]:

4) Thrust

$$T_{thrust} = \frac{4}{\pi^2} \rho \Omega^2 R_p^4 c_T \quad (7)$$

Therefore, R_p is the propeller radius, Ω is the rotational speed, and c_T is an approximation at a given advance ratio value that is computed from the flight speed and the rotational speed of propeller at each instant. The propeller efficiency considered for the calculations was of 70%.

D. Linear State-space model of the UAV

The equations of motion for an aircraft are a set of 12 nonlinear, coupled, first-order, ordinary differential equations and are describe in [8]. In order to produce reduced-order transfer function and state-space models more feasible for control system design, we linearized and decoupled the equations of motion. The transfer function model for the longitudinal dynamics are organized by the following variables: the pitch angle (θ), the angle of attack (α), the pitch rate (Q), the altitude (h), and the airspeed (V_T). The control signals used to influence the longitudinal dynamics are the elevator ($\delta_{elevator}$) and the throttle ($\delta_{throttle}$). The nonlinear longitudinal equations of the system are described below:

$$\dot{V}_T = \left[\frac{(T_{thrust} \cdot \cos(\alpha) - D_{drag})}{m} \right] - g \cdot \sin(\gamma) \quad (8)$$

$$\dot{\alpha} = \left[\frac{-T_{thrust} \cdot \sin(\alpha) - L_{lift} + m \cdot (V_T \cdot Q + g \cdot \cos(\gamma))}{(m \cdot V_T + \bar{q} \cdot S \cdot C_{L\dot{\alpha}})} \right] \quad (9)$$

$$\dot{\theta} = Q \quad (10)$$

$$\dot{Q} = \frac{(\bar{q} \cdot S \cdot \bar{c} \cdot (C_{Mq} + D) + T_{thrust} \cdot Z_W)}{I_{yy}} \quad (11)$$

where:

$$D = \left[\frac{\frac{1}{2} \cdot \bar{c} \cdot (C_{Mq} \cdot Q + C_{M\dot{\alpha}} \cdot \dot{\alpha})}{V_T} \right] \quad (12)$$

$$\dot{h} = V_T \cdot \sin(\gamma) \quad (13)$$

Considering that the objective of this work is to simulate modes of flight and control in cruise level flight, the state-space model linearized about the trim condition for the longitudinal dynamics is represented as follows. In other words, we solved the non-linear equations of the longitudinal dynamics for null derivatives with respect to the output and input state variables. The vector of states of the longitudinal dynamics (14) is formed by the airspeed, Euler angles, and altitude, which are the variables of interest for the chosen flight conditions. The control input vector (15) is composed of the deflection of the elevator control surface of the UAV plus the level position of the throttle.

$$X^T = [V_T \quad \alpha \quad \theta \quad Q \quad h] \quad (14)$$

$$U^T = [\delta_{elevator} \quad \delta_{throttle}] \quad (15)$$

The Jacobian of equations (8-13) were derived and linearized at trim condition. Utilizing the state-space notation (16), we can rewrite the linearized system in the matrix form as described in section V.

$$\begin{aligned} \dot{X}_L &= A.X_L + B.U_L \\ Y_L &= C.X_L + D.U_L \end{aligned} \tag{16}$$

Where the vector of state and control of the linearized equation are:

$$\begin{aligned} X_L &= X - X_E \\ U_L &= U - U_E \end{aligned} \tag{17}$$

After performing the Jacobian, we obtained the linearized transfer function model and replaced all forces, moment, thrust force, the gravity acceleration (g), and other variables that have been previously defined.

The detailed steps and calculations for the mathematical modeling of the longitudinal dynamics of a fixed wing UAV are described in [5]. The analytical simulation was developed in a MATLAB language code. Through the *datcomimport* function, the stability and control derivatives of the model generated by DATCOM (Table II) are exported to MATLAB and introduced in the calculations of the coefficients and aerodynamic forces and moment. Then, all the values were replaced in the linearized equations of the longitudinal dynamics to obtain the state-space model.

V. RESULTS AND DISCUSSIONS

Based on the real RC Flying Wing specifications, the Flying Wing was successfully modeled in Plane-Maker environment, as illustrated in Fig. 2. Then, flight tests were performed in X-Plane as shown in Fig. 10, using the same flight conditions as on the analytical flight tests.



Fig. 10. Flying Wing model in-flight simulation.

A model is based on approximate representation of some aspect of a real system, rather than exact values, therefore, its predictions are only as accurate as the data and assumptions used in its construction and operation. X-Plane separates the aircraft into small sections to calculate several aerodynamic factors on each, thus, the dimensioning of the reference model was fundamental to ensure that the entire aircraft was being computed.

X-Plane has proven to be a reasonably accurate tool in predicting the flight characteristics of an aircraft if its input parameters are correct. This is seen in the very close correlation of obtained flight test stall speeds to flight manual data, as well

as the correlation of maximum level flight acceleration and speeds to flight manual data.

With the Flying Wing model in flight, it was necessary to adjust the PID Controller, in order to enter the cruise flight phase at a specific velocity and no roll attitude. Thus, once on a cruise flight, the aerodynamic coefficients, propulsive, and aerodynamic forces were obtained and could be compared directly with those calculated by the analytical method. This comparison is shown in Table III. It is important to emphasize that X-Plane considers the wing area as the wing area plus the area of the horizontal stabilizer, differently from the normal way adopted by the aerodynamic concept. Thus, this factor together with the fact that X-Plane also calculates the environment effects on the model, can be the reasons of the differences found.

Some advantages that X-Plane model has over the analytical model were found, such as a digital model may be more easily modified and recalibrated than the analytical model, which will allow increasingly accurate simulations as more data become available; the digital model should provide faster responses to complex maneuvers; and the digital model program has the capability to simulate changes in the aerodynamic performance resulting from environment effects. A series of tests were run on both the analytical and the digital models to compare solutions and to verify that the digital version was accurate and could produce the same results as the mathematical model.

TABLE III
COMPARISON OF FORCES AND AERODYNAMIC COEFFICIENTS FROM BOTH SIMULATIONS

Parameter	Analytical Method	X-Plane Simulation
Lift	2.936 lb	2.849 lb
Drag	0.312 lb	0.337 lb
Lift/Drag	9.410 ratio	8.466 ratio
Thrust	1.136 lb	1.205 lb
C_L	0.508	0.413
C_D	0.046	0.049

The results at trim conditions from both simulations were significantly close, which allow us to conclude that the digital prototype modeled in X-Plane is a reliable model of the Flying Wing aircraft enabling the implementation of HIL simulation as further studies. This method was also proved to be a solid tool for flight data creation, transmission, storage and processing, thus contributing significantly for the aeronautical industry.

The communication and processing flight data between MATLAB/Simulink and X-Plane required a lot of study in communication and information systems and revision of the content available in the literature. One of the biggest challenges was to understand the configuration needed to exchange data between the simulator and the external software (SIL). As well as platform response frequency compatibility for more realistic results. We obtained an acceptable integration between the X-Plane and Simulink platforms through UDP communication using a response frequency of 60.00 rate/s.

$$\dot{X}_L = \begin{bmatrix} 0.0000 & -1.9248 & -0.0098 & 0 \\ 0.0046 & -0.0142 & 0 & 0.0010 \\ 0 & 0 & 0 & 0.0010 \\ 0 & -0.0027 & 0 & 0.0000 \\ 0 & -0.0150 & 0.0150 & 0 \end{bmatrix} \cdot X_L + \begin{bmatrix} 0 & 2.8918 \\ 0 & 2.1948 \\ 0 & 0 \\ -0.0262 & 0 \\ 0 & 0 \end{bmatrix} \cdot U_L \quad (18)$$

The aircraft has unstable modes in the longitudinal degree of freedom. These open loop instabilities make the aircraft challenging to fly manually. Fortunately, since the longitudinal dynamics is controllable, the unstable poles were shifted to the open left side of the s-plane during the design of the controller, ensuring closed-loop stability and enabling flight attitude control. In order to compare the analytical and X-Plane simulations, in-flight tests were performed using controllers to stabilize the unstable dynamic of the model enabling us to control necessary flight condition for the simulations, such as the altitude, velocity and roll attitude.

Therefore, the nonlinear equations of the longitudinal dynamic (8-13) were linearized at trim condition, as described below in S.I. units (19). From this, it was possible to rewrite the system in state-space notation (18). Thus, the control project was designed and optimized.

$$X^T = [40 \ 4.8000 \ 4.8000 \ 0 \ 150] \quad (19)$$

$$U^T = [-2.7300 \ -0.0705]$$

The analytical simulation of the longitudinal dynamics of the aircraft when stimulated by unitary step showed a difference in the percentage of overshoot to the different amplitudes, according to Table IV. Whereas in X-Plane simulation the overshoot tends to decrease with the increase in amplitudes variation. The time until steady-state for both methods are relatively equal.

TABLE IV
ANALYSIS OF STEP RESPONSE OF DIFFERENT AMPLITUDES SIMULATING PITCH ANGLE VARIATION. RESULTS OF ANALYTICAL AND X-PLANE SIMULATIONS.

Method	Amplitude	Overshoot (%)	Time until steady-state (s)
Analytical	5	14	7.5
	7	14.3	7
	10	15	7.5
X-Plane	5	40	8
	7	35.7	9
	10	30	8

Fig. 11 illustrates the comparison of the simulations described in Table IV, where the blue line represents the analytical response and the red line represents the X-Plane response. Note that the analytical method has smaller overshoots before stabilizing when compared with X-Plane method, which besides the overshoot it has also a small undershoot before stabilizing. The difference between the simulations can be justified by the different algorithms used by the platforms, being the MATLAB a purely numerical simulation, whereas X-Plane, being a flight simulator, that simulates the dynamics of the model in a more realistic way; as well as the difficulties in achieving perfect cruise flight at a constant speed in flight simulations.

CONCLUSION AND FUTURE WORK

Both platforms used for this project are great tools for simulation of early aircraft models to obtain flight data in different conditions, facilitating a prediction of its behavior and performance during the flight before manufacturing a prototype. The flight simulator X-Plane along with MATLAB/Simulink have shown high confidence level and reliability in modeling and simulating, providing strong expandability for flight data creation, collection and processing, and wide application in aircraft project development.

The mathematical model and its analytical simulations have validated the digital model, showing it is consistent and reliable. The model created in Plane-Maker opens up numerous possibilities for applications, enabling in depth studies of Flying Wings Aircraft design, also the testing of several techniques for control and navigation algorithms. Additionally, this simulation platform is a great tool for data generation for embedded systems, software data processing, and storage and simulation of flight recorders.

Developing both a simulation and an analytical model is a powerful method of getting answers to different sets of questions most effectively through one approach vs. the other.

The next development step, which will be tackled in future work, is the Hardware In The Loop (HIL) simulation and other control laws, which in turn, would receive the state variables generated from the simulator and calculate the control signals injected into the simulated plant at different flight conditions. The HIL simulator system simulates the aircraft characteristics, sensor and actuator modeling while communicating with the autopilot hardware.

The acquisition of data from the modeling of an aircraft, and its flight simulation as well as its data processing, could lead to a disruptive approach in aircraft design, potentially reducing time and costs necessary to manufacture a final prototype.

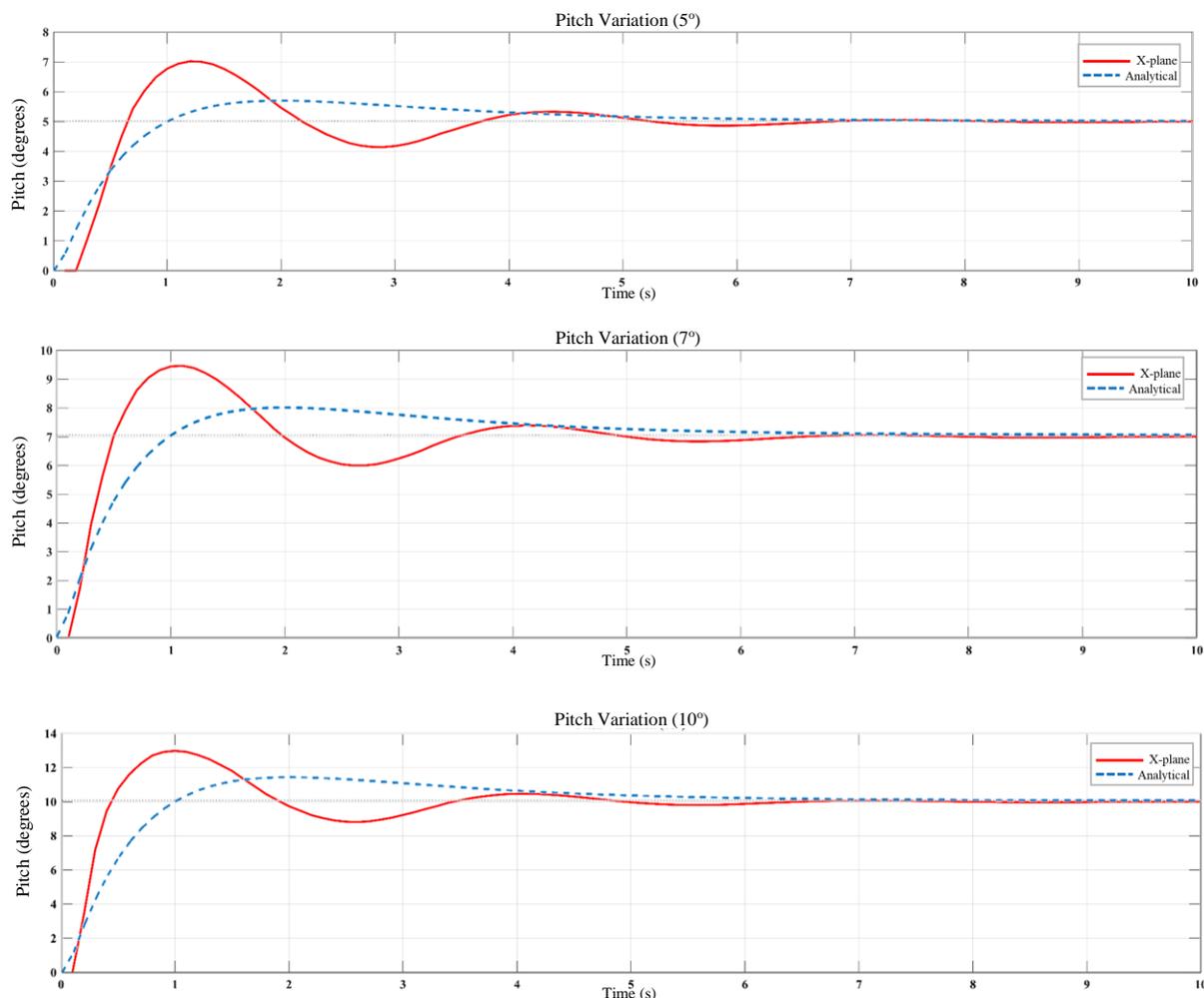


Fig. 11. Attitude response for comparison between analytical model and X-plane simulation. Results shown for 5°, 7°, 10° pitch angle variation.

REFERENCES

- [1] SKIRA, Charles A., "Reducing the cost of advanced technology engines", AIAA 97-3156, 33rd Joint Propulsion Conference, Seattle, WA, doi: 10.2514/6.1997-3156
- [2] BITTAR, A; FIGUEIREDO, H. V; GUIMARAES, P. A; MENDES, A.C. "Guidance Software-In-The-Loop simulation using X-Plane and Simulink for UAVS." International Conference on Unmanned Aircraft Systems (ICUAS) May 27-30, 2014. Orlando, FL, USA, doi: 10.1109/ICUAS.2014.6842350
- [3] SANTOS, S. R. B; JUNIOR, S. N. G; JUNIOR, C. L. N; BITTAR, A; OLIVEIRA, N. M. F. "Modeling of a Hardware-In-The-Loop Simulator for UAV autopilot controllers." 21st Brazilian Congress of Mechanical Engineering October 24-28, 2011, Natal, RN, Brazil, doi: 10.1007/s10846-013-9905-8
- [4] MARKIN, S. "Multiple Simultaneous Specification Attitude Control of a Mini Flying-Wing Unmanned Aerial Vehicle." Thesis for the degree of Master of Applied Science. University of Toronto.
- [5] SILVA, W. R. "Modelagem, simulação e controle de um veículo aéreo não tripulado (VANT) de asa fixa." Thesis for the degree of Automation Engineering, Universidade Federal De Santa Maria, Santa Maria - RS, dez 2015.
- [6] FIGUEIREDO, H. V. "Simulation Platform for Quadricopter: Using MATLAB/Simulink and X-Plane." 2012. Thesis for the degree of Automation Engineering - Department of graduate studies of the Aeronautics, Institute of Technology, Instituto Tecnológico de Aeronáutica – ITA, São Jose dos Campos, SP - Brazil.
- [7] HENZ, G. "Modelagem e desenvolvimento de controlador de voo de um veículo quadricopter." Thesis for the degree of Automation Engineering, Universidade Federal de Santa Maria, Santa Maria - RS, dez 2015.
- [8] BEARD, R.; MCLAIN, T. "Small Unmanned Aircraft: theory and practice." Princeton University Press, 41 William Street, Princeton, New Jersey 08540, doi: 10.2514/1.61067
- [9] *Plane Maker App Manual*. Laminar Research, 2013.
- [10] JUNG, D & TSIOTRAS, Panagiotis. (2007). Modeling and hardware-in-The-loop simulation for a small unmanned aerial vehicle. AIAA Infotech at Aerospace AIAA. 07-2763.
- [11] FIGUEIREDO, H. V; BITTAR, A. "Platform for Quadricopter: Analysis and Applications." International Conference on Unmanned Aircraft Systems (ICUAS). Orlando, USA - May, 2014, doi: 10.1109/ICUAS.2014.6842332
- [12] L. R. Ribeiro, "Plataforma de testes para sistemas de piloto automático utilizando MATLAB/Simulink e simulador de voo X-Plane." Institute of Technology, 2011, Instituto Tecnológico de Aeronáutica – ITA, São Jose dos Campos, SP - Brazil.
- [13] A. Bittar, "Piloto automático para VANTs tutorial comunicação X-PLANE SIMULINK." 2011. [Online]. Available: <https://Pilotoautomaticovant.Wikispaces.Com/Tutorial+Xplane+%26+S+imulink>
- [14] SILVA, N. M. "Modelagem e Simulação de Veículos Aéreos Não Tripulados no Simulador de Voo X-Plane." Instituto Tecnológico de Aeronáutica - ITA, São Jose dos Campos, SP, Brazil.
- [15] Foster, T. M. (2005). "Dynamic stability and handling qualities of small unmanned-aerial-vehicles." Brigham: A thesis submitted to the faculty of Brigham Young University.
- [16] HULL, D. G. "Fundamentals of Airplane Flight Mechanics." [S.l.]: Springer, 2007.
- [17] COMPANY, M. D. A. "The USAF Stability and Control Digital DATCOM User's Manual." 1979.



Fernanda Cavalcante da Silva was born in the state capital of São Paulo, Brazil, in 1994. She holds a B.Sc. degree in Aerospace Engineering from the Universidade do Vale do Paraíba – UNIVAP, São José dos Campos, São Paulo, Brazil. In 2013, she was an undergraduate researcher at the Instituto Nacional de Pesquisas Espaciais (INPE) sponsored by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP). From 2014 to 2015, she was an exchange student at Western Michigan University under the Science Without Borders scholarship from CAPES. Her B.Sc. dissertation was presented in the XII Symposium on Signal Processing in Universidade Federal do ABC (UFABC), São Bernardo do Campo, São Paulo, Brazil, in 2017. Her paper was awarded one of the best in the conference and invited to contribute an extended version to the Journal of Communication and Information System (JCIS). From 2016 to 2017, she was an intern at EMBRAER, São José dos Campos, São Paulo, Brazil, in the engineering systems area. Currently, she is pursuing her M.Sc. degree in Electromechanical Engineering in Universidade da Beira Interior, Covilhã, Portugal.



Marco Antonio Vieira Macedo Grinet was born in Cruzeiro, São Paulo, Brazil in 1989. He received his B.Sc. degree in Biomedical Engineering at the Universidade do Vale do Paraíba – UNIVAP, São José dos Campos, São Paulo, Brazil in 2016. He is currently pursuing his M.Sc. in Biomedical Engineering at the Universidade Federal de São Paulo – UNIFESP, São José dos Campos, São Paulo, Brazil. He partook on an exchange program during his undergraduate studies to the Johns Hopkins University between fall 2014 and spring 2015. Was accepted as an undergraduate researcher as a summer position at the Harvard Biodesign Lab, at Harvard University during the summer of 2015. His research interests are in Medical Imaging, System Modeling, Data Processing, Machine Learning, and Embedded Systems.



Helosman Valente de Figueiredo was born in Cruzeiro do Sul, Acre, Brazil, in 1988. He received the B.Sc. degree in Mechatronic Engineering from the Universidade do Estado do Amazonas, Manaus, Amazonas, Brazil. In 2009 M.Sc. degree in Aerospace Systems and Mechatronics from the Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, Brasil. He is currently Ph.D. candidate at the Instituto Tecnológico de Aeronáutica, São Paulo, Brasil. His research interests are aerospace systems, mechatronics, sensors, simulation, control, vibration, modal analysis, communication. Since 2018, he has been with the Avibras, São José dos Campos, São Paulo, Brasil, where he is currently a Vibration Test Engineer. From 2015 to 2018, he was Professor of electrical engineering at the Universidade do Vale do Paraíba – UNIVAP, São Paulo, Brazil. In the past, he served as the Guest Editor for 6th International Conference on Control, Decision and Information Technologies (CODIT'18), Conference of Computational Interdisciplinary Science (CCIS-2016).