ABCM 40 anos
Associação Brasileira de Engenharia e Ciências Mecânicas

CONEM 2016
CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA

21 - 25
AGOSTO DE 2016
FORTALEZA - CEARÁ

Experimental Validation of a Trajectory Tracking Control using the AR.Drone Quadrotor

Ícaro Bezerra Viana, icaro@ita.br[1]
Luiz Manoel Santos Santana, luizmss@ita.br[1]
Raphael Ballet, raphaelballet@hotmail.com[1]
Davi Antônio dos Santos, davists@ita.br[1]
Luiz Carlos Sandoval Góes, goes@ita.br[1]

[1] Instituto Tecnológico de Aeronáutica, Sao José dos Campos, Brazil 12228-900

Abstract: *In this paper, we describe a hardware-in-the-loop (HIL) architecture to validate a position control for a multirotor helicopter in indoor environment. For the implementation of the proposed architecture, it is used two central computers: one dedicated to design the execution of the position control, and the another is dedicated to determine the localization of the vehicle. The proposed system separate the whole position control into an altitude and a horizontal control. To estimate the horizontal position and velocity of the multirotor, we use a static Kinect for Windows sensor fixed on the ceiling and pointing downwards, while the altitude estimation is provided by the ultrasonic sensor embedded in the vehicle. Experimental results using the low-cost quadrotor AR.Drone 2.0 validate the position control along a circular trajectory as well as a hovering flight test subject to disturbances is evaluated.*

Palavras-chave: *Aerial Robotics, AR.Drone 2, Position Control.*

1. INTRODUCTION

Research and application of Unmanned Aerial Vehicles (UAVs) increased dramatically in last two decades, mainly motivated to reduce the risk to human life, reduce costs and increase operational capabilities in military missions (Shaferman and Shima, 2008). Recently, it was the high price of quadrotors and a position capturing system that prohibited the usage of quadrotors in a bigger scale (Stevek and Miroslav, 2016). With the advent of commercial available products like AR.Drone, this platform becomes attractive for the control education in the wide range.

In this work we use Parrot AR.Drone 2 platform and a static Kinect for Windows sensor to create a low-cost position control experiment. The choice of the AR.Drone 2 micro UAV was due to the fact that this system relies on state-of-the-art indoor navigation systems combining low-cost inertial sensors, computer vision techniques, sonar, and accounting for aerodynamics models (Bristeau *et al.*, 2011). Complementarily, the Kinect sensor have two separate cameras, one that provides color images and another with an infrared-based depth image, making it possible, in this way, its use in robot tracking applications. The main motivation to propose this system is due our intention to continue carrying out experimental research in the Aerial Robotics Laboratory (LRA). The another motivation is the possibility to make a large number of possible experiments with AR.Drone 2 as illustrated in (Stevek and Miroslav, 2016).

The main objective in this paper is propose a hardware-in-the-Loop (HIL) architecture to validate position control methods for multirotor helicopters in indoor environment by performing flight tests. Before do flight experiments, we test a position control method in a complete simulator of a six degrees of freedom (6DoF) multirotor dynamic model implemented in MATLAB/Simulink. This tool is very helpful to verify the correctness of the multirotor dynamic model and to test the control performance before do flight test experiments. After, we evaluate, experimentally, the position control by using the HIL framework. As experimental environment, we use an indoor environment equipped with a low-cost system based on Kinect sensor to estimate the position and velocity of the AR.Drone 2, identified by a color mark.

The rest of the body text is organized as follows: Section 2 performs the modeling of a multirotor helicopter through differential equations and also presents the model parameters of the AR.Drone 2 used for simulation. Section 3 describes the position control method. Section 4 describes the HIL scheme. Section 5 presents the results based on computer simulations and experiments and Section 6 contains the conclusions and suggestions for future works.

2. MULTIROTOR DYNAMIC MODELING

Consider the multirotor vehicle and the three Cartesian coordinate systems (CCS). In Figure 1 is assumed that the vehicle has a rigid structure. The body CCS $S_{\rm B} \triangleq \{X_{\rm B}, Y_{\rm B}, Z_{\rm B}\}$ is fixed to the structure and its origin coincides with the center of mass (CM) of the vehicle. The reference CCS $S_{\rm G} \triangleq \{X_{\rm G}, Y_{\rm G}, Z_{\rm G}\}$ is Earth-fixed and its origin is at point O. Finally, the CCS $S_{\rm R} \triangleq \{X_{\rm R}, Y_{\rm R}, Z_{\rm R}\}$ is defined to be parallel to $S_{\rm G}$, but its origin is shifted to CM. Assume that $S_{\rm G}$ is an inertial frame.

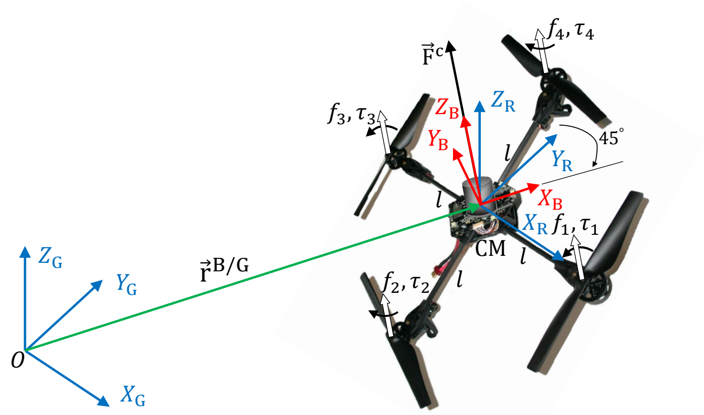

Figure 1: The Cartesian coordinate systems.

2.1 Attitude Kinematics

It can be shown that the attitude kinematics of $S_{\rm B}$ relative to $S_{\rm R}$ is modeled in matrix of attitude by the following differential equation (Shuster, 1993):

$$\dot{\mathbf{D}}^{\rm B/R} = -\left[\boldsymbol{\Omega}_{\rm B}^{\rm B/R}\times\right]\mathbf{D}^{\rm B/R}, \tag{1}$$

where, $\boldsymbol{\Omega}_{\rm B}^{\rm B/R} \triangleq [\Omega_{\rm x}\ \Omega_{\rm y}\ \Omega_{\rm z}]^{\rm T} \in \mathbb{R}^3$ is the angular velocity of the vehicle represented in $S_{\rm B}$ and $\left[\boldsymbol{\Omega}_{\rm B}^{\rm B/R}\times\right]$ is the cross-product matrix given by

$$\left[\boldsymbol{\Omega}_{\rm B}^{\rm B/R}\times\right] = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}. \tag{2}$$

2.2 Attitude Dynamics

Applying the Newton's second law of the rotational motion, obtain

$$\mathbf{T}_{\rm B} = \dot{\mathbf{H}}_{\rm B} + \boldsymbol{\Omega}_{\rm B}^{\rm B/R} \times \mathbf{H}_{\rm B}, \tag{3}$$

where the total angular momentum of the vehicle is defined by

$$\mathbf{H}_{\rm B} \triangleq \mathbf{J}_{\rm B}\boldsymbol{\Omega}_{\rm B}^{\rm B/R} + \sum_{i=1}^{4}\mathbf{H}_{\rm B}^{i}, \tag{4}$$

$\mathbf{H}_{\rm B}^{i}$ is the angular momentum of the rotor i represented in $S_{\rm B}$.

Knowing that the resultant torque $\mathbf{T}_{\rm B}$ is composed by the external control torques $\mathbf{T}_{\rm B}^{\rm c} \triangleq [T_x^c\ T_y^c\ T_z^c]^{\rm T} \in \mathbb{R}^3$ and by the disturbance torques $\mathbf{T}_{\rm B}^{\rm p} \triangleq [T_x^p\ T_y^p\ T_z^p]^{\rm T} \in \mathbb{R}^3$, and isolating the components of $\dot{\boldsymbol{\Omega}}_{\rm B}^{\rm B/R}$ in 3, the dynamic equation can be finally obtained,

$$\dot{\Omega}_x = \frac{(J_y - J_z)}{J_x}\Omega_y\Omega_z + \frac{1}{J_x}(T_x^c + T_x^p) - \frac{I_r}{J_x}\Omega_y(\omega_1 + \omega_3 - \omega_2 - \omega_4). \tag{5}$$

$$\dot{\Omega}_y = \frac{(J_z - J_x)}{J_y}\Omega_x\Omega_z + \frac{1}{J_y}(T_y^c + T_y^p) + \frac{I_r}{J_y}\Omega_x(\omega_1 + \omega_3 - \omega_2 - \omega_4). \tag{6}$$

$$\dot{\Omega}_z = \frac{(J_x - J_y)}{J_z}\Omega_x\Omega_y + \frac{1}{J_z}(T_z^c + T_z^p) + \frac{I_r}{J_z}(\dot{\omega}_1 + \dot{\omega}_3 - \dot{\omega}_2 - \dot{\omega}_4). \tag{7}$$

2.3 Translational Motion

The translation kinematics of S_{B} with relation to S_{G} was modeled on the following differential equation:

$$\dot{\mathbf{r}}_{\mathrm{G}}^{\mathrm{B/G}} = \mathbf{v}_{\mathrm{G}}^{\mathrm{B/G}}, \tag{8}$$

where $\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}} \triangleq [r_{\mathrm{x}}\ r_{\mathrm{y}}\ r_{\mathrm{z}}]^{\mathrm{T}} \in \mathbb{R}^3$ is the position vector of CM represented in S_{G} and $\mathbf{v}_{\mathrm{G}}^{\mathrm{B/G}} \triangleq [v_{\mathrm{x}}\ v_{\mathrm{y}}\ v_{\mathrm{z}}]^{\mathrm{T}} \in \mathbb{R}^3$ is the velocity vector.

Invoking the second Newton's law, the translational dynamics of the multirotor, illustrated in Figure 1, can be immediately described in S_{G} by the following second order differential equation:

$$\dot{\mathbf{v}}_{\mathrm{G}}^{\mathrm{B/G}} = \frac{1}{m}(\mathbf{D}^{\mathrm{B/R}})^{\mathrm{T}}\mathbf{F}_{\mathrm{B}}^{\mathrm{c}} + \frac{1}{m}\mathbf{F}_{\mathrm{G}}^{\mathrm{g}} + \frac{1}{m}(\mathbf{D}^{\mathrm{B/R}})^{\mathrm{T}}\mathbf{F}_{\mathrm{B}}^{\mathrm{p}}, \tag{9}$$

where $\mathbf{F}_{\mathrm{B}}^{\mathrm{c}} \triangleq [F_{\mathrm{x}}\ F_{\mathrm{y}}\ F_{\mathrm{z}}]^{\mathrm{T}} \in \mathbb{R}^3$ is the total thrust vector, m is the mass of the vehicle,

$$\mathbf{F}_{\mathrm{G}}^{\mathrm{g}} \triangleq \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{10}$$

is the gravitational force with g the gravitational acceleration and $\mathbf{F}_{\mathrm{B}}^{\mathrm{p}}$ is the disturbance force represented in S_{B}. As illustrated in Figure 1, $\mathbf{F}_{\mathrm{B}}^{\mathrm{c}}$ is perpendicular to the rotor plane.

2.4 Rotor

Dynamics and Speed Controller

The set composed of the motor model and the speed controller can be modelled in a reduced form, without losing its main features. The dynamic of this set, which has as an input signal an angular velocity $\bar{\omega}$ and output signal ω, is modelled as a first order system (Valavanis, 2007). Mathematically,

$$\frac{\omega(s)}{\bar{\omega}(s)} = \frac{k_m}{\tau_m s + 1}. \tag{11}$$

The parameter k_m is the gain of the motor and τ_m is the time constant. The value of τ_m should provide a sufficiently fast response for the actuators.

Aerodynamic coefficients

The rotor modelling consider two important aerodynamic coefficients: the thrust factor k_f and torque factor k_τ. From Blade Element Theory, both the thrust f and torque τ, created by a rotor with a linearly twisted bladed, can be related with the rotor angular velocity by (Miller, 2011).

$$f = k_f\omega^2, \tag{12}$$

$$\tau = k_\tau\omega^2, \tag{13}$$

where k_f and k_τ are coefficients that depend on the air density, the propeller geometry, angle of attack and the air flow regime.

Forces and Torques

Let the quadrotor with "X-type"configuration illustrated in Figure 1 and knowing that $\overrightarrow{F}^c$ and $\overrightarrow{T}^c$ in S_B are, respectively, the resultant force and the resultant torque of f_i and τ_i over the vehicle, with $i = 1, ..., 4$. Then the input matrix of the system can be expressed as:

$$\begin{bmatrix} F \\ \mathbf{T}_B \end{bmatrix} = \Gamma_{QX}\mathbf{f}, \tag{14}$$

Table 1: Parameters of AR Drone 2.

Variables	Values
Mass of AR Drone 2, m	0.429 kg
Lenght of arm, l	0.1785 m
Inertia for x-axes, J_x	2.237568×10^{-3} kg.m^2
Inertia for y-axes, J_y	2.985236×10^{-3} kg.m^2
Inertia for z-axes, J_z	4.80374×10^{-3} kg.m^2
Inertia of each rotor, I_r	2.029585×10^{-5} kg.m^2
Thrust factor, k_f	8.048×10^{-6} N/(rad/s)^2
Torque factor, k_τ	2.423×10^{-7} N.m/(rad/s)^2
Time constant, τ	4.718×10^{-3} s
Maximum speed, $\omega_{\max}$	1047.2 rad/s

where $\mathbf{f} \triangleq [f_1\ f_2\ f_3\ f_4]^{\mathrm{T}}$ and,

$$\Gamma_{QX} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l \\ -\frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l \\ k & -k & k & -k \end{bmatrix}. \tag{15}$$

In Equation 15, $k \triangleq \frac{k_\tau}{k_f}$ and l denotes the distance between the center of mass and the center of the rotor.

2.5 Disturbance Model

Adopt the following Gauss-Markov models for the representation of $\mathbf{F}_{\mathrm{B}}^{\mathrm{p}}$ and $\mathbf{T}_{\mathrm{B}}^{\mathrm{p}}$:

$$\dot{\mathbf{F}}_{\mathrm{B}}^{\mathrm{p}}(t) + \beta_{\mathrm{F}}\mathbf{F}_{\mathrm{B}}^{\mathrm{p}}(t) = \mathbf{w}_{\mathrm{F}}(t), \tag{16}$$

$$\dot{\mathbf{T}}_{\mathrm{B}}^{\mathrm{p}}(t) + \beta_{\mathrm{T}}\mathbf{T}_{\mathrm{B}}^{\mathrm{p}}(t) = \mathbf{w}_{\mathrm{T}}(t), \tag{17}$$

where β_{F} and β_{T} are constant parameters, $\mathbf{w}_{\mathrm{F}}$ and $\mathbf{w}_{\mathrm{T}}$ are white noise, Gaussian, with zero mean and covariance $\alpha_{\mathrm{F}}\mathbf{I}_3$ and $\alpha_{\mathrm{T}}\mathbf{I}_3$, respectively. α_{F} and α_{T} are constant parameters.

2.6 Model Parameters

The quadrotor simulator take into account the physical parameters of the AR.Drone 2. To emulate this real quadrotor platform, the present work make use of the parameters values measured in (Li, 2014) to set the simulator, as showed in Table 1.

3. POSITION CONTROL

To simplify the control problem, we separate the design of the position control into a horizontal control, maintaining constant references for the vertical motion and rotation around yaw axis. To the horizontal position control, consider the model of the translational motion presented in Section 2, and the following:

Assumption 1. In the design of the controller, the force disturbances will not be considered.

Knowing that the equation (9) is non-linear, to linearise the equation, consider the variable change:

$$\mathbf{F}_{\mathrm{B}}^{\mathrm{c}} = u - \mathbf{F}_{\mathrm{G}}^{\mathrm{g}} \tag{18}$$

In according with Assumptions 1 and using (18), a design model, for the horizontal control, can be immediately obtained explicitly from equation (9) as:

$$\begin{aligned} \dot{r}_x &= v_x \\ \dot{r}_y &= v_y \\ \dot{v}_x &= \frac{1}{m}u_x \\ \dot{v}_y &= \frac{1}{m}u_y \end{aligned} \tag{19}$$

So we can define the state vector $\mathbf{x}_i = [x_{i1}x_{i2}]^{\mathrm{T}}$ with $x_{i1} \triangleq r_i$ and $x_{i2} \triangleq v_i$, where $i = x, y$. In this case, (19) can be rewrite in a state space form as:

$$\dot{\mathbf{x}}_i = \mathbf{A}\mathbf{x}_i + \mathbf{B}^{\mathrm{c}} u_i \tag{20}$$

where,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad ; \quad \mathbf{B}^{\mathrm{c}} = \begin{bmatrix} 0 & \frac{1}{m} \end{bmatrix}. \tag{21}$$

For the horizontal control, consider the following linear control law with state feedback with control input for $x_{i1} = r_i$:

$$u_i = -\mathbf{K}_i\mathbf{x}_i + N_i\bar{r}_i \tag{22}$$

where $\bar{r}_i = e_i^{\mathrm{T}}\bar{\mathbf{r}}_{\mathrm{G}}^{\mathrm{B/G}}$, and the gains of the controller are defined as $\mathbf{K}_i \triangleq [K_{i1}K_{i2}] \in \mathbb{R}^{1\times 2}$. The gain $N_i \in \mathbb{R}$ produce zero control error in steady state and is given by $N_i = K_{i1}$ (Franklin *et al.*, 2010). The gains K_{i1} and K_{i2} that allocate the two poles of the *i*-th degree of freedom in λ_i^* are given by:

$$\begin{aligned} K_{i1} &= m(\lambda_i^*)^2, \\ K_{i2} &= -2m\lambda_i^*. \end{aligned} \tag{23}$$

To perform the altitude control a simple proportional controller were added in the architecture to control the position of the vehicle. The control signal to manipulate the yaw movement of the multirotor is calculated following a proportional control law. Figure 2 shows the block diagram of the implemented position control system.

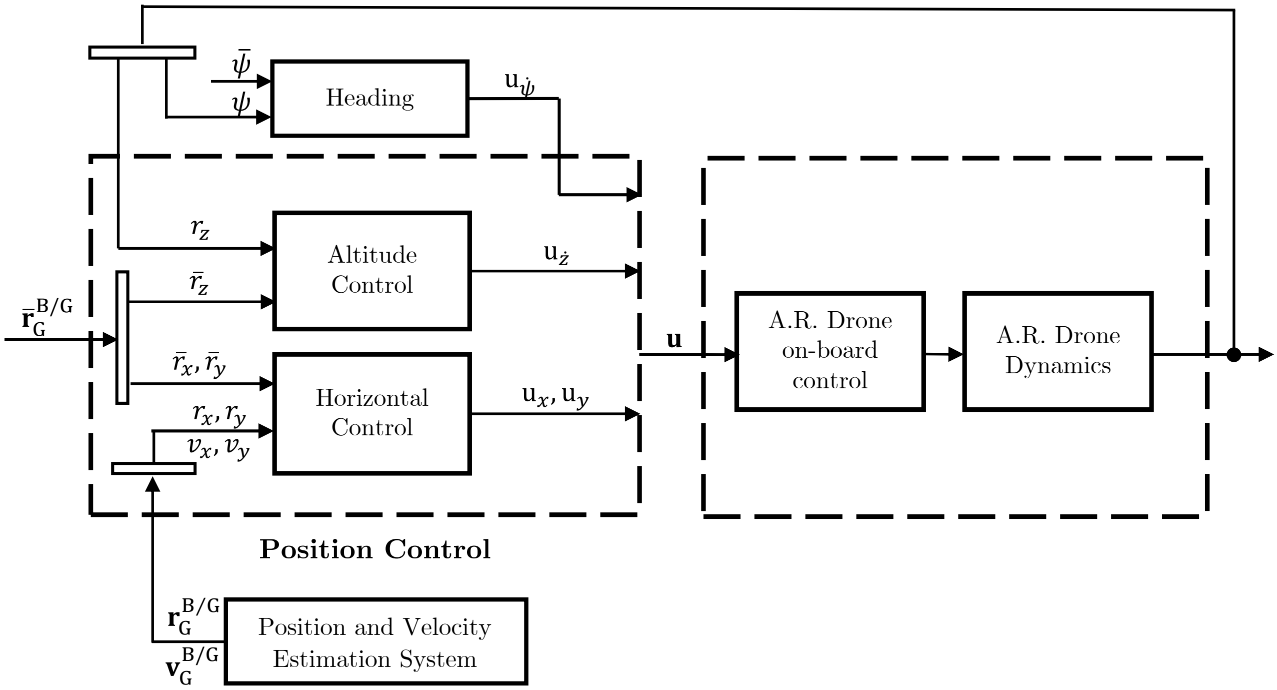

Figure 2: Position control system block diagram.

4. HARDWARE-IN-THE-LOOP SCHEME

Since the purpose of experimentally validate position control methods, this section shows some details about the hardware, software and data flow of the experiment. Here is ilustrated a scheme to control the position of one vehicle. Figure 3 illustrates the main elements of the proposed HIL architecture. The main reasons in the selection of the AR.Drone as a test platform is because this quadrotor can be easily purchased in the market at a reduced cost[1]. Moreover, the Parrot Inc. provides a set of free software tools that facilitate the development of communication and control algorithms. Details about the platform can be found in (Piskorski *et al.*, 2012).

[1] Around USD 300.00

In the following the main elements of the HIL scheme are described:

1. **Position and Velocity Estimation:** describing this part in a short form, because the system estimation based on cameras is not the object of study here, the positions and velocities of the AR.Drone 2, identified by a color mark, are obtained by using a stationary Kinect sensor fixed in the ceiling and pointing downwards, connected in a desktop computer. The algorithms to estimate position and velocity were implemented using the MATLAB software with the Image Processing Toolbox to facilitate the use of computer vision functions. First, the computer vision methods were used to identify and tracking a round red color mark attached over the AR.Drone 2. Then, the image color model from RGB color space is converted to HSV (Hue Saturation Value) color space. After the conversion, the HSV image is converted into binary image format which only shows black and white. To find the round color mark and determine its centers the Hough Transform (HT) is used. The HT method is one of the best techniques to find circular shapes if you know the approximated object range radius (Rad *et al.*, 2003). Therefore, any object with certain shape and color can be detected.

Finally, if the color mark is well detected and we knows its coordinate and size, to determine the coordinates of r_x and r_y of the vehicle we use geometric transformation in which the complete formulation is detailed in another paper that will be published. To estimate the velocities v_x and v_y, we aid a Kalman Filter by using data of position, estimated by the KINECT, to update the algorithm.

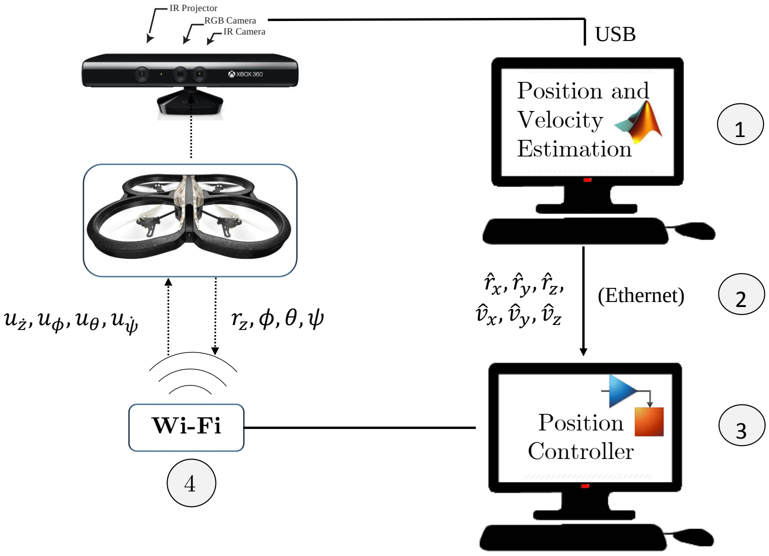
IR Projector
RGB Camera
IR Camera
USB
Position and Velocity Estimation
1
$u_{\dot{z}}, u_\phi, u_\theta, u_{\dot{\psi}}$
r_z, ϕ, θ, ψ
$\hat{r}_x, \hat{r}_y, \hat{r}_z,$
$\hat{v}_x, \hat{v}_y, \hat{v}_z$
(Ethernet)
2
Wi-Fi
4
Position Controller
3

Figure 3: Hardware-in-the-loop environment.

2. **Ethernet Network:** the two computers that compose the HIL scheme are connected by a network of Ethernet communication. Each computer is managed by the operating system Windows 7 Ultimate with Service Pack 1 and runs MATLAB/Simulink version 2015a. As showed in Figure 3, the first computer sends the data obtained from Kinect system for the position control, that are implemented in another computer of the architecture. Figure 4 summarizes the configuration of the transmit data address used in the network.

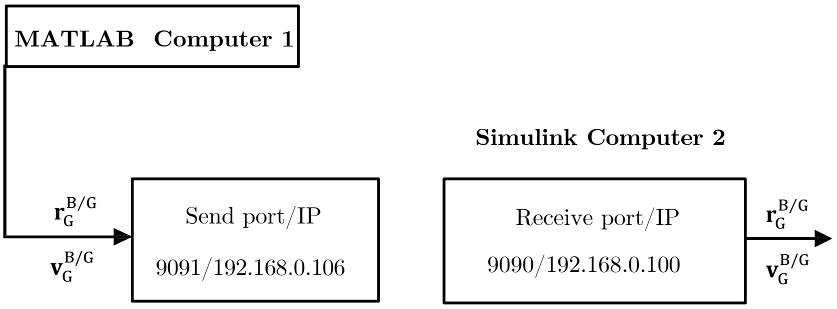

Figure 4: Ethernet network configuration.

Table 2: Parameters of the position controller.

Parameter	Value
Horizontal control, u_ϕ	$K_{x1} = 1.71$, $K_{x2} = 1.71$
Horizontal control, u_θ	$K_{y1} = 1.71$, $K_{y2} = 1.71$
Altitude control, $u_{\dot{z}}$	$K_p = 1$
Heading control, $u_{\dot{\psi}}$	$K_p = 1.5$

3. **Position Control:** design the execution of the position control in SIMULINK. All commands are computed in SIMULINK using the informations of horizontal position and velocity estimated by the Kinect system, and using the altitude z and the yaw angle ψ that are provided by the AR.Drone navigation system. Commands sent to the quadrotor onboard system are normalized and given by:

$$u = \left[u_\phi, u_\theta, u_{\dot{\psi}}, u_{\dot{z}}\right]^{\mathrm{T}} \in [-1, 1], \tag{24}$$

where u_ϕ is the reference for roll ϕ, u_θ is the reference for pitch θ, $u_{\dot{\psi}}$ is the reference for yaw rate $\dot{\psi}$ and $u_{\dot{z}}$ is the reference for vertical speed v_z. The vector **u** are transformed in references for the inner controllers of the AR.Drone. Details about the inner control loops are described in (Bristeau *et al.*, 2011).

4. **Wi-fi Communication:** the communication with AR.Drone 2 is provided by its Wi-Fi network through a UDP protocol. To do this task, we uses a free development kit[2], created from a research project of the Mathworks Inc. It is a tool with simulation blocks developed in SIMULINK that communicates with the quadrotor through its Wi-Fi network. The Wi-Fi control blocks are capable of sending commands and reading the states $\{z, \phi, \theta, \psi\}$ of the AR.Drone in real-time.

5. RESULTS

In this section, the proposed position control method is evaluated on the basis of computational simulations and experiments. In this evaluation, scenarios with different type of trajectories are tested, illustrating the capability of hovering and tracking desired values.

The simulation was implemented by using the MATLAB/SIMULINK software. The 6DOF dynamics of the AR.Drone is simulated using the Runge-Kutta 4 as the solver with an integration step of 0.001s. The physical parameters of the platform was already presented in the Table (1) of Section 2. In the (inner) attitude control loop, a saturated proportional-derivative controls laws are tuned so as to make the attitude dynamics have a bandwidth significantly larger than the bandwidth of the position control dynamics. The gains of horizontal control law were adjusted to allocate the two poles of the uncoupled horizontal position dynamics in $\lambda_i^* = -2$. The proportional gains of the altitude and heading controllers was adjusted empirically. Table 2 shows the values of the gains of the (outer) position control loop. In the simulations tests was adopted a Gauss-Markov disturbance force and torque with parameters $\alpha_{\mathrm{F}} = 0.001, \beta_{\mathrm{F}} = 0.3, \alpha_{\mathrm{T}} = 0.0001, \beta_{\mathrm{T}} = 0.03$, acting in the vehicle every time step.

To validate the position control, we conducted flight tests by using the Hardware-in-the-Loop (HIL) environment proposed in Section IV. The first test, showed in Figure 5, had a simple goal which was maintain a vehicle in hovering commanding the multirotor to the waypoint $\bar{\mathbf{r}}_{\mathrm{G}}^{\mathrm{B/G}} = [0\ 0\ 1]^{\mathrm{T}}\,m$ and keeping $\bar{\psi} = 0$. Figure 5-a shows the comparison between the simulations results that consider disturbance effects and the experimental responses. To better analyse the hovering flight, Figure5-b shows a 3D graphic, where the Kinect position estimation data is compared to the desired waypoint.

It is observed that the state feedback control action acts to make the horizontal position states to converge for their respective desired values. Note that the horizontal position error oscillate by $\pm 0.2cm$ around their desired values, that is a good precision. This oscillation is related to the accuracy range of the Kinect position estimation system and also because the vehicle operated with the indoor hull attached and close to walls, which generated little wind gusts near the region of the experiment.

Considering the same flight condition, but subjecting position disturbances in the longitudinal axis, around 0.4 m, it generated the responses to the linear position, only for the experimental flight, as illustrated in Figure 6. It is noted that even under disturbing effect, the position controller acts to correct the position error and the control system recovers the hovering condition.

[2] http://www.mathworks.com/matlabcentral/fileexchange/43719-ar-drone-simulink-development-kit-v1-1

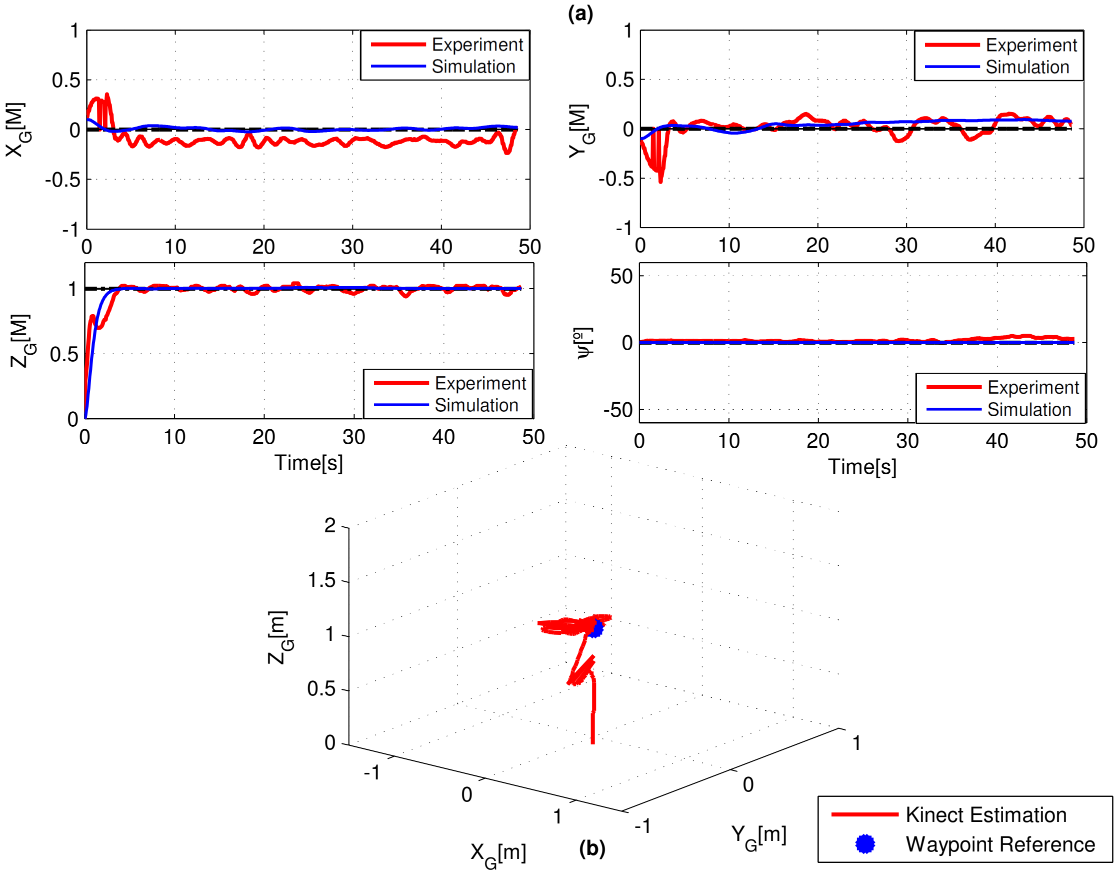

Figure 5: Hovering over a waypoint.

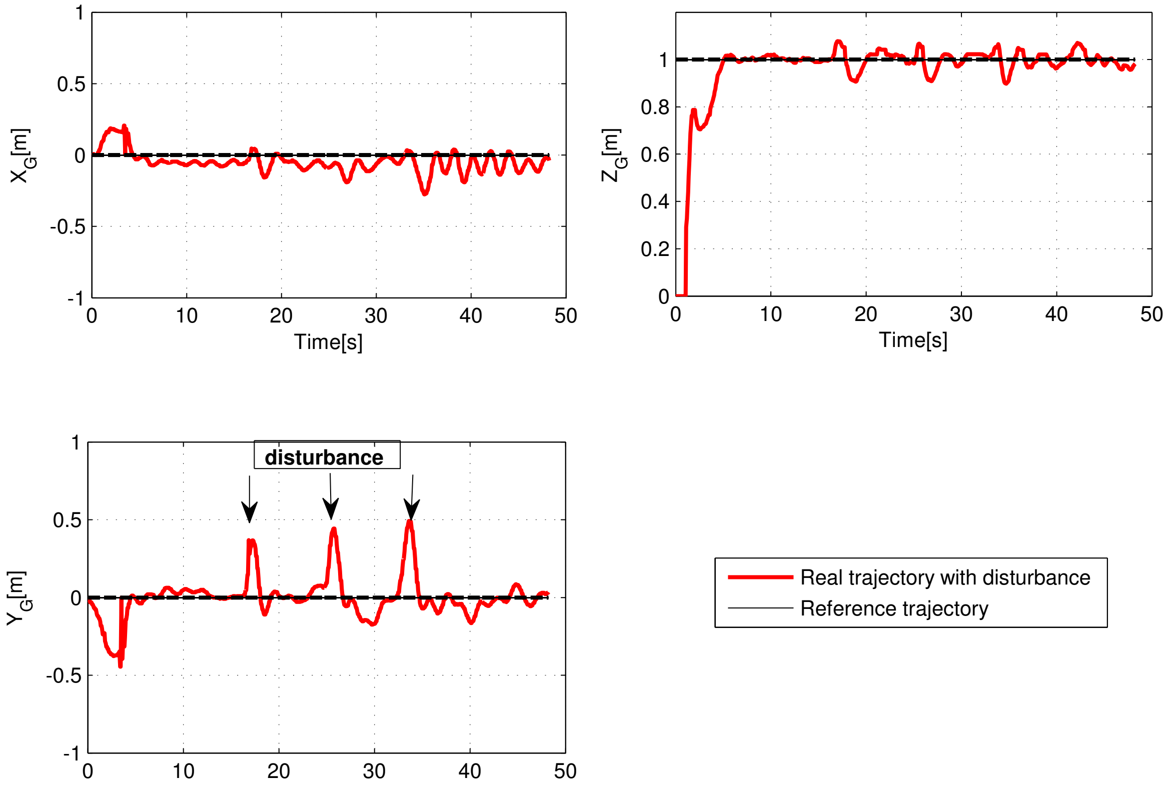

Figure 6: Hovering over a waypoint with disturbance.

Finally, to test the capability of trajectory tracking, consider a circular trajectory parametrized by,

$$\bar{r}(t) = [\bar{r}_x(t)\ \bar{r}_y(t)\ \bar{r}_z(t)]^{\mathrm{T}}, \tag{25}$$

with $\bar{r}_x(t) = 0.5cos(t)$, $\bar{r}_y(t) = 0.5sin(t)$ and $\bar{r}_z(t) = 0.6$, where $t \geq 0$ denotes the continuous time. The plot Figure in 7 shows how the vehicle track the desired circular trajectory comparing the computational simulation with the flight test experiment. In both cases, as seen in Figure 7-a, the states followed the trajectory and it is observed that the system outputs are very similar, principally, in the horizontal motion. Figure 7-b shows that the vehicle was capable to follow the circular path in many instances.

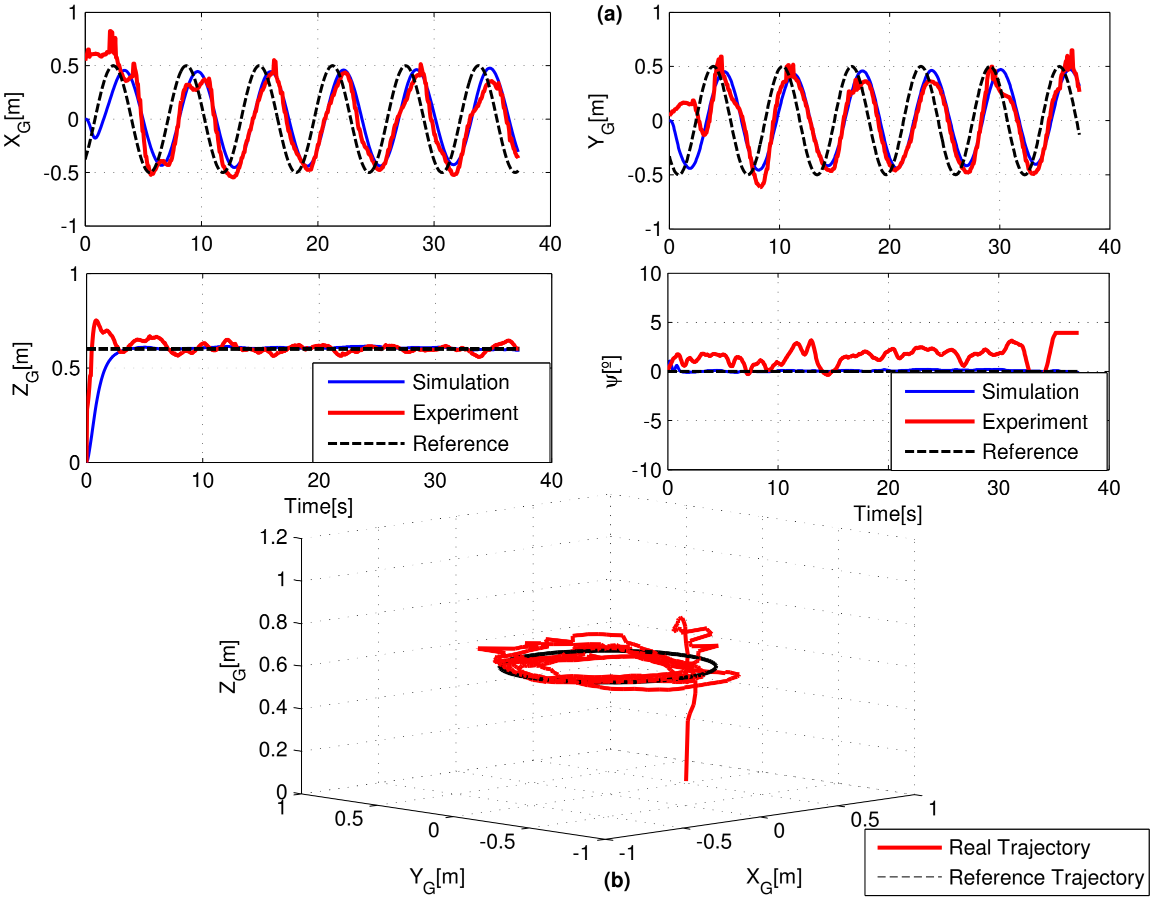

Figure 7: Circular trajectory.

6. CONCLUSIONS

This article discussed the position control problem of a multirotor aerial vehicle, focusing in use a hardware-in-the-loop (HIL) environment for experimental evaluation of the proposed method. The method was evaluated through computer simulations, considering that the vehicle was subjected to disturbance forces and torques, and by real flight tests considering hovering and a circular trajectory tracking. It is concluded that the HIL scheme was useful in validate the position control method, creating a realistic application for this new experimental setup. For a future work, it is planned validate others position control methods that requires more computational capabilities, as example of the control recently proposed by the author in (Viana *et al.*, 2015), and also extend this experiment to formation control of multiple multirotor helicopters. To evaluation of the proposed formation control, the estimation system based on cameras will be extended to determine position and velocity for two vehicles.

7. ACKNOWLEDGEMENTS

The authors acknowledge the support of Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), by research project grant 475251/2013-0 and doctoral scholarship, and the Instituto Tecnológico de Aeronáutica (ITA) by the necessary support to realize the present work.

8. REFERENCES

Bristeau, P., Callou, F. and Vissière, D., 2011. “The navigation and control technology inside the ar.drone micro uav”. In *18th IFAC World Congress*. IFAC, Milano, Italy.

Franklin, G.F., Powell, J.D. and Emami-Naeini, A., 2010. *Feedback Control of Dynamic Systems*. Pearson, 1st edition.

Li, Q., 2014. *Grey-Box System Identification of a Quadrotor Unmanned Aerial Vehicle*. M.s.thesis, Delft University.

Miller, D., 2011. *Open loop system identificaiton of a micro quadrotor helicopter from closed loop data*. M.s.thesis, University of Maryland.

Piskorski, S., Brulez, N., Eline, P. and DHaeyer, F., 2012. *AR.Drone Developer Guide*. Parrot.

Rad, A.A., Faez, K. and Qaragozlou, N., 2003. “Fast circle detection using gradient pair vectors.” In *DICTA*. pp. 879–888.

Shaferman, V. and Shima, T., 2008. “Co-evolution genetic algorithm for uav distributed tracking in urban environments”. In *9th Biennial Conference on Engineering Systems Design and Analysis*. ASME, Haifa, Israel, Vol. 1, pp. 669–677.

Shuster, M.D., 1993. “A Survey of Attitude Representations”. *Journal of the Astronautical Sciences*, Vol. 41, pp. 439–517.

Stevek, J. and Miroslav, F., 2016. “Teaching aids for laboratory experiments with ar.drone2 quadrotors”. In *11th IFAC Symposium on Advances in Control Education*. IFAC, Bratislava, Slovakia, Vol. 1.

Valavanis, K., 2007. *Advances in Unmanned Aerial Vehicles - State of the Art the Road to Autonomy*. Springer, 1st edition.

Viana, I.B., Prado, I.A.A., Santos, D.A. and Góes, L.C.S., 2015. “Trajectory tracking control of an aerial robot with obstacle avoidance”. In *Proceedings...* IFAC Symposium on Robot Control (SYROCO’2015), IFAC, pp. p.88–93.