23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

STOCHASTIC MODELING AND CLASSICAL CONTROL OF A QUADROTOR

Fernando Augusto de Noronha Castro Pinto
PEM/COPPE/UFRJ - P.O. Box 68503 – 21945-970 – Rio de Janeiro, RJ, Brazil.
Federal University of Rio de Janeiro
fcpinto@ufrj.br

David Julián González Maldonado
PEM/COPPE/UFRJ - P.O. Box 68503 – 21945-970 – Rio de Janeiro, RJ, Brazil.
Federal University of Rio de Janeiro
david.julian@ufrj.br

Thiago Gamboa Ritto
PEM/COPPE/UFRJ - P.O. Box 68503 – 21945-970 – Rio de Janeiro, RJ, Brazil.
Federal University of Rio de Janeiro
tritto@mecanica.ufrj.br

Abstract.
The objective of this paper is to present the design and construction of a Quadrotor for indoor purposes. The aircraft will be capable of stabilize inside a closed environment by means of a digital controller programed in its processing system. Firstly, a deterministic model is introduced using Newton-Euler equations and some considerations are taken in order to simplify the model. Next, a stochastic model will be considered to handle the uncertainties related to the sensors by adding gaussian noise. Later, a Kalman Filter is used to estimate the states and output of the system. Finally, four independent digital PID controllers are designed to control the height, roll, pitch and yaw of the Quadrotor.

Keywords: *Quadrotor, Dynamic Modeling, Newton-Euler Equations, Classical Control, Kalman Filter*

1. INTRODUCTION

Aeronautics is a highly developed area due to scientific and technological contributions from persons motivated by the desire of flying. Initially, the fixed-wing aircraft was one of the first architectures successfully developed, bringing along a wide range of application used nowadays. However, there are some areas where it is necessary another type of aircraft architecture that allows slow-speed flight to land in difficult access places. This demand is satisfied currently by rotary-wing aircraft, such as helicopters, which have a complex rotating mechanism along with a set of propellers that provide the necessary thrust to take off. On that basis, similar aircraft architecture have emerged in which multiple rotors are used. One of them is called Quadrotor.

A Quadrotor is a rotary-wing aircraft comprised of 4 propellers symmetrically spaced and driven by 4 independent motors. In relation to the helicopter, the Quadrotor uses direct coupling between the propeller and motor; it has constant-pitch propeller; it does not need to tilt its rotor to move; it has a low mechanical complexity; it has few gyroscopic effects; among others advantages. This provides interesting features such as: high maneuverability, low weight, reduced manufacturing cost, small size, high reliability, low maintenance frequency, access to constrained areas. These advantages attracted the interest from academic, industrial and military sectors, which are increasingly deepening the knowledge in order to provide solutions to research lines and technological development.

The Quadrotor has 6 degrees of freedom (DOF): 3 DOF for position and 3 DOF for orientation. To represent the position, a vector linking the Quadrotor center of mass and the origin of a inertial frame is used. To represent the orientation, Euler angles or Quaternions are used. The mathematical model is based on 2 approaches: lagrangian and newtonian. In the former, the system energy is analyzed. In the latter, the change of the linear and angular momentum is calculated. From the newtonian mechanics, used in this paper, the Newton-Euler method provides a set of differential equation system that describes the coupled dynamic between the position and orientation.

Several controllers have been proposed and implemented successfully in Quadrotors. In Bouabdallah (2007), Pounds *et al.* (2004), Hoffmann *et al.* (2012), Bresciani (2008),Pounds *et al.* (2010), Schmidt (2011) e Bouabdallah *et al.* (2004) PID and LQR controllers where used to reach hovering flights with slow trajectory following. Bouffard (2012) used a Learning-based Model Predictive Controller to improve system performance by updating the controller parameters while flying. Bertrand *et al.* (2011) implemented a Hierarchical Controller in which a temporal separation between the translational and rotational dynamic is applied. Raffo *et al.* (2010) used a $\mathscr{H}_\infty$ controller to cope with unknown perturbations and trajectories. Waslander *et al.* (2005) formulated a Integral Sliding Modes controller and a Reinforcement Learning controller capable of creating a dynamic model using the input and output measurements. Bouabdallah (2007) proposed a Backstepping controller by dividing the system equations into 4 subsystems and analyzing their stability.

2. Dynamic Model

This sections presents the dynamic model of a Quadrotor in order to describe the aircraft behavior in air. A Newton-Euler method is used to obtain a set of 6 equations relating the position and orientation.

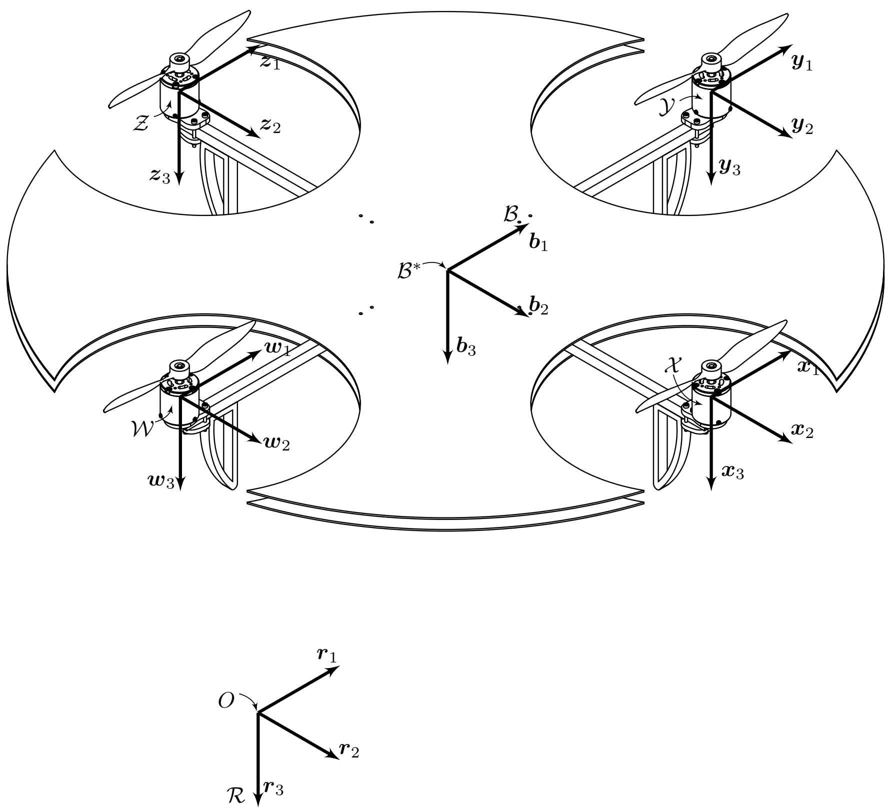

Figure 1. Reference frames $\mathcal{R}$, $\mathcal{B}$, $\mathcal{W}$, $\mathcal{X}$, $\mathcal{Y}$ e $\mathcal{Z}$

2.1 Reference Frames

Before formulating the differential equations that governs the Quadrotor movement, 6 reference frames are defined to express the vectors and tensors. Each body is assigned a referential with their unit vectors in the direction of their principal axes of inertia. The origin of each referential is located in the center of mass of each body to simplify the model (see Figure 1).

2.2 Representation of the Position

The position is represented by the vector $\boldsymbol{p}^{\mathcal{B}^*/O}$, that starts at point O and ends at the center of mass $\mathcal{B}^*$. This vector is defined as follows.

$$\boldsymbol{p}^{\mathcal{B}^*/O} = x\boldsymbol{r}_1 + y\boldsymbol{r}_2 - z\boldsymbol{r}_3 \tag{1}$$

2.3 Representation of Orientation

The orientation is represented using 3-2-1 Euler angles, which relate $\mathcal{R}$ and $\mathcal{B}$ by consecutive rotations around 3 auxiliary reference frames. These frames are labeled as $\mathcal{E}'$, $\mathcal{E}''$, $\mathcal{E}'''$. Thus, the mapping that relates the 2 reference frames is shown in Equation 2.

$$
{}^{\mathcal{R}}\mathbf{R}^{\mathcal{B}} = {}^{\mathcal{R}}\mathbf{R}^{\mathcal{E}'}\,{}^{\mathcal{E}'}\mathbf{R}^{\mathcal{E}''}\,{}^{\mathcal{E}''}\mathbf{R}^{\mathcal{B}} = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{2}
$$

where, ${}^{\mathcal{R}}\mathbf{R}^{\mathcal{E}'}$ is the rotation of the reference frame $\mathcal{E}'$ around vector $\boldsymbol{e}_3'$ with an angle ψ, ${}^{\mathcal{E}'}\mathbf{R}^{\mathcal{E}''}$ is the rotation of the reference frame $\mathcal{E}''$ around vector $\boldsymbol{e}_2''$ with an angle θ and ${}^{\mathcal{E}''}\mathbf{R}^{\mathcal{E}'''} = {}^{\mathcal{E}''}\mathbf{R}^{\mathcal{B}}$ is the rotation of the reference frame $\mathcal{E}'''$ around vector $\boldsymbol{e}_1'''$ with an angle ϕ.

2.4 Differential Kinematic Equations

So far, the problem of describing the instantaneous position and orientation by a rotation matrix has been addressed. Now, a time dependency between 2 reference frames is considered. These relation is describe by the Kinematic Differential Equations. For this purpose, the generalized vector of Equation 3 is used to represent the position and orientation. It is comprised of the linear velocity, expressed in inertial reference frame, and the angular velocity, expressed in the 3 auxiliary reference frames.

$$
\dot{\boldsymbol{\xi}} = \left[[{}^{\mathcal{R}}\boldsymbol{v}^{\mathcal{B}^*}]_{\mathcal{R}} \quad [{}^{\mathcal{R}}\boldsymbol{\omega}^{\mathcal{B}}]_{\mathcal{E}} \right]^T = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \tag{3}
$$

For simplicity in the dynamic model, it is useful to express the linear and angular velocity in the reference frame $\mathcal{B}$. For this, the following generalized vector is proposed.

$$
\boldsymbol{\nu} = \left[[{}^{\mathcal{R}}\boldsymbol{v}^{\mathcal{B}^*}]_{\mathcal{B}} \quad [{}^{\mathcal{R}}\boldsymbol{\omega}^{\mathcal{B}}]_{\mathcal{B}} \right]^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{4}
$$

Hence, the vectors and tensors involved in the dynamic model can be expressed either in $\boldsymbol{\nu}$ or $\dot{\boldsymbol{\xi}}$, as appropriate. The relation of these generalized vectors shown in Equation

$$
\dot{\boldsymbol{\xi}} = {}^{\mathcal{C}}\mathbf{T}^{\mathcal{B}}\boldsymbol{\nu} \tag{5}
$$

where,

$$
{}^{\mathcal{C}}\mathbf{T}^{\mathcal{B}} = \begin{bmatrix} {}^{\mathcal{R}}\mathbf{R}^{\mathcal{B}} & \mathbf{0} \\ \mathbf{0} & {}^{\mathcal{E}}\mathbf{T}^{\mathcal{B}} \end{bmatrix}, \quad {}^{\mathcal{E}}\mathbf{T}^{\mathcal{B}} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix} \tag{6}
$$

The Equation 6 allows to transform the linear and angular velocity expressed in reference frame $\mathcal{B}$ to the linear and angular velocity expressed in the reference frames $\mathcal{R}$, $\mathcal{E}'$, $\mathcal{E}''$ e $\mathcal{E}'''$. Note that ${}^{\mathcal{E}}\mathbf{T}^{\mathcal{B}}$ has a singularity when $\theta = \pm\pi/2$. However, this situation will not be reached if the Quadrotor is in hovering flight. In the case in which the Quadrotor performs aggressive maneuvers where the singularity is reached, it is advisable to use a different representation of the orientation, such as Quaternions.

2.5 Newton-Euler Formulation

The Quadrotor dynamic equations are developed by means of a newtonian approach. The Newton-Euler formulation provides a set of 6 equations involving the position and orientation. For the position, a conservation of linear momentum is applied. For the orientation, a conservation of angular momentum is applied. Mathematically, these laws are expressed in Equation 7.

$$
\begin{aligned}
\mathbf{F}^i &= \frac{{}^{\mathcal{R}}d}{dt}\left(m_i\,{}^{\mathcal{R}}\boldsymbol{v}^{i^*}\right), \quad i = \{\mathcal{B}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}\} \\
\mathbf{M}^{F^i/i^*} &= \frac{{}^{\mathcal{R}}d}{dt}\left(\mathbf{J}^{i/i^*}\,{}^{\mathcal{R}}\boldsymbol{\omega}^i\right), \quad i = \{\mathcal{B}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}\}
\end{aligned} \tag{7}
$$

where, $\mathbf{F}^i$ are the forces applied to body i, m_i is the mass of body i, ${}^{\mathcal{R}}\boldsymbol{v}^{i^*}$ is the linear velocity of body i with respect to $\mathcal{R}$, $\mathbf{M}^{F^i/i^*}$ are the moments and torques applied to body i, $\mathbf{J}^{i/i^*}$ is the moment of inertia of body i with respect to its center of mass and ${}^{\mathcal{R}}\boldsymbol{\omega}^i$ is the angular velocity of body i with respect to $\mathcal{R}$. Using Equations 7, 1, 2, 3 and 4, the Quadrotor dynamic model is presented in Equation 8.

$$\begin{aligned}
\ddot{x} &= (-\sin\psi\sin\phi - \cos\psi\sin\theta\cos\phi)\frac{b(\Omega_w^2+\Omega_x^2+\Omega_y^2+\Omega_z^2)}{m}\\
\ddot{y} &= (\cos\psi\sin\phi - \sin\psi\sin\theta\cos\phi)\frac{b(\Omega_w^2+\Omega_x^2+\Omega_y^2+\Omega_z^2)}{m}\\
\ddot{z} &= g - \cos\theta\cos\phi\frac{b(\Omega_w^2+\Omega_x^2+\Omega_y^2+\Omega_z^2)}{m}\\
\dot{p} &= \frac{J_{YY}-J_{ZZ}}{J_{XX}}q\,r + \frac{J_{33}q\,(-\Omega_w+\Omega_x-\Omega_y+\Omega_z)}{J_{XX}} + \frac{l\,b(-\Omega_x^2+\Omega_z^2)}{J_{XX}}\\
\dot{q} &= \frac{J_{ZZ}-J_{XX}}{J_{YY}}p\,r + \frac{J_{33}p\,(-\Omega_w+\Omega_x-\Omega_y+\Omega_z)}{J_{YY}} + \frac{l\,b(-\Omega_w^2+\Omega_y^2)}{J_{YY}}\\
\dot{r} &= \frac{J_{XX}-J_{YY}}{J_{ZZ}}p\,q + \frac{d(-\Omega_w^2+\Omega_x^2-\Omega_y^2+\Omega_z^2)}{J_{ZZ}}
\end{aligned} \tag{8}$$

where m is the total mass of Quadrotor, g is the gravitational force, b is the thrust coefficient of propellers, d is the drag coefficient of propellers, l is the center of Quadrotor to center of propeller distance, J_{XX}, J_{YY} and J_{ZZ} are the moments of inertia of Quadrotor, J_{33} the moment of inertia of propellers and Ω_w, Ω_x, Ω_y and Ω_z are the propellers speeds.

3. Linear Model

A linear model of Equation 8 is obtained by applying a linearization around a point of operation. First, the following state vector is used:

$$\boldsymbol{x} = \begin{bmatrix} z & \dot{z} & \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix}^T \tag{9}$$

As can bee seen, 2 states are lacking in Equation 9. This is done because the objective is to reach a hovering flight, without horizontal movement. Also, the following input vector is defined:

$$\boldsymbol{u} = \begin{bmatrix} U_z & U_\phi & U_\theta & U_\psi \end{bmatrix}^T \tag{10}$$

Therewith, the following equations represent the reduced dynamic model.

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{z} \\ g - c_\theta\, c_\phi \dfrac{U_z}{m} \\ \dot{\phi} \\ \dfrac{J_{YY}-J_{ZZ}}{J_{XX}}\dot{\theta}\,\dot{\psi} + \dfrac{J_{33}}{J_{XX}}\dot{\theta}\,\Omega + \dfrac{U_\phi}{J_{XX}} \\ \dot{\theta} \\ \dfrac{J_{ZZ}-J_{XX}}{J_{YY}}\dot{\phi}\,\dot{\psi} + \dfrac{J_{33}}{J_{YY}}\dot{\phi}\,\Omega + \dfrac{U_\theta}{J_{YY}} \\ \dot{\psi} \\ \dfrac{J_{XX}-J_{YY}}{J_{ZZ}}\dot{\phi}\,\dot{\theta} + \dfrac{U_\psi}{J_{ZZ}} \end{bmatrix} \tag{11}$$

$$\boldsymbol{y} = g(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} z & \dot{z} & \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix}^T \tag{12}$$

The linearization of Equation 11 and 12 is performed by the first-order Taylor polynomials of Equation 13 and 14 around the point $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})$.

$$\dot{\boldsymbol{x}} \approx f(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}) + \frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}(\boldsymbol{x}-\bar{\boldsymbol{x}}) + \frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}(\boldsymbol{u}-\bar{\boldsymbol{u}}) \tag{13}$$

$$\boldsymbol{y} \approx g(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}) + \frac{\partial g(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}(\boldsymbol{x}-\bar{\boldsymbol{x}}) + \frac{\partial g(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}(\boldsymbol{u}-\bar{\boldsymbol{u}}) \tag{14}$$

Finally, the linearized equations are expressed in space-state form in Equation 15.

$$\begin{aligned}
\dot{\tilde{\boldsymbol{x}}} &= \mathbf{A}\tilde{\boldsymbol{x}} + \mathbf{B}\tilde{\boldsymbol{u}}\\
\tilde{\boldsymbol{y}} &= \mathbf{C}\tilde{\boldsymbol{x}} + \mathbf{D}\tilde{\boldsymbol{u}}
\end{aligned} \tag{15}$$

where,

$$\mathbf{A} = \frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}, \mathbf{B} = \frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}, \mathbf{C} = \frac{\partial g(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}, \mathbf{D} = \frac{\partial g(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}} \tag{16}$$

3.1 Discretization

The processing system where the controller is implemented requires a discrete-time representation of the Quadrotor dynamic equations. The digital controller system needs an ADC (Analog-to-Digital Converter) to convert the continuous-time signals $\boldsymbol{y}(t)$ into a digital discrete-time signal $\boldsymbol{y}(t_k)$ that represents its amplitude at instant t_k. Thereafter, the microcontroller sends the control signals $\boldsymbol{u}(t_k)$ that is converted into an continuous-time signal $\boldsymbol{u}(t)$ by a DAC (Digital-to-Analog Converter). The mathematical model to be developed relates the discrete-time signal $\boldsymbol{u}(t_k)$ and the discrete-time signal $\boldsymbol{y}(t_k)$. This relation provides an equivalent discrete-time model of the continuous-time model. For that purpose, the Equation 17, which is the solution of a linear system, is substituted in Equation 15.

$$\boldsymbol{x}(t) = e^{\mathbf{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-s')}\mathbf{B}\boldsymbol{u}(s')ds' \tag{17}$$

This leads to the discrete-time linear system of Equation 18.

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{\Phi}\boldsymbol{x}(k) + \boldsymbol{\Gamma}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \mathbf{C}\boldsymbol{x}(k) \end{aligned} \tag{18}$$

where,

$$\begin{aligned} \boldsymbol{\Phi} &= e^{Ah} \\ \boldsymbol{\Gamma} &= \int_0^h e^{As} ds\mathbf{B} \end{aligned} \tag{19}$$

4. PID Controller

The PID is a versatile controller used in many linear and non-linear systems. Its discrete-time equation is presented as follows.

$$U(z) = K_p \left(K_b R(z) - Y(z) + \frac{h}{T_i(z-1)} \Big(R(z) - Y(z) \Big) - \frac{T_d N(z-1)}{(T_d + Nh)z - T_d} Y(z) \right) \tag{20}$$

where, K_p is the proportional gain, K_b is the input gain, T_i is the integral time, T_d is the derivative time, N is the derivative filter coefficient and h the sampling frequency. In Figure 2 the digital control system scheme is shown. $\boldsymbol{r}(t)$ represents the set-point signal, $\boldsymbol{e}(t)$ is the error signal, $\boldsymbol{u}(t)$ is the control signal, $\boldsymbol{y}(t)$ is the output signal and $\hat{\boldsymbol{y}}(t)$ is the estimated output signal.

5. Kalman Filter

In the previous section, a PID controller was implemented in Simulink with the deterministic equations of the Quadrotor. However, the sensors that measures the variables are affected by noise, generally gausian-distributed. To model and reduce this noise, the following stochastic model is proposed:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{\Phi}\boldsymbol{x}(k) + \boldsymbol{\Gamma}\boldsymbol{u}(k) + \boldsymbol{w}(k) \\ \boldsymbol{y}(k) &= \mathbf{C}\boldsymbol{x}(k) + \boldsymbol{v}(k) \end{aligned} \tag{21}$$

where $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ are perturbations with a normal distribution. Once the noise has been modeled, an optimal state estimator is needed to correct the data obtained from sensors. For systems affected by gaussian noise, the Kalman filter is a powerful tool that uses a 2-step estimation to minimize the error variance between the estimated states and measured states. The firs step is the prediction state, represented by the following equation:

$$\hat{\boldsymbol{x}}(k+1|k) = \boldsymbol{\Phi}\hat{\boldsymbol{x}}(k|k-1) + \boldsymbol{\Gamma}\boldsymbol{u}(k) + \mathbf{K}(k)\Big(\boldsymbol{y}(k) - \mathbf{C}\hat{\boldsymbol{x}}(k|k-1)\Big) \tag{22}$$

On the other hand,the second step, the update step, is represented as follows:

$$\hat{\boldsymbol{x}}(k|k) = \hat{\boldsymbol{x}}(k|k-1) + \mathbf{K}_f(k)\Big(\boldsymbol{y}(k) - \mathbf{C}\hat{\boldsymbol{x}}(k|k-1)\Big) \tag{23}$$

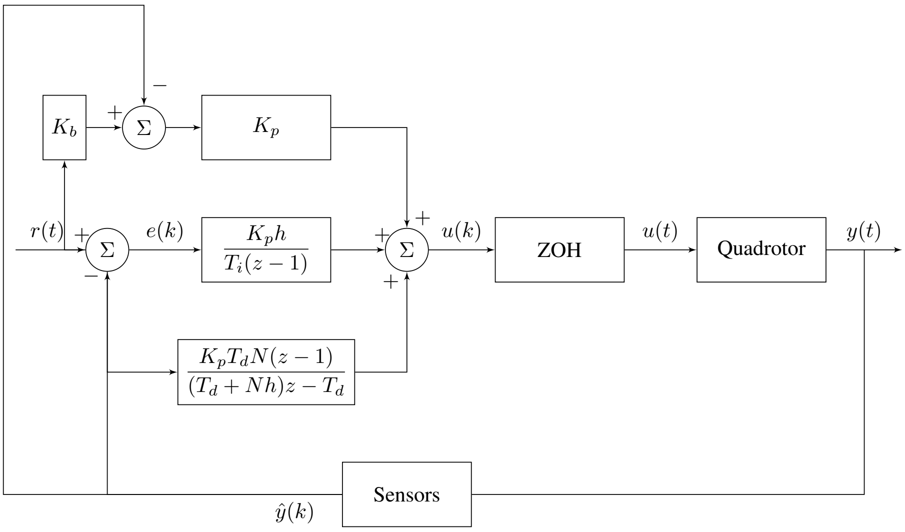

Figure 2. Block diagram of the discrete-time PID controller.

In the prediction step, the previous state estimate is used to get an estimate of the estate at the current time-step. In the update step, the current prediction is combined with the current observation to improve the estate estimate. To test the Kalman filter, a simulation in Simulink is performed with the dynamic equations and PID controller previously addressed. In the Figure 3 is shown the response of the 4 states with the system affected by a gaussian noise with zero mean and a standard deviation of 0.01. The blue signal shows the response with no Kalman filter, while the red signal is the filtered response. As can be seen, the noise is attenuated considerably and, thus, the microcontroller will receive a smoother signal, improving the controller performance.

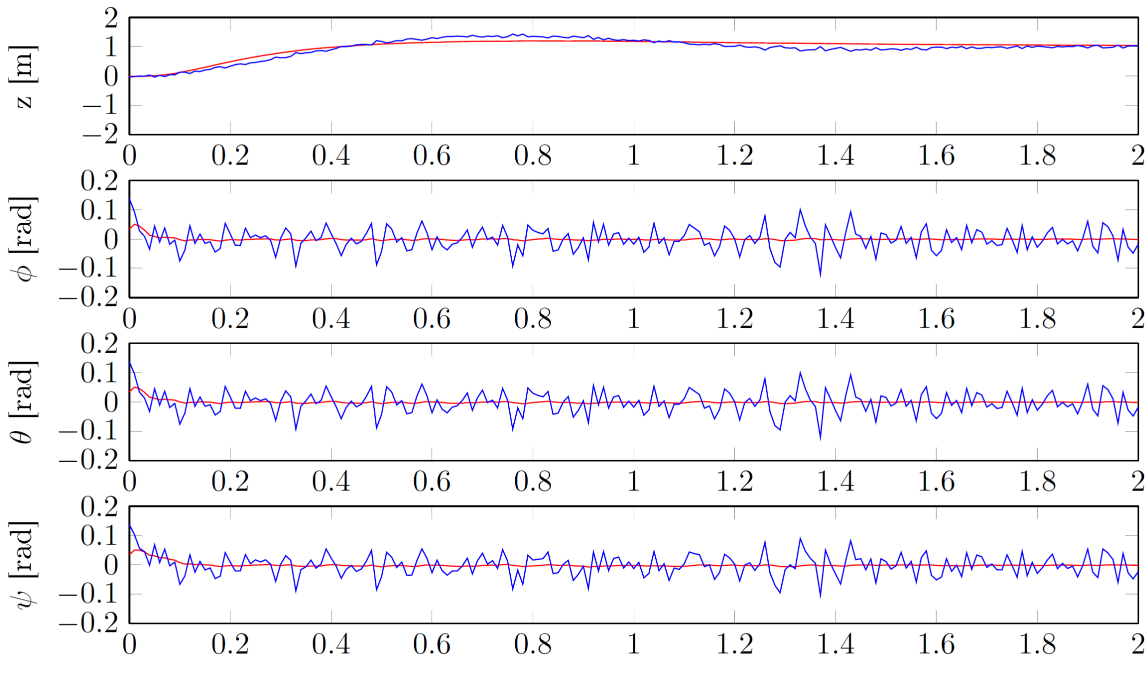

Figure 3. Behavior of the sates variables z, ϕ, θ, ψ using the PID controller. The blue signals are the controlled states using the Kalman filter. The red signals are the controlled states with no Kalman Filter.

6. Implementation and Results

An indoor Quadrotor was built in the Acoustics and Vibration Laboratory (LAVI) at the Federal University of Rio de Janeiro. The Quadrotor, shown in Figure 4, is comprised of a structure, 4 mechanical motors, 4 propellers, 1 inertial sensor, 1 ultrasonic sensor, 1 battery and 1 microcontroller.

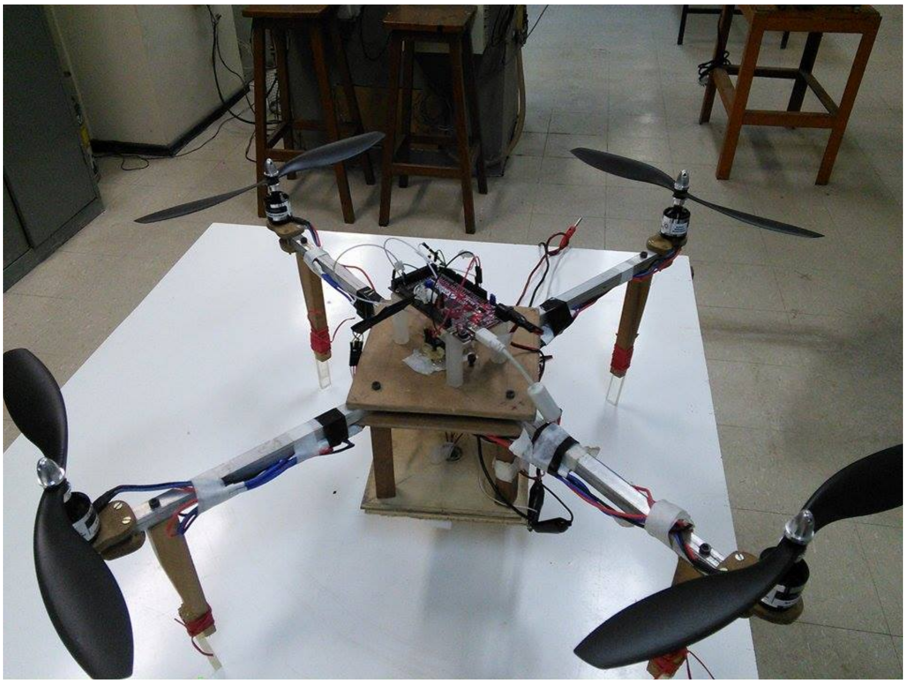

Figure 4. Quadrotor built at the Acoustics and Vibrations Laboratory at Federal University of Rio de Janeiro

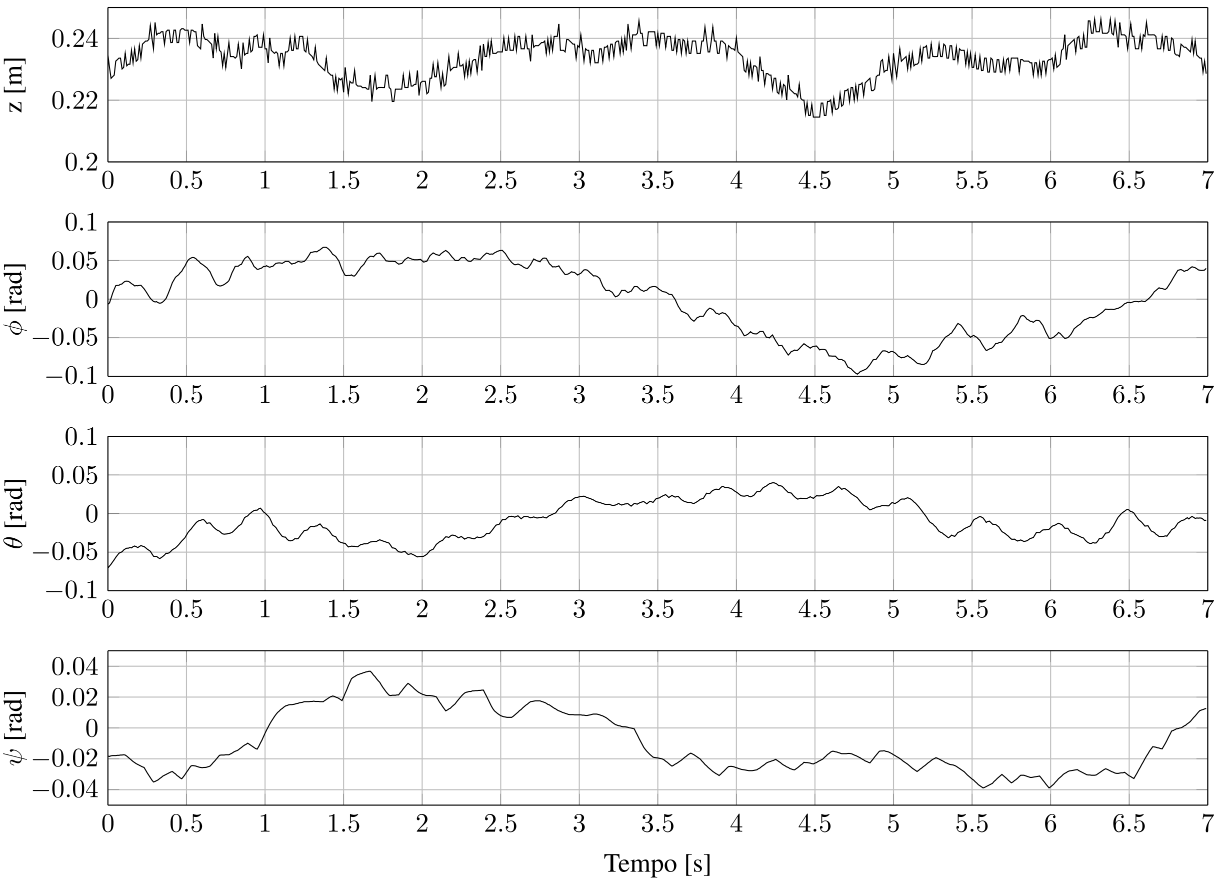

Figure 5. Behavior of the measured states with the real Quadrotor and digital PID.

The control algorithm presented previously is implemented in the microcontroller PIC32 MX795F512L, embedded in the ChipKit Max32 Board. The chip is programed with the software MPLAB X IDE and the C32 compiler. A serial communication between a PC and the Max32 board is establish to adjust the PID parameters and to receive the sensors data while the Quadrotor is flying. The IMU board has a Atmega328 microcontroller to manage the sensor data. Before sending these data to the PIC32, a processing is performed to reduced the measurement error. First, a DCM algorithm is implemented to compute the orientation using accelerometers, gyroscopes and a magnetometer. Then, the measurement is optimized by implementing a Kalman filter. This processing was written in Arduino language and an update rate of 100 Hz was achieved.

Although the PID parameters where tuned many times, it was not possible to maintain the Quadrotor at hovering because it has a horizontal drift due to angles oscillations. This drift caused a fast horizontal movements that made impossible to leave the Quadrotor free. This happened because, in the controller design, the x and y directions where not considered as no sensor where used to measured those variables. Despite this fact, a steel cable, that do not transmit moment, was attached bellow the Quadrotor to constrain the horizontal movement, leaving the other 4 states free. With this modification, the Quadrotor maintained a hovering flight. The 4 states behavior during this flight can be seen in Figure 5.

7. CONCLUSIONS

In this paper the dynamic model of a Quadrotor was determined using Newton-Euler equations. This model provided a set of 6 non-linear and coupled equations: 3 for position and 3 for orientation. This fact makes necessary to vary the orientation to change the position and vice-versa. Nevertheless, the system equations where linearized in order to decouple the variables and simplify the controller design. A PID controller was designed to mantain the Quadrotor in hovering flight. As the sensors measurements are affected by noise, an Extended Kalman Filter was used. Initially, the PID controller and EK filter were tested via simulations in Matlab. Then, the filter and controller equations were discretized and implemented in a microcontroller mounted on the Quadrotor. Since only the height was the only variable controlled in the position equations, the Quadrotor presented horizontal drift, hindering the Quadrotor hovering. To resolve this issue and to asses the 4 DOF controller, a thin steel cable was attached to the Quadrotor base to constrain horizontal movement. Thus, the controller managed to keep the Quadrotor in hovering flight.

8. REFERENCES

Bertrand, S., Guénard, N., Hamel, T., Piet-Lahanier, H. and Eck, L., 2011. "A hierarchical controller for miniature VTOL UAVs: Design and stability analysis using singular perturbation theory". *Control Engineering Practice*, Vol. 19, No. 10, pp. 1099–1108.

Bouabdallah, S., 2007. *Design and Control of Quadrotors with Application to Autonomous Flying*. Tese de D.Sc., Faculté des Sciences et Techniques de L'ingénieur, École Polytechnique Fédérale de Lausanne, Lausanne, Suíça.

Bouabdallah, S., Noth, A. and Siegwan, R., 2004. "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor". In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sendal, Japão.

Bouffard, P., 2012. "On-board Model Predictive Control of a Quadrotor Helicopter: Design, Implementation, and Experiments". Technical Report UCB/EECS-2012-241, University of California at Berkeley, California, Estados Unidos.

Bresciani, T., 2008. *Development of a INS/GPS Navigation Loop for an UAV*. Tese de M.Sc., Department of Automatic Control, Lund University, Lund, Suécia.

Hoffmann, G.M., Huang, H., Waslander, S.L. and Tomlin, C.J., 2012. "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment". In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. Minnesota, Estados Unidos.

Pounds, P., Mahony, R. and Corke, P., 2010. "Modelling and Control of a Large Quadrotor Robot". *Control Engineering Practice*, Vol. 18, No. 7, pp. 691–699.

Pounds, P., Mahony, R. and Gresham, J., 2004. "Towards Dynamically-Favourable Quad-Rotor Aerial Robots". In *Proceedings of Australasian Conference on Robotics and Automation*. Canberra, Australia.

Raffo, G.V., Ortega, M.G. and Rubio, F.R., 2010. "An integral predictive/nonlinear $\mathscr{H}_\infty$ control structure for a quadrotor helicopter". *Automatica*, Vol. 46, No. 1, pp. 29–39.

Schmidt, M.D., 2011. *Simulation and Control of a Quadrotor Unmanned Aerial Vehicle*. Tese de M.Sc., Department of Electrical Engineering, University of Kentucky, Kentucky, Estados Unidos.

Waslander, S.L., Hoffmann, G.M., Jang, J.S., and Tomlin., C.J., 2005. "Multi-agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Alberta, Canada.

9. RESPONSIBILITY NOTICE

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