In-Flight Parameter Estimation for Multirotor Aerial Vehicles

Davi Ferreira de Castro[1,a], Igor Afonso Acampora Prado[1,b], and Mateus de Freitas Virgílio Pereira[1,c], Davi Antônio dos Santos[1,d] and José Manoel Balthazar[1,e]

Aeronautics Institute of Technology, ITA, Brazil.

Abstract. This paper proposes a method for in-flight parameter estimation for Multirotor Aerial Vehicles (MAV). This task is important because it provides parameters with better accuracy for the actual vehicle operation. In order to simulate a flight it is adopted a simulation environment Software-In-the-Loop (SIL).

1 Introduction

When dealing with MAVs, flight tests, regardless of being performed in outdoor or indoor environments, involve risks, both for equipment and for the people who are participating in the experiment. These risks can be explained by the MAVs high speed propellers [4] and, from the standpoint of control, they are unstable, due to nonlinearities, aerodynamic disturbances [5] and a complex and coupled dynamic [1].

To avoid hazards during flight tests of such vehicles, simulation is an alternative to be used. Nevertheless, the simulation does not eliminate the need for flight tests. Therefore, it is only a preliminary step.

In general, simulations save time during the project development, and they can be said to be more efficient than experimental tests time-wise. Even so, simulations may not truly represent reality. An usual approach is SIL simulations, in which all the components are simulated, such as sensors, actuators and vehicle model.

SIL simulations can validate a system before an experimental test, and it also allows the design of a compatible hardware with the software while avoiding damages of equipment. In this context, a simulation environment is useful for estimating in-flight parameters, since there are difficulties and risks to perform a real data acquisition.

This paper is organized as follows: in section 2 the MAV modeling is presented, in section 3 the simulation environment SIL is described, in section 4 the method for estimation is presented, in section 5 the simulation results are shown and finally in section 6 some conclusions and future work suggestions are proposed.

2 MAV Modeling

Consider an octocopter and three Cartesian coordinate systems (CCS) illustrated in Figure 1.

[a] e-mail: davifc@ita.br
[b] e-mail: igorap@ita.br
[c] e-mail: mateusfvp@hotmail.com
[d] e-mail: davists@ita.br
[e] e-mail: jmbaltha@ita.br

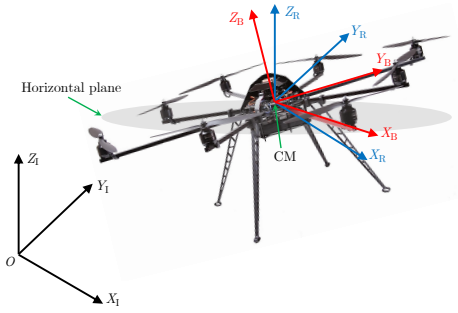


Fig. 1. Cartesian coordinate systems.

The CCS of the body $S_B \triangleq \{X_B,\ Y_B,\ Z_B\}$ is fixed to the vehicle structure and centered on the center of mass (CM). The inertial CCS $S_I \triangleq \{X_I,\ Y_I,\ Z_I\}$ is fixed to the earth at the point O. The CCS $S_R \triangleq \{X_R,\ Y_R,\ Z_R\}$ is parallel to S_I and centered in CM.

2.1 Mathematical Model

The differential equations that describe the dynamic behavior of the multirotor can be obtained by the Newton-Euler formalism. This work chooses to represent the attitude by the quaternion of rotation $\mathbf{q} \in \mathbb{R}^4$, however for visualization is adopted Euler angles (ϕ, θ and ψ).

The kinematic attitude model for the quaternion of rotation is given by the following differential equation [7]:

$$\dot{\mathbf{q}} = \boldsymbol{\Omega}\mathbf{q} \tag{1}$$

where

$$\boldsymbol{\Omega} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}\times] \end{bmatrix}, \tag{2}$$

being $\boldsymbol{\omega} \triangleq \begin{bmatrix}\omega_x\ \omega_y\ \omega_z\end{bmatrix}^T \in \mathbb{R}^3$ the angular velocity of the vehicle represented in S_R and $[\boldsymbol{\omega}\times]$ is given by:

$$[\boldsymbol{\omega}\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{3}$$

Rewriting the equation (1) explicitly, one obtains:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \tag{4}$$

Applying Newton's second law for rotational motion and neglecting disturbance torques, one can obtain:

$$\boldsymbol{\tau} = \dot{\mathbf{h}} + \boldsymbol{\omega} \times \mathbf{h}, \tag{5}$$

where $\boldsymbol{\tau} \triangleq [\tau_x \ \tau_y \ \tau_z]^{\mathrm{T}} \in \mathbb{R}^3$ is the resulting propulsion torque in the vehicle represented in S_{B} and $\mathbf{h} \triangleq \mathbf{I}\,\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular momentum of the body represented in S_{B}, being $\mathbf{I} \in \mathbb{R}^{3x3}$ the matrix of inertia of the vehicle represented in S_{B}. It is assumed that the vehicle has symmetrical structure with respect to the coordinate axes. The matrix $\mathbf{I}$ resulting in a diagonal matrix defined by:

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}. \tag{6}$$

Rewriting the equation (5) explicitly, one can obtain:

$$\dot{\omega}_x = \frac{(I_y - I_z)}{I_x}\omega_y\omega_z + \frac{\tau_x}{I_x}. \tag{7}$$

$$\dot{\omega}_y = \frac{(I_z - I_x)}{I_y}\omega_x\omega_z + \frac{\tau_y}{I_y}. \tag{8}$$

$$\dot{\omega}_z = \frac{(I_x - I_y)}{I_z}\omega_x\omega_y + \frac{\tau_z}{I_z}. \tag{9}$$

Applying the second Newton's law of motion and neglecting any perturbation, we obtain the following model of translational movement in S_{I}:

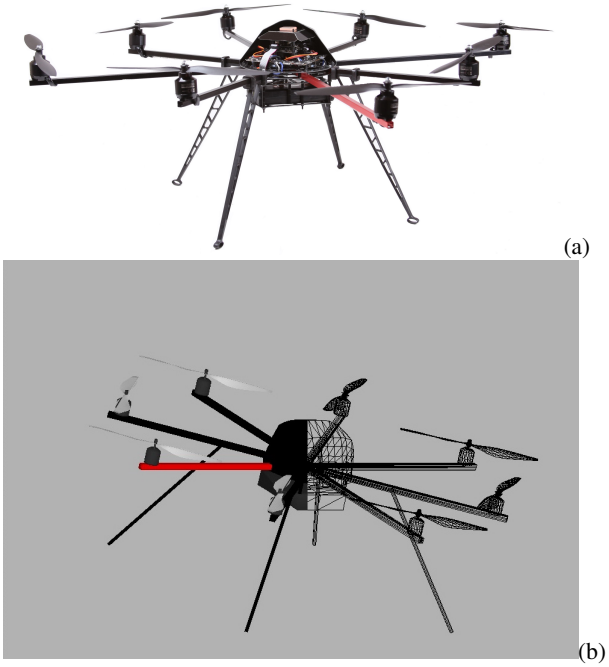
$$\ddot{\mathbf{r}} = \frac{1}{m} f\mathbf{n} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \tag{10}$$

where $\mathbf{r} \triangleq [r_x \ r_y \ r_z]^{\mathrm{T}} \in \mathbb{R}^3$ is the three-dimensional position of the CM represented in S_{I}, $\mathbf{n} \triangleq [n_x \ n_y \ n_z]^{\mathrm{T}} \in \mathbb{R}^3$ is the normal unit vector perpendicular to the rotor plane represented in S_{I}, f is the total thrust, g is the acceleration of gravity and m is the mass of the vehicle.

2.2 X-Plane Model

In this work it was considered the Gyro-200ED-X8 octocopter of the company Gyrofly Innovations [1], as illustrated in Figure 2(a). There was obtained a graphic model of Gyro-200ED-X8 octocopter for simulation in X-Plane. The tool used in the implementation of the model was the Plane-Maker. The Plane Maker is a software provided with X-Plane that allows the implementation of new aircraft models. The Figure 2(b) illustrates a graphical model of the Gyro-200ED-X8 octocopter implemented in the software Plane-Maker.

[1] http://www.gyrofly.com.br



(a) (b)

Fig. 2. (a)Illustration of Gyro-200ED-X8 UAV of Gyrofly Innovations and (b) Octocopter model Gyro-200ED-X8 implemented in Plane-Maker.

3 Simulation Environment SIL

The environment consists in a computer running two software: MATLAB/Simulink and X-Plane. The attitude control laws and position are implemented in Simulink, while the vehicle dynamics and the flight environment are simulated in X-Plane. The sequence of sending and receiving data by the software MATLAB/Simulink and XPlane, can be observed in Figure 3.

The standard method of X-Plane to communicate with external processes and machines is through User Datagram Protocol (UDP). The steps in this data flow are: the X-Plane sends the outputs of the octocopter model to the control system using UDP (in this work it was used a rate of 90 Hz); the MATLAB/Simulink receives the data and performs the calculation of the control input signals, which are sent to the X-Plane; the X-Plane receives control input signals via UDP with values for throttle command. The works of [6] and [2] show more details of the communication between MATLAB/Simulink and X-Plane.

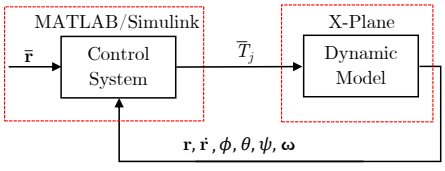


Fig. 3. Data flow between X-Plane and MATLAB/Simulink.

4 Parameter Estimation X-Plane Model

Define the set of identification indexes for the rotors of the octocopter as $\mathcal{J} \triangleq \{1, 2, ..., 8\}$. The analytical model of equations (7) - (10) has as inputs the total thrust magnitude f and the net propulsion torque $\boldsymbol{\tau}$. However, inputs of X-Plane model are throttles $T_j, \forall j \in \mathcal{J}$. The T_j has a magnitude which varies from 0 to 1. Since the control systems of both models are identical, it is necessary to convert f and $\boldsymbol{\tau}$ to T_j, in order to make the simulation with the X-Plane model equivalent to the analytical model, therefore allowing comparisons. The initial step of this conversion is to calculate the individual thrust f_j.

Consider the octocopter illustrated in Figure 4. In this figure, $\forall j \in \mathcal{J}$, f_j denotes the real thrust produced by the j-th rotor, τ_j denotes the torque reaction of the j-th rotor. The arm length of the octocopter is denoted by l.

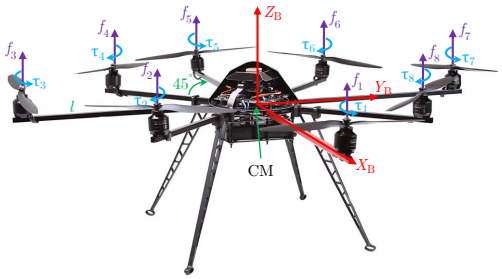


Fig. 4. Thrust and torque reaction of the rotors.

Through inspection of Figure 4, we can obtain the magnitude of the total thrust f and the components of the resultant propulsion torque $\boldsymbol{\tau} \triangleq [\tau_x \ \tau_y \ \tau_z]^{\mathrm{T}} \in \mathbb{R}^3$, as being

$$\begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{\Gamma}_O \mathbf{f}, \tag{11}$$

Based on equation (11), one can also affirm that the individual thrust commands $\overline{f}_j, \forall j \in \mathcal{J}$, are related to $\overline{f}$ and $\overline{\boldsymbol{\tau}}$, by:

$$\begin{bmatrix} \overline{f} \\ \overline{\boldsymbol{\tau}} \end{bmatrix} = \boldsymbol{\Gamma}_O \bar{\mathbf{f}}, \tag{12}$$

where

$$\bar{\mathbf{f}} \triangleq [\overline{f}_1 \ \overline{f}_2 \ \overline{f}_3 \ \overline{f}_4 \ \overline{f}_5 \ \overline{f}_6 \ \overline{f}_7 \ \overline{f}_8]^{\mathrm{T}} \tag{13}$$

and

$$\boldsymbol{\Gamma}_O \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -\frac{\sqrt{2}}{2}l & -l & -\frac{\sqrt{2}}{2}l & 0 & \frac{\sqrt{2}}{2}l & l & \frac{\sqrt{2}}{2} \\ -l & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & l & \frac{\sqrt{2}}{2}l & 0 & -\frac{\sqrt{2}}{2}l \\ k & -k & k & -k & k & -k & k & -k \end{bmatrix}, \tag{14}$$

where k is the coefficient of yaw torque, defined by $k \triangleq k_\tau / k_f$.

Using the Moore-Penrose pseudo-inverse matrix, the equation (12) is inverted, obtaining

$$\bar{\mathbf{f}} = \boldsymbol{\Gamma}_O^{\dagger} \begin{bmatrix} \overline{f} \\ \overline{\boldsymbol{\tau}} \end{bmatrix}. \tag{15}$$

The last step is to convert $\overline{f}_j$ to $\overline{T}_j$. It was made some tests in order to estimate how $\overline{T}_j$ relates to both $\overline{f}_j$ and $\overline{\tau}_j$.

The tests consisted in varying $\overline{T}_j$ and measuring both $\overline{f}_j$ and $\overline{\tau}_j$. It is expected that the measured values are equal to the command values. These tests have shown that $\overline{T}_j$ relates to both $\overline{f}_j$ and $\overline{\tau}_j$ by a linear equations below:

$$\overline{f}_j = k_f \overline{T}_j, \tag{16}$$

$$\overline{\tau}_j = k_\tau \overline{T}_j. \tag{17}$$

5 Simulation Tests

The dynamics of the mathematical model is integrated using the fourth order Runge-Kutta method with time step of $T = 0.002$ s, which is the same time step for the X-Plane model. The mass of the vehicle is $m = 2.132$ kg and the acceleration of gravity is $g = 9.796$ m/s^2.

In order to estimate the parameters k_f and k_τ, it was elaborated a test that makes use of a simulation environment that was presented Section 3.

The test is the following procedures: stabilize the vehicle attitude and height; add a random signal δ_1 to $\overline{T}_1$ to excite the models; measure f_1 and τ_1; estimate k_f and k_τ using the criterion of least squares and the measurements of f_1 and τ_1; and validate the estimated models. Only the data for rotor 1 are used, since all rotors are considered to be identical.

Let $\tilde{f}_1$ and $\tilde{\tau}_1$ denote measures of f_1 and τ_1, respectively. Modeling the measures by:

$$y(t) = x(t)\boldsymbol{\Theta} + \epsilon(t), \tag{18}$$

where $y(t) \triangleq [\tilde{f}_1(t) \ \tilde{\tau}_1(t)]^{\mathrm{T}} \in \mathbb{R}^2$, $x(t) \triangleq [\overline{T}_1(t)] \in \mathbb{R}$, $\boldsymbol{\Theta} \triangleq [k_f \ k_\tau]^{\mathrm{T}} \in \mathbb{R}^2$ and $\epsilon(t) \triangleq [\epsilon_1(t) \ \epsilon_2(t)]^{\mathrm{T}} \in \mathbb{R}^2$ represents eventual modeling errors or noise in the measurement.

Applying the data sets in the equation (18), for each sample at an instant of time $t = t_1, t_2, ..., t_n$ and writing in matrix form, one obtains:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\Theta} + \mathbf{E}, \tag{19}$$

where $\mathbf{Y} \triangleq [y(t_1) \ y(t_2) \ ... \ y(t_n)]^{\mathrm{T}}$, $\mathbf{X} \triangleq \begin{bmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ x(t_n) \end{bmatrix}$ e $\mathbf{E} \triangleq [\epsilon(t_1) \ \epsilon(t_2) \ ... \ \epsilon(t_n)]^{\mathrm{T}}$.

Denoting the estimates of $\boldsymbol{\Theta}$ by $\widehat{\boldsymbol{\Theta}}$, one obtains $\widehat{\boldsymbol{\Theta}}$ according to the criterion of least squares (LS) [3], given by:

$$\widehat{\boldsymbol{\Theta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}. \tag{20}$$

For the models validation, a test was performed according to the following procedures: insertion of the estimated parameters in the models; stabilization of the vehicle attitude and height; random signal input μ_1 in $\overline{\tau}$; and comparison between the data for $\overline{f}_1$ and $\tilde{f}_1$.

In Figure 5 the random signals δ_1 and μ_1 are presented. In Figure 6 the comparison between $\overline{f}_1$ and $\tilde{f}_1$ is shown, in which the data for $\tilde{f}_1$ is expected to be equivalet to $\overline{f}_1$. In Figure 6(a) the test results before the models estimation are

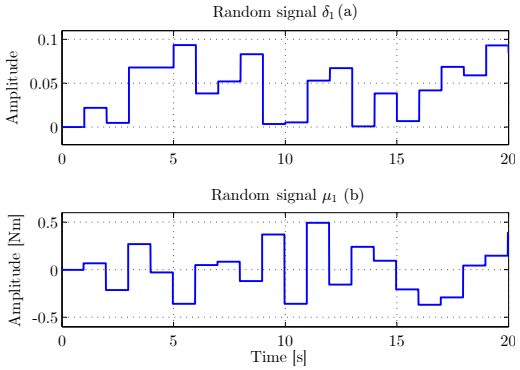


Fig. 5. Random signals δ_1 and μ_1.

presented and one can note that $\overline{f}_1$ and $\tilde{f}_1$ are not similar. For these tests the values for k_f and k_τ are arbitrary. In Figure 6(b) the results after the models estimation are shown and it can be seen that $\overline{f}_1$ and $\tilde{f}_1$ are equivalent, since the values used for k_f and k_τ were the estimated.

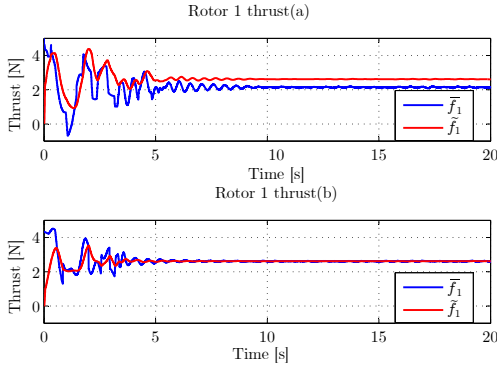


Fig. 6. (a) Tests before the parameters estimation with arbitrary values for k_f and k_τ. (b) Tests after the parameters estimation with the estimated values for k_f and k_τ.

For the estimation of (I_x, I_y e I_z), three tests are carried out. Each test consists in moving the vehicle along a single axis where the estimates have to be made, while the movement in the other axes are disregarded.

The Table 1 presents the estimated parameters.

Table 1. Parameter estimation k_f,k_τ,I_x, I_y and I_z

k_f [N]	k_τ [N m]	I_x [Kg m^2]	I_y [Kg m^2]	I_z [Kg m^2]
28, 0269	0, 3869	0, 0628	0, 0762	0, 1014

6 Conclusion

The methodology presented proved to be effective, because it is possible to estimate the MAVs in-flight parameters through a simulation environment SIL. In this way we can minimize the risks in the operation of such vehicles. This methodology can be applied before flight tests. As future work, the presented procedures can be implemented in flight tests.

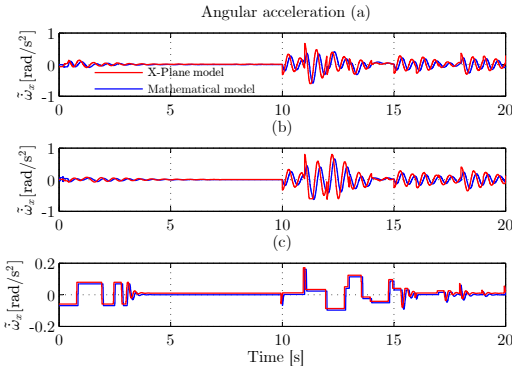


Fig. 7. (a) Data for $\tilde{\omega}_x$ for the X-Plane and the mathematical models obtained through tests carried out after the estimation of I_x in order to validate the estimated parameters. It should be noted that, after the stabilization of the models, μ_1 is added at 10 sec. (b) and (c) correspond, respectively, to the data for $\tilde{\omega}_y$ and $\tilde{\omega}_z$ for both models.

Acknowledgements

We would like to thanks Fundação de Amparo à Pesquisa do Amazonas (FAPEAM), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the grants awarded.

References

1. Z. Fang, X. Y. Wang, and J. Sun. Design and nonlinear control of an indoor quadrotor flying robot. In *Proceedings...*, pages 429–434, Piscataway, 2010. World Congress on Intelligent Control and Automation, IEEE.
2. H. V. Figueiredo and O. Saotome. Simulation platform for quadricopter: Using Matlab/Simulink and X-plane. pages 51–55. Brazilian Robotics and Latin American Robotics Symposium, 2012.
3. L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Clifs, New Jersey, 2 edition, 1987.
4. F. Mutter, S. Gareis, B. Schatz, A. Bayha, F. Gruneis, M. Kanis, and D. Koss. Model-driven in-the-loop validation: Simulation-based testing of UAV software using virtual environments. In *Proceedings...*, pages 269–275, Piscataway, Apr. 2011. International Conference and Workshops on Engineering of Computer Based Systems, IEEE.
5. G. V. Raffo, M. G. Ortega, and F. R. Rubio. An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter. *Automatica*, 46:29–39, 2010.
6. L. R. Ribeiro and N. M. F. Oliveira. UAV autopilot controllers test platform using Matlab/Simulink and X-Plane. pages 1–6. ASEE/IEEE Frontiers in Education Conference, 2012.
7. J. R. Wertz. *Spacecraft Attitude Determination and control*. Kluwer Academic Publishers, The Netherlands, 1978.