Estimation and Control of the 3D Position of a Quadrotor in Indoor Environments

Lucas Vago Santana
Department of Industrial Automation
Federal Institute of Espírito Santo
Linhares, ES, Brazil
Email: lucas@ifes.edu.br

Mario Sarcinelli-Filho
Graduate Program on Electrical Engineering
Federal University of Espírito Santo
Vitória, ES, Brazil
Email: mario.sarcinelli@ufes.br

Ricardo Carelli
Institute of Automatics
National University of San Juan
San Juan, Argentine
Email: rcarelli@inaut.unsj.edu.ar

***Abstract*—This paper presents a method for the estimation and control of the 3D position of a quadrotor in indoor environments, which can stabilize the aircraft hovering over a reference as well as follow a moving reference. The proposed approach adopts the Extended Kalman Filter - EKF - as framework to fuse the information provided by the sensors available onboard the quadrotor, and a PD controller to regulate the flight. Experimental results are also presented, aiming at demonstrating the effectiveness of the proposed method.**

I. Introduction

The study of control techniques applied to unmanned aerial vehicles (UAV) has become a subject extensively explored in academic world, with remarkable results already published [1], [2], [3]. A meaningful part of these works has a common factor the use of rotary-wings aircrafts as experimental platform, much probably because of their versatility to perform in-flight maneuvers. However, these results are limited, in real-world applications, mostly because they are based on the use of an external computer vision system (whose cameras are not onboard the aircraft) to determine its localization. The problem in such a case, is that the aircraft can not go beyond the field of vision of the cameras. Thus, some recent studies have been devoted to investigate ways to enable the autonomous navigation of an UAV using only sensors available onboard the aircraft in indoor environments or environments in which it is not possible to get GPS data.

Amongst many works involving control of rotary-wing UAVs using external vision systems, one can cite [2] and [1]. Amongst those which use only the sensors installed onboard the vehicle one can cite [4], [5], [6] and [7]. A brief description of those works, which have inspired this research, is presented in the sequel.

In [4] a series of experiments based on servo-visual position control is presented. A model for localization and automatic navigation of the quadrotor AR.Drone 1.0 is proposed, comparing their response to control signals with the one of linear systems. The position estimation and control are achieved through geometric transformations applied to data collected by a vision system. One should notice here the constant need of viewing the target to close the control loop. However, the results are still quite interesting.

On the other hand, in [5], [6] and [7] methods that increase the robustness are discussed. These results probably represent the state of the art in position estimate and control for UAVs by using inertial sensors and computer vision. All of them explore the extended Kalman filter (EKF) as method for fusing of the data originated from the various sensors.

In [5], in particular, the visual data are obtained by extracting features of the images through employing the algorithm SURF [8] and, from a correlation between images, transforms such features into metrics that are integrated with readings coming from other sensors through an EKF. In [6] and [7], in turn, the PTAM technique [9] is adopted as the tool to extract the visual data. Although developed for augmented reality, PTAM calculates and returns the position and the orientation of the video camera that is taking the images, based on correspondence of features. As a common characteristic, however, both works use EKF for visual and inertial data integration.

In such context, this work proposes a technique inspired in those successful examples, adopting an EKF to integrate visual and inertial data, as well as a control algorithm to stabilize the UAV when hovering over a target in the ground. To discuss these two topics, the paper is hereinafter organized as follows: Section II briefly presents the experimental platform and its main characteristics, whereas Section III presents the state estimation technique to explain and discuss how the sensor data available are fused using an EKF implementation. In the sequel, Section IV discusses the control method adopted, whereas Section V shows and discusses some experimental results. Finally, in the Section VI some important observations are pointed out and some future work are outlined.

II. The AR.Drone 2.0 Platform

The experimental platform chosen in this work was the quadrotor AR.Drone 2.0, from the manufacturer Parrot. This UAV and the coordinate system adopted in this work are shown in Figure 1.

It is an autonomous aerial vehicle (a rotorcraft vehicle) commercialized as a hi-tech toy, originally designed to be controlled through smartphones and tablets via Wi-Fi network. In this way, the AR.Drone accepts control signals provided by any device compatible with this format of communication, what is the main reason for its selection as the experimental platform in this work. It is important to highlight that this

978-1-4799-2722-7/13/$31.00

Fig. 1. AR.Drone 2.0 and the coordinate systems adopted ($\{w\}$ is the global coordinate system and $\{b\}$ is the coordinate system of the vehicle).

platform is easily purchased in the market, at a reduced cost[1], if compared to similar platforms. Moreover, Parrot provides a set of software tools, which facilitates developing control algorithms for the AR.Drone platform (additional details can be found in [10]).

A. Sensors Data

The AR.Drone 2.0 comes from the factory equipped with accelerometers, gyroscopes, magnetometers, two video cameras and an onboard computer that manages the data coming from such sensors and the wireless communication network of the vehicle. The firmware installed on the UAV is capable of performing automatically the procedures of take-off, landing and flight stabilization, besides responding to external motion commands, upon receiving one. Thus, as result of the internal algorithms, the set of variables $\mathbf{q} = [z \quad v_x \quad v_y \quad \phi \quad \theta \quad \psi]$ becomes available for the developer, where

- z represents the altitude of the vehicle relative to the ground underneath him;
- v_x and v_y represent the linear velocities developed by the vehicle, relative to the reference axes x_b and y_b;
- ϕ, θ and ψ represent the inclination angles of the AR.Drone, all referenced to the global coordinate system.

Such information, as well as the access to them, are discussed in [10]. In [11] the techniques internally applied for obtaining such information are discussed. Therefore, it is recommended to the interested reader to check these references to get more knowledge about the technology involved in the firmware of the AR.Drone.

B. AR.Drone Motion Control

The motion commands for the AR.Drone are sent, under a specific protocol, over its Wi-Fi network. In this protocol, the command signals are normalized, and arranged as elements of the vector of control signals $\mathbf{u} = \begin{bmatrix} u_{\dot{z}} & u_{\dot{\psi}} & u_{\phi} & u_{\theta} \end{bmatrix} \in [-1.0, +1.0]$, where

- $u_{\dot{z}}$ represents a command of linear velocity, which causes displacements over the z_w axis;
- $u_{\dot{\psi}}$ represents a command of angular velocity, which causes rotations around the z_w axis;
- u_{ϕ} represents a command of inclination, related to the x_w axis;
- u_{θ} represents a command of inclination, related to the y_w axis.

Likewise reported in [4] and [12], it is possible to assume that the response of the UAV to the application of a control signal is linear. This way, such signals ($\mathbf{u}$) are transformed in references to the internal controllers of the AR.Drone. In numeric terms, this means that there is an upper bound internally established for the linear velocity in the direction z, which is $v_{z_{max}} = 0.7$ m/s, and that upon receiving the command $u_{\dot{z}} = 0.5$, the AR.Drone will perform a velocity $v_z = 0.35$ m/s. The same reasoning is valid for the other control signals in the vector $\mathbf{u}$.

Despite these characteristics, the AR.Drone does not have the capability of keeping hovering in a completely autonomous way. The take-off and landing maneuvers happen autonomously, but once in the air the UAV starts sliding away from its initial position. This effect is known as drifting in the literature, and results mostly from measurement errors integrated along time. Thus, to ensure better performance when hovering in a given 3D reference as well as when tracking a mobile target, it is necessary to estimate its position and to correct the position error through a closed loop control system.

III. 3D Position Estimation via EKF

The Extended Kalman Filter has been applied as a tool for state estimation in robot control [13]. The filter requires a model of state transition and an observation model, in the form

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k \end{aligned} \tag{1}$$

where f and h are differentiable nonlinear functions, and $\mathbf{w}$ and $\mathbf{v}$ represent Gaussian noises with zero average. The subscript k refers to a discrete time instant.

For its implementation, the first step is to define the states to be estimated by the filter, in this case

$$\mathbf{x}_k = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & v_x & v_y & \phi & \theta & \psi & \dot{\psi} \end{bmatrix}^T, \tag{2}$$

where x, y, z, $\dot{x}$, $\dot{y}$ and $\dot{z}$ are the positions, in (m), and the linear velocities, in (m/s), of the AR.Drone in the three directions of the global reference system. As for v_x and v_y, they are the linear velocities developed by the vehicle, in its own reference system, in (m/s). Finally, ϕ, θ and ψ are the angular positions, in (rad), and $\dot{\psi}$ is the angular velocity around the z axis, in (rad/s).

A. State Prediction Model

The prediction model f used in the EKF structure defines how a state evolves from an instant of time to the next one. The equations here adopted are

[1]Currently around three hundred American dollars - USD 300.00

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \dot{z}_{k+1} \\ v_{x_{k+1}} \\ v_{y_{k+1}} \\ \phi_{k+1} \\ \theta_{k+1} \\ \psi_{k+1} \\ \dot{\psi}_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \delta t \cdot \dot{x}_k \\ y_k + \delta t \cdot \dot{y}_k \\ z_k + \delta t \cdot \dot{z}_k \\ c_{\psi_k} \cdot v_{x_k} - s_{\psi_k} \cdot v_{y_k} \\ s_{\psi_k} \cdot v_{x_k} + c_{\psi_k} \cdot v_{y_k} \\ \dot{z} + \delta t \cdot \ddot{z}_k(\mathbf{x}, \mathbf{u}) \\ v_{x_k} + \delta t \cdot \dot{v}_{x_k}(\mathbf{x}) \\ v_{y_k} + \delta t \cdot \dot{v}_{y_k}(\mathbf{x}) \\ \phi_k + \delta t \cdot \dot{\phi}_k(\mathbf{x}, \mathbf{u}) \\ \theta_k + \delta t \cdot \dot{\theta}_k(\mathbf{x}, \mathbf{u}) \\ \psi_k + \delta t \cdot \dot{\psi}_k \\ \dot{\psi}_k + \delta t \cdot \ddot{\psi}_k(\mathbf{x}, \mathbf{u}) \end{bmatrix}, \tag{3}$$

where δt represents the sampling interval, and c_ψ and s_ψ stands for $\cos(\psi)$ and $\sin(\psi)$, respectively. It is noteworthy that such models were obtained observing the behavior of the AR.Drone in flight and taking into consideration the reports in [4], [6] and [12].

Due to the underactuated nature of the quadrotor [14], it is assumed that the horizontal accelerations $\dot{v}_x$ and $\dot{v}_y$ are generated indirectly by the states $\mathbf{x}$. To model this, it was considered that the propulsion force (f_{prop}) resultant from the four rotors is constant, such that $\dot{v}_x$ and $\dot{v}_y$ are modeled as functions only of the inclination of the vehicle and the drag force of the air (f_{drag}), here considered as of magnitude proportional to the speed and in the opposite direction, i.e.,

$$\begin{aligned} \dot{v}_x &= f_{prop_x} - f_{drag_x} \\ \dot{v}_y &= f_{prop_y} - f_{drag_y}, \end{aligned} \tag{4}$$

or

$$\begin{aligned} \dot{v}_x &= K_1(s_\psi s_\phi c_\theta + c_\psi s_\theta) - K_2 v_x \\ \dot{v}_y &= K_3(-c_\psi s_\phi c_\theta + s_\psi s_\theta) - K_4 v_y, \end{aligned} \tag{5}$$

where the constants K_1, K_2, K_3 and K_4 represent coefficients of proportionality which should be experimentally identified.

Similarly, the influence of the control signals $\mathbf{u} = \begin{bmatrix} u_{\dot{z}} & u_{\dot{\psi}} & u_\phi & u_\theta \end{bmatrix}$ is incorporated to the prediction model through the calculation of $\ddot{z}(\mathbf{x}, \mathbf{u})$, $\ddot{\psi}(\mathbf{x}, \mathbf{u})$, $\dot{\phi}(\mathbf{x}, \mathbf{u})$ and $\dot{\theta}(\mathbf{x}, \mathbf{u})$, as the linear systems

$$\begin{aligned} \ddot{z}(\mathbf{x}, \mathbf{u}) &= K_5 u_{\dot{z}} - K_6 \dot{z} \\ \ddot{\psi}(\mathbf{x}, \mathbf{u}) &= K_7 u_{\dot{\psi}} - K_8 \dot{\psi} \\ \dot{\phi}(\mathbf{x}, \mathbf{u}) &= K_9 u_\phi - K_{10} \phi \\ \dot{\theta}(\mathbf{x}, \mathbf{u}) &= K_{11} u_\theta - K_{12} \theta, \end{aligned} \tag{6}$$

where the constants K_5 to K_{12} also represent proportionality coefficients and should be experimentally identified.

It is important to highlight that at no moment it was stated that the proposed model is complete. However, despite the simplifying considerations it is still good enough to provide the results needed to achieve the automatic position control either in a hover or in an object tracking flight, as it can be seen in the sequence of the paper.

B. State Observation Model

To complete the filter it is necessary to establish a model of observation for the states. In this paper, the method employed was used to establish two distinct models: $\mathbf{h}_1$, to be used when there is no visual information available, and $\mathbf{h}_2$ for use when there is visual information available, being

$$\mathbf{h}_1 = \begin{bmatrix} z \\ v_x \\ v_y \\ \phi \\ \theta \\ \psi \end{bmatrix} \quad \text{and} \quad \mathbf{h}_2 = \begin{bmatrix} x_w \\ y_w \\ z \\ v_x \\ v_y \\ \phi \\ \theta \\ \psi \end{bmatrix}, \tag{7}$$

where z, v_x, v_y, ϕ, θ and ψ are dealt with as direct observations from the data provided by the AR.Drone firmware, while x_w and y_w represents the global coordinates of the vehicle in these directions. Such readings are obtained as described in Section III-C.

Thus, when there is visual information available, it is possible to use it and to correct the rotorcraft position, or, at least, to decrease the position uncertainty. Note, however, that even if this does not happen, the position continues to be estimated, but with the cost of a greater uncertainty.

C. Visual Data Estimation

The coordinates x_w and y_w represents the point where the AR.Drone is positioned, with respect to a target at the origin of the global coordinate system. Surely this estimate includes uncertainties and noise, but provides data for updating the states in the EKF. Figure 2 illustrates the relationship between the coordinate systems, as shown in the target.

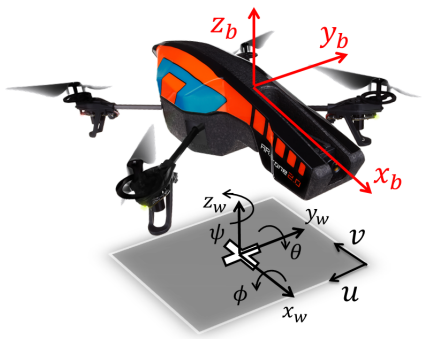

Fig. 2. Relationship between the image, the AR.Drone and the global coordinate systems.

For calculating the x_w and y_w values, the applied technique makes use of algorithms present in the OpenCV library. The motivation for adopting it comes first from its free license and public access and, following, from the reliable results obtained with the implementation of some of its algorithms. As a reference in its use, a study was conducted in [15] where it was found out the possibility of accomplishing a color filtering associated to a centroid search. Thus, what is done as visual

processing is to capture an image with the bottom camera of the AR.Drone and seek for green color elements in such image. When an area with a certain amount of green pixels is detected, the coordinates of the centroid of such area are calculated through using the Hu's invariant moment algorithm [16]. Once the coordinates of the centroid are obtained, the following equation is applied

$$\begin{bmatrix} x_w \\ y_w \end{bmatrix} = z \cdot \begin{bmatrix} c_\psi & -s_\psi \\ s_\psi & c_\psi \end{bmatrix} \cdot \left(\begin{bmatrix} 0 & \frac{1}{fs_y} \\ \frac{1}{fs_x} & 0 \end{bmatrix} \cdot \begin{bmatrix} u - c_u \\ v - c_v \end{bmatrix} + \begin{bmatrix} s_\theta \\ -s_\phi \end{bmatrix} \right), \quad (8)$$

from where the position of the camera in relation to the target centroid is obtained, i.e., x_w and y_w are obtained, considering the origin of the AR.Drone coincident with the camera. In this equation, u and v are the centroid coordinates in the image (pixels), while c_u, c_v, are the image center coordinates, fs_x and fs_y are the scale factors of the camera intrinsic parameters, all obtained through calibration.

It is important to say that despite being simple, the proposed visual method allied with the inertial sensors trough the EKF is able to guarantee the necessary robustness to control the AR.Drone position in a hover flight and, additionally, to allow the rotorcraft to track and follow the target under its movement. Moreover, the visual data is extracted without needing to use any black box algorithm, as would be the case of SURF and PTAM algorithms.

IV. Position Control

For controlling the 3D position of the rotorcraft, the method adopted was the PD controller with compensation for rotations in the angle ψ. Thus, given a reference in the three-dimensional space $\mathbf{P}_d = [x_d \quad y_d \quad z_d \quad \psi_d]$, along with the EKF output, the control is carried out as

$$\begin{aligned} u_{\dot{z}} &= K_{p1}(z_d - z) - K_{d1}\dot{z} \\ u_{\dot{\psi}} &= K_{p2}(\psi_d - \psi) - K_{d2}\dot{\psi} \\ u_\phi^{'} &= K_{p3}(y_d - y) - K_{d3}\dot{y} \\ u_\theta^{'} &= K_{p4}(x_d - x) - K_{d4}\dot{x} \\ u_\phi &= u_\phi^{'} s_\psi - u_\theta^{'} c_\psi \\ u_\theta &= u_\phi^{'} c_\psi + u_\theta^{'} s_\psi. \end{aligned}$$

The performance of this PD controller has proven to be enough in the regulation of a desired position, achieving the convergence of the errors to zero. This is due to the nature of the AR.Drone's response to the control signals. Observe that $u_{\dot{z}}$ and $u_{\dot{\psi}}$ already represents velocity commands. Therefore, this behavior is understood as an integral nature, which, when associated to the controller, tends to eliminate the steady state errors. On their turn, u_ϕ and u_θ, if well analyzed, also behave in the same way, because they indirectly represent velocity gains in the directions y_b and x_b, respectively, since the ψ rotation is compensated. That is, u_ϕ causes a speed $-v_y$ and u_θ causes a velocity $+v_x$. The reader's attention is called to the command u_ϕ, whose reaction is contrary to speed gain towards the y_b direction, what explains the presence of the negative sign.

It is important to say that if the desired positions are $x_d = 0.0$ and $y_d = 0.0$ the target point coincides with the center of the colored object, positioned in the origin of the global coordinate system. So, if the colored target is moved the tendency of the AR.Drone is to follow the object, trying to achieve its desired relative position. This way, the system here proposed is also capable of making the rotorcraft to track a reference target.

V. Experimental Results

This section presents the experimental results obtained for position and orientation control executed by the method described in Section IV. The comparative lines in the graphics were plotted through using the position and orientation data outputted by the EKF ($\mathbf{P}_{ekf} = [x_f \quad y_f \quad z_f \quad \psi_f]$) and the same variables, now obtained using the odometry of the vehicle ($\mathbf{P}_{odm} = [x_o \quad y_o \quad z \quad \psi]$), where x_o and y_o were obtained by means of integrating and rotating the velocities v_x and v_y, while z and ψ were directly collected from the raw values provided by the AR.Drone firmware.

The experiment 1, characterized in Figure 3, had the simple goal of keeping the rotorcraft anchored over the referential, with $\mathbf{P}_d = [0.0 \quad 0.0 \quad 1.5 \quad 0.0]$. Note that without the visual correction provided for the proposed system, the odometry tends to provide values with an error that increases along time. This is due to the effect of drifting, where numeric errors, accumulated in the integration of the speed signals, increase with time.

The experiment 2, illustrated in Figure 4, was carried out with the same control objective of the experiment 1. The difference is that there were position disturbances imposed to the rotorcraft during the flight, in the order of 0.5m, moving the AR.Drone away from the hovering reference. It is interesting to point out, even without mathematical analysis in this sense, the robustness of the system: the rotorcraft is able to recover its hovering position even after the visual target disappears of the field of vision of the camera onboard it. This is guaranteed by the use of odometry when no visual information is available.

The experiment 3, illustrated in Figure 5 is more complete. Its goal was to achieve the desired position $\mathbf{P}_d = [0.0 \quad 0.0 \quad 1.2 \quad -\pi/4]$. During the execution of the experiment, the room lights were turned off and on. Then the AR.Drone was subject to disturbances in terms of position. In this experiment it is important highlight that the rotation compensation for the PD controller made possible the realization of the flight in a referential $\psi_d \neq 0$.

Figure 6 illustrates the experiment 4. The control goal was initially established as $\mathbf{P}_d = [0.0 \quad 0.0 \quad 1.3 \quad 0.0]$. However, the visual target is moved, through a displacement over the x_w axis. Figure 6 shows how the proposed control system reacts: it tries to achieve the desired position all the time. Indeed, the motion of the target in the world was interpreted by the control system as a position error, which needed to be corrected. As a result, the AR.Drone follows the object, trying to achieve the control goal. Thus, the proposed system is able to make the rotorcraft to follow a moving target.

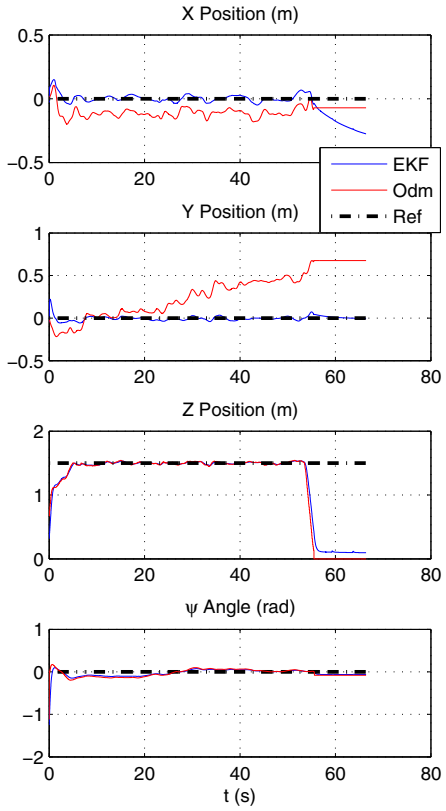

Fig. 3. Results of experiment 1 - hovering.

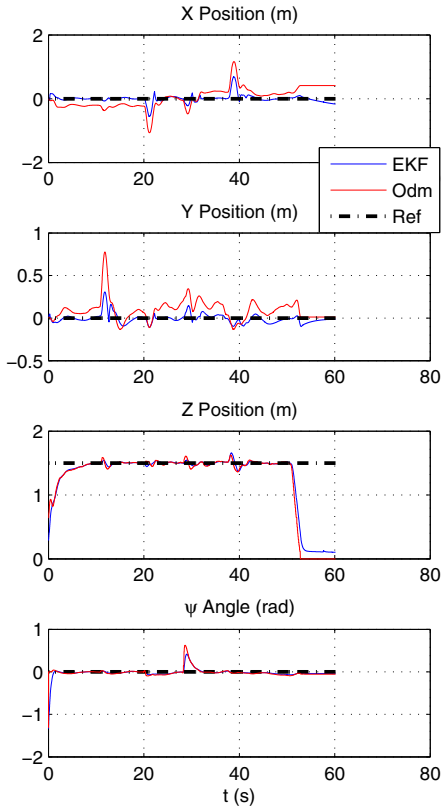

Fig. 4. Results of experiment 2 - disturbed hovering.

Finally, for a better comprehension of this section, the reader is invited to watch the videos of the experiments that originated the plots presented here, in the following links:

http://youtu.be/HQeNkbyDoY8,

http://youtu.be/HYOvbKORHhw,

http://youtu.be/_DwUfg7TTTU and

http://youtu.be/PCUNDuYpeik.

VI. Conclusion and Future Work

The method proposed in this work adds robustness to the problem of hovering over an object, for being based on two sensorial information, knowing the visual and inertial information. The method adopts the fusion of both sensorial data, thus compensating for the problems associated to the inertial sensor and for the problems associated to computer vision as well (bad illumination and the situation in which the target is out of the field of view). As shown through the experiments here reported, the proposed procedure allows the rotorcraft to momentarily loose the target object from its field of vision, retrieving that information later.

As seen in Section V, the estimation method can be used to control the position over a three-dimensional referential, even under orientations ψ_d different of zero and if it is submitted to disturbances. Additionally, it was shown the inherent characteristic of the system to follow a reference that is moving in the world. In the future, it is intended to apply this inherent characteristic into a decentralized leader-follower cooperation system: the idea is that using just its own sensors the UAV will be capable to estimate the data correspondent to the position, velocity and orientation of a terrestrial robot, which are essential to solve this kind of problem, and thus to control its own position, velocity and orientation in order to keep a formation with the terrestrial robot.

Surely some practical limitations require attention and treatment, to generate best control results. Amongst them, it is important to highlight the need to deal with the delays in communication, originated by the Wi-Fi network or inherent to the system, like the notorious differences in the update rate between the inertial and visual data. In spite of that, however, the proposed method has proven to be effective in terms of accomplishment of the proposed task.

Acknowledgment

The authors thank CNPq - a Brazilian agency that supports scientific and technological development (grant 473185/2012-1) - for the financial support granted. They also thank the Federal Institute of Espírito Santo and the Federal University of Espírito Santo for supporting the development of this research. Dr. Sarcinelli Filho also thanks the additional financial support

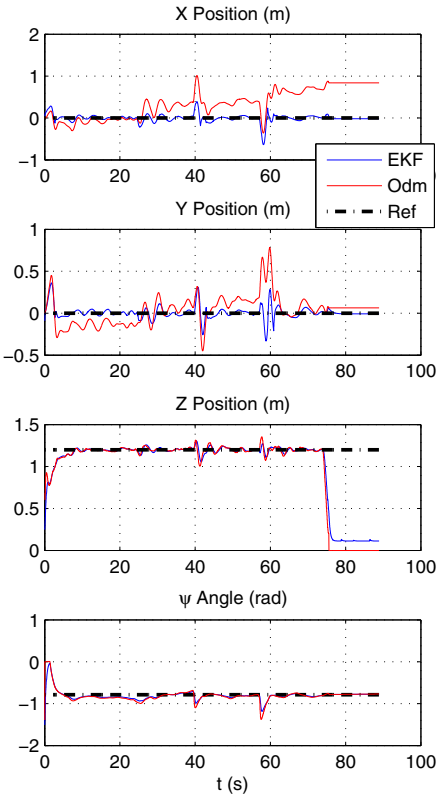

Fig. 5. Results of experiment 3 - hovering with $\psi_d \neq 0$.

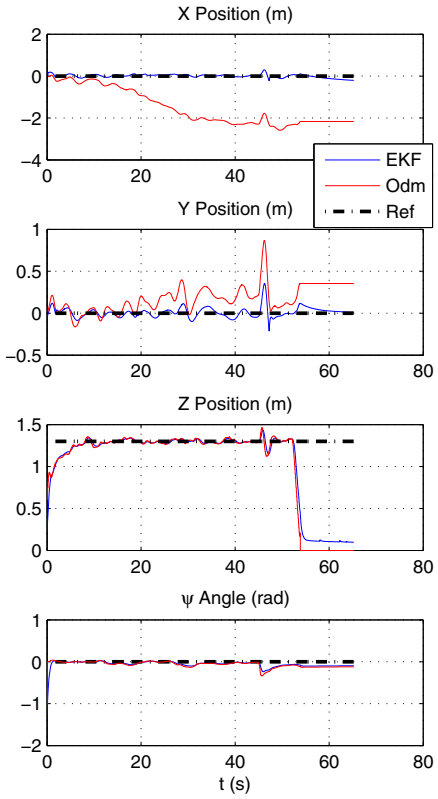

Fig. 6. Results of experiment 4 - following a moving target.

of FAPES - Fundação de Amparo à Pesquisa do Espírito Santo to the project.

REFERENCES

[1] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 2520–2525.

[2] M. W. Müller, S. Lupashin, and R. D'Andrea, "Quadrocopter ball juggling," in *Proceedings of the 24th IEEE/RSJ International Conference on Intelligent Robot Systems*, San Francisco, USA, September 2011, pp. 5113–5120.

[3] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart, "Versatile distributed pose estimation and sensor self-calibration for an autonomous mav," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, St. Paul, MN, USA, May 2012, pp. 31–38.

[4] T. Krajnik, V. Vonasek, D. Fiser, and J. Faigl, "Ar-drone as a platform for robotic research and education," in *Research and Education in Robotics - EUROBOT 2011*, ser. Communications in Computer and Information Science, D. Obdrzalek and A. Gottscheber, Eds. Springer, 2011, vol. 161, pp. 172–186.

[5] N. Dijkshoorn and A. Visser, "Integrating sensor and motion models to localize an autonomous ar.drone," *International Journal of Micro Air Vehicles*, vol. 3, no. 4, pp. 183–200, December 2011.

[6] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, October 2012, pp. 2815–2821.

[7] S. M. Weiss, "Vision based navigation for micro helicopters," PhD Thesis, Department of Electrical Engineering and Information Technology, Eidgenössische Technische Hochschule (ETH), Zurich, Switzerland, 2013.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.

[9] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *Proceedings of the Eigth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*. Washington, DC, USA: IEEE Computer Society, October 2009, pp. 83–86.

[10] S. Piskorski, N. Brulez, P. Eline, and F. DHaeyer, *AR.Drone Developer Guide*, Parrot, December 2012, sDK Version 2.0.

[11] P.-J. Bristeau, F. Callou, D. Vissiere, and N. Petit, "The navigation and control technology inside the ar.drone micro uav," in *Proceedings of the 18th IFAC World Congress*, vol. 18, Milan, Italy, August-September 2011, pp. 1477–1484.

[12] I. Sa, H. He, V. Huynh, and P. Corke, "Monocular Vision based Autonomous Navigation for a Cost-Effective Open-Source MAVs in GPS-denied Environments," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Wollongong, Australia, July 2013.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[14] A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "High-level underactuated nonlinear control for rotorcraft machines," in *Proceedings of the IEEE International Conference on Mechatronics*. Vicenza, Italy: IEEE, Febrary 27 – March 1 2013, pp. 279–285.

[15] G. R. Bradski and A. Kaehler, *Learning OpenCV - computer vision with the OpenCV library: software that sees*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2008.

[16] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.