Journal of Intelligent & Robotic Systems (2019) 93:5–16
https://doi.org/10.1007/s10846-018-0799-3

An Adaptive Dynamic Controller for Quadrotor to Perform Trajectory Tracking Tasks

Milton Cesar Paes Santos[1] · **Claudio Darío Rosales**[2] · **Jorge Antonio Sarapura**[2] · **Mário Sarcinelli-Filho**[3] · **Ricardo Carelli**[2]

Received: 3 October 2017 / Accepted: 21 February 2018 / Published online: 12 March 2018

Abstract

This work proposes an adaptive dynamic controller to guide an unmanned aerial vehicle (UAV) when accomplishing trajectory tracking tasks. The controller structure consists of a kinematic controller that generates reference commands to a dynamic compensator in charge of changing the reference commands according to the system dynamics. The final control actions thus generated are then sent to the UAV to make it to track an arbitrary trajectory in the 3D space. The parameters of the dynamic compensator are directly updated during navigation, configuring a directly updated self-tuning regulator with input error, aiming at reducing the tracking errors, thus improving the system performance in task accomplishment. After describing the control system thus designed, its stability is proved using the Lyapunov theory. To validate the proposed system simulations and real experiments were run, some of them are reported here, whose results demonstrate the effectiveness of the proposed control system and its good performance, even when the initial values of the parameters associated to the dynamic model of the UAV are completely unknown. One of the conclusions, regarding the results obtained, is that the proposed system can be used as if it were an on-line identification subsystem, since the parameters converge to values that effectively represent the UAV dynamics.

Keywords Adaptive control · Trajectory tracking · Quadrotor · Lyapunov theory

1 Introduction

Research on unmanned aerial vehicles (UAVs) has been deserving increasing interest in recent years, due to the variety of tasks these robots can perform, such as surveillance and monitoring, inspection of crop rows and automatic spraying in precision agriculture, search and rescue in disasters management, for instance [2, 9]. Among such robots, the rotorcraft-type UAVs, such as the quadrotors, have played an important role, due to their suitability for many applications.

Quadrotors are four-rotor vehicles whose rotational and translational motions are performed through varying the speed of the four rotors in a coordinated way, to produce the desired maneuver. The main benefit of quadrotor-type UAVs is that they have simpler structure than helicopters [23], thus making easier to develop control systems for such vehicles. However, according to [12] there are some challenges for the designers of control systems for quadrotors: the vehicle dynamics is multiple-input multiple-output, highly nonlinear and highly coupled, and involves various uncertainty sources, including parametric uncertainties. As

✉ Milton Cesar Paes Santos
milton.santos@ifes.edu.br

Claudio Darío Rosales
crosales@inaut.unsj.edu.ar

Jorge Antonio Sarapura
jsarapura@inaut.unsj.edu.ar

Mário Sarcinelli-Filho
mario.sarcinelli@ufes.br

Ricardo Carelli
rcarelli@inaut.unsj.edu.ar

[1] Federal Institute of Espírito Santo, Rodovia ES-080, Km 93, São João de Petrópolis, Santa Teresa, ES 29660-000, Brazil

[2] Institute of Automatics, National University of San Juan and CONICET, Av. Libertador San Martin, 1112 (O), San Juan, Argentina

[3] Department of Electrical Engineering, Federal University of Espírito Santo, Av. Fernando Ferrari, 514, Goiabeiras, Vitória, ES, Brazil

a matter of fact, the parameters modeling the dynamics of such vehicles are difficult to measure and can even vary with the environmental conditions or the task itself.

Most control systems for UAVs, and also for quadrotors, found in the literature are based on complex mathematical models and algorithms, such as in [8], which demand low rates of commands updating (50 to 200 Hz) for its execution. As a consequence the UAV thus controlled only executes slow maneuvers. In addition, low refresh rates require precise models and exact algorithms because small linearization errors can result in instability of the UAV [6] [4, 13]. Other approaches, such as [7], employ simpler models, using optimized components at a high refresh rate (1 kHz) for quick reactions, high stability and maneuverability. However, this approach requires a high knowledge and a high degree of accessibility to the hardware and software of the aerial platform, which is not always possible when dealing with commercial vehicles. Still considering model based control, the authors of [15] proposed to use the model reference adaptive control (MRAC) technique to control a quadrotor under various conditions, with parametric and non-parametric uncertainties in the model. An accurate simulation, including empirical dynamic models of battery, sensors, and actuators was performed to validate the stability of the closed loop system. However, such a paper only studied the problem of controlling the altitude and attitude of the quadrotor.

Thus, to control and stabilize a UAV requires a controller whose control algorithm provides robustness in the presence of uncertain parameters, external perturbations and non-modeled dynamics. In this sense, the field of adaptive control provides approaches and solutions that allow taking into account the unknown system dynamics and determining uncertain parameters during the task execution, improving the robustness of the control system while the aerial vehicle performs the task assigned to it. This is the case in [1], an extension of [7], which uses the adaptive reference model control combined with a nonlinear control structure based on the inverse dynamics method at high updating rates, with a complete dynamic model of the UAV ensuring long-term stability and robustness against non-modeled dynamics. Other schemes employing a complex dynamic model of the UAV are presented in [3, 5]. The first one presents a composite adaptive control where the dynamic model of the aerial vehicle is divided into two subsystems, using inverse dynamics and a sliding mode to control the internal dynamics, while an adaptive controller controls the completely actuated system assuming unknown parameters. As for the second one, it uses an adaptive sliding mode control to stabilize the pose and the trajectory of a UAV considering uncertain parameters. Both works validate their schemes only through simulations. By its turn, in [14] the authors present an adaptive technique also based on a dynamic model but applied to the control of a terrestrial robot that carries an unknown and variable load. In [21] a different scheme is developed using an ANFIS model with the combined benefits of neural network techniques in the absence of a complex dynamic model.

In such a context, this work proposes a nonlinear controller for trajectory tracking tasks accomplished by a quadrotor, using a simplified dynamic model whose parameters can be roughly known or even completely unknown and an adaptive controller represented by a directly tuned self-tuning regulator with control error directly inputted [22], whose parameter adaptation law is responsible for reducing the tracking errors due to the simplified dynamics by adjusting the unknown dynamic parameters of the simplified model. The objective is to provide robustness to the non-modeled dynamics and the uncertainties in the model parameters. The controller thus designed allows the tracking of arbitrary trajectories in the 3D space with a certain UAV orientation. The use of a simplified model with parameters that do not need to be known before executing the task represents a great advantage of our method in comparison with the methods above mentioned: it allows faster refresh rates than those that use exhaustive and complex dynamic models, and allows getting great robustness during the tracking of any trajectory in the 3D space by adjusting a minimum set of parameters, whose consequence is a meaningful reduction in the tracking errors, mainly for trajectories that excites the dynamics of the vehicle.

To give details of the control system here proposed, this work is hereinafter organized as follows: Section 2 presents the simplified kinematic and dynamic models of a quadrotor, whereas Section 3 brings an analysis of the control structure adopted, which involves the kinematic controller for the task of trajectory tracking, the dynamic controller adopted to deal with the dynamics of the quadrotor and the parameter adaptation law adopted to reduce control errors. In the sequel Sections 4 and 5 present, respectively, simulation and experimental results that validate the proposed approach. Finally, Section 6 points out some conclusions about the proposal.

2 Mathematical Model

This work considers the commercial quadrotor AR.Drone 2.0 Power Edition®, from Parrot Inc., which has been used by several research groups from various universities around the world [11]. Figure 1 shows the UAV together with the coordinate systems adopted.

Such a platform is an autonomous aerial vehicle (a rotorcraft one) commercialized as a hi-tech toy, originally designed to be controlled through smart-phones or tablets via

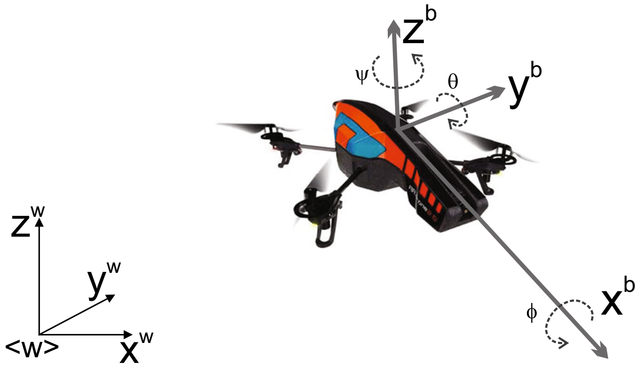


Fig. 1 The inertial frame $\langle w\rangle$ and the mobile frame $\langle b\rangle$ attached to the body of the quadrotor *AR.Drone 2.0 Power Edition*®

a Wi-Fi network, with specific communication protocols. It has some sensors onboard, including two cameras (one facing horizontally forward and the other facing vertically downwards), a 3 axis gyroscope, a 3 axis accelerometer, a 3 axis magnetometer, a pressure sensor and an ultrasound altimeter. The onboard computer is composed of a 1GHz ARM Cortex-A8 processor running a Linux distribution, 1GB of RAM memory and a Wi-Fi module.

2.1 Kinematic Model

In this work a kinematic model of the Ar.Drone robot is used. Originally such a drone accepts four control signals, associated to the pitch angle θ, the roll angle ϕ and the velocities $\dot{z}$ and $\dot{\psi}$, all of them related to the onboard coordinate system $\langle b\rangle$. However, as it has an internal autopilot that limits the values of the angles θ and ϕ to a quite small values, one can assume that such values are null, although generating velocities in the directions x^b and y^b, respectively. Thus, one can assume that the four commands to be sent to the AR.Drone are effectively velocities commands, namely $\dot{x}^b$, $\dot{y}^b$, $\dot{z}^b$ and $\dot{\phi}^b$, a simplification adopted hereinafter. The result is that the UAV can be considered, in a simplified way, as a horizontal platform able to move along the three axes of the coordinate frame $\langle b\rangle$ and to rotate around the axis z^b. Therefore, the axis z^b is considered parallel to the axis z^w, whereas the axes x^b and y^b are rotated by the angle ψ^b with respect to the axes x^w and y^w, respectively, thus resulting in $\dot{z}^w = \dot{z}^b$ and $\dot{\psi}^w = \dot{\psi}^b$, whereas $\dot{x}^w$ and $\dot{y}^w$ are the projections of $\dot{x}^b$ and $\dot{y}^b$ according to the angle ψ^b.

Taking into account the simplification above mentioned, the velocities

$$\dot{\mathbf{x}}^b = \left[\dot{x}^b,\ \dot{y}^b,\ \dot{z}^b,\ \dot{\psi}^b\right]^T,$$

expressed in the frame $\langle b\rangle$ of the vehicle, becomes the commands for the AR.Drone, which correspond to the three translational speeds along the axes of the frame $\langle w\rangle$ expressed in the body frame $\langle b\rangle$ and the rotational speed around the axis z^b of the frame $\langle b\rangle$, which changes the orientation ψ^b of the UAV.

The velocities $\dot{\mathbf{x}}^b$ are thus transformed into the velocities $\dot{\mathbf{x}}^w$, expressed in the inertial frame $\langle w\rangle$, through a homogeneous transformation matrix

$$\mathbf{F}(\psi^b) = [\mathbf{R}(\mathbf{z}^b)\ \ \mathbf{0};\ \mathbf{0}\ \ \boldsymbol{I}_2],$$

with $\mathbf{R}(\mathbf{z}^b)$ being a 2x2 rotation matrix driven by the angle ψ^b, $\mathbf{0}$ being 2x2 null matrices and $\boldsymbol{I}_2$ being the 2x2 identity matrix. In other words, the kinematic model is written as

$$\dot{\mathbf{x}}^w = \begin{bmatrix}\dot{x}^w\\ \dot{y}^w\\ \dot{z}^w\\ \dot{\psi}^w\end{bmatrix} = \underbrace{\begin{bmatrix}\cos\psi^b & -\sin\psi^b & 0 & 0\\ \sin\psi^b & \cos\psi^b & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}}_{\mathbf{F}(\psi^b)} \underbrace{\begin{bmatrix}\dot{x}^b\\ \dot{y}^b\\ \dot{z}^b\\ \dot{\psi}^b\end{bmatrix}}_{\dot{\mathbf{x}}^b}. \tag{1}$$

2.2 Dynamic Model

To design the control scheme for the UAV, a simplified dynamic model is adopted, which considers the dynamics of the control axes decoupled [11, 16]. This assumption is approximately verified in the robot Ar.Drone and in other micro UAVs, since the onboard autopilot of this type of micro aerial vehicles stabilizes the aircraft during the flight. The model presented in [16] relates the velocities $\dot{\mathbf{x}}^b$ and accelerations $\ddot{\mathbf{x}}^b$ of the UAV, expressed in its frame $\langle b\rangle$, with the normalized control commands $\mathbf{u}^b = [u_{\dot{x}},\ u_{\dot{y}},\ u_{\dot{z}},\ u_{\dot{\psi}}]^T$ (values in the ± 1 range) expressed in frame $\langle b\rangle$ using an approximate two-state linear model. Notice that the commands to be sent to the vehicle are all velocity commands, as explained in the previous subsection. Under such assumptions the dynamic behavior of the Ar.Drone UAV can be modeled as

$$\ddot{\mathbf{x}}^b = \mathbf{K}_u\mathbf{u}^b - \mathbf{K}_v\dot{\mathbf{x}}^b, \tag{2}$$

where $\mathbf{K}_u = diag\big(k_1,\ k_3,\ k_5,\ k_7\big)$ and $\mathbf{K}_v = diag\big(k_2,\ k_4,\ k_6,\ k_8\big)$ are diagonal matrices with the model parameters.

Assuming that the variations of $\dot{\psi}^b$ are very small, i.e., $\dot{\psi}^b \cong 0$, $\dot{\mathbf{F}}\big(\psi^b,\ \dot{\psi}^b\big) \cong \mathbf{0}$, and the model in Eq. 2 can also be written with its states expressed in the frame $\langle w\rangle$, resulting in

$$\ddot{\mathbf{x}}^w = \mathbf{F}(\psi^b)\mathbf{K}_u\mathbf{u}^b - \mathbf{K}_v\dot{\mathbf{x}}^w. \tag{3}$$

After rearranging the terms the model in Eq. 2 is now written as

$$\mathbf{u}^b = \mathbf{A}\ddot{\mathbf{x}}^b + \mathbf{B}\dot{\mathbf{x}}^b, \tag{4}$$

where $\mathbf{A} = \mathbf{K}_u^{-1}$ and $\mathbf{B} = \mathbf{K}_u^{-1}\mathbf{K}_v$.

The dynamic model in Eq. 4 admits a linear parametrization given by

$$\mathbf{u}^b = \mathbf{M}\boldsymbol{\xi}, \tag{5}$$

or

$$\underbrace{\begin{bmatrix} u_{\dot{x}} \\ u_{\dot{y}} \\ u_{\dot{z}} \\ u_{\dot{\psi}} \end{bmatrix}}_{\mathbf{u}^b} = \underbrace{\begin{bmatrix} \ddot{x}^b & 0 & 0 & 0 & \dot{x}^b & 0 & 0 & 0 \\ 0 & \ddot{y}^b & 0 & 0 & 0 & \dot{y}^b & 0 & 0 \\ 0 & 0 & \ddot{z}^b & 0 & 0 & 0 & \dot{z}^b & 0 \\ 0 & 0 & 0 & \ddot{\psi}^b & 0 & 0 & 0 & \dot{\psi}^b \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} 1/k_1 \\ 1/k_3 \\ 1/k_5 \\ 1/k_7 \\ k_2/k_1 \\ k_4/k_3 \\ k_6/k_5 \\ k_8/k_7 \end{bmatrix}}_{\boldsymbol{\xi}},$$

where $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_8]^T = [1/k_1,\ 1/k_3,\ 1/k_5,\ 1/k_7,\ k_2/k_1,\ k_4/k_3,\ k_6/k_5,\ k_8/k_7]^T$ is the vector of dynamic parameters and $\mathbf{M}$ is the regression matrix. Notice that with such a parametrization it is possible to identify the vector $\boldsymbol{\xi}$, as described in [20], thus allowing getting the parameters in the matrices $\mathbf{K}_u$ and $\mathbf{K}_v$ in Eqs. 2, 3 and 4.

3 Control Structure

The control structure developed in this paper is presented in Fig. 2. The models of Eqs. 1 and 2 represent the robot kinematics and the robot dynamics, respectively. Therefore, two control laws are implemented, based on feedback linearization, one for the kinematic model and the other for taking into account the dynamic model of the robot. Finally, a law for adjusting the model parameters is proposed, to take into account non modeled dynamics and parameter uncertainties, aiming at reducing control errors.

3.1 Kinematic Controller

The kinematic controller, based on inverse kinematics, implements the control law

$$\mathbf{u}_k^b = \begin{bmatrix} u_{k\dot{x}} \\ u_{k\dot{y}} \\ u_{k\dot{z}} \\ u_{k\dot{\psi}} \end{bmatrix} = \mathbf{F}(\mathbf{x}^b)^{-1} \left(\dot{\mathbf{x}}_d^w + \boldsymbol{\kappa}_1 \tanh(\boldsymbol{\kappa}_2 \tilde{\mathbf{x}}^w) \right), \tag{6}$$

where $\boldsymbol{\kappa}_1$ and $\boldsymbol{\kappa}_2$ are positive definite diagonal gain matrices and $\tilde{\mathbf{x}}^w(t) = \mathbf{x}_d^w(t) - \mathbf{x}^w(t)$ is the posture error of the aircraft. The function tanh in Eq. 6 is introduced to saturate the control signal for large posture errors, which prevents the saturation of the physical actuators of the vehicle, thus preventing that an unpredicted non-linearity causes system instability. The choice for the function tanh is because such a function provides a smooth transition from the negative saturation to the positive one, allowing dealing with small errors accordingly, which would not be the case using the saturation function sgn, for instance. Effectively, $sgn(value) = \pm 1$ even for small positive/negative $value$, while $\tanh(value) \cong value$ in such a case. For high positive/negative $value$, however, both saturation functions perform similarly. Thus, using the saturation function tanh allows getting small corrections in the control signal for small control errors, what is not the case when using the saturation function sgn.

3.2 Stability Analysis for the Kinematic Controller

This stability analysis is performed using the theory of Lyapunov and assuming a perfect velocity tracking, or $\mathbf{u}_k^b = \dot{\mathbf{x}}^b$ (such an assumption will be relaxed later). This means that the UAV instantaneously responds to any velocity command, what is the case when only the kinematics of the vehicle is considered. In Section 3.4, where the vehicle dynamics is considered when designing the dynamic

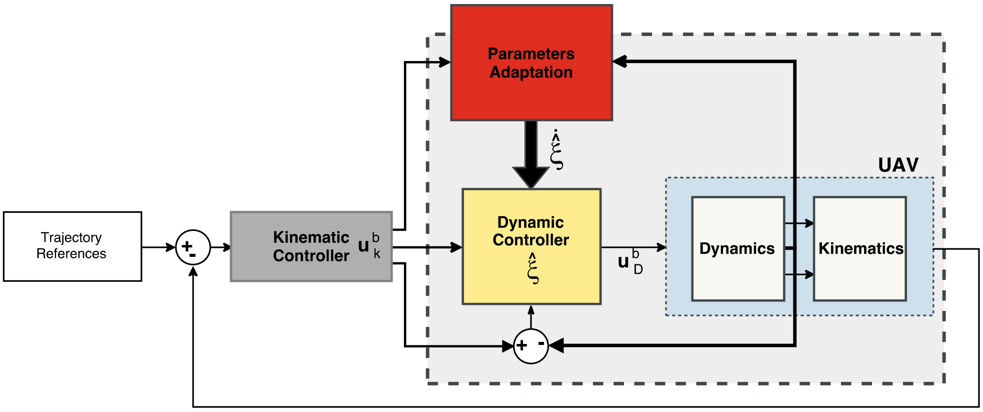


Fig. 2 Control structure

controller, such an assumption will be relaxed. Therefore, considering the assumption of perfect velocity track one can equate (1) and (6), getting

$$\mathbf{F}^{-1}\left[\dot{\mathbf{x}}_d^w + \boldsymbol{\kappa}_1 \tanh(\boldsymbol{\kappa}_2 \tilde{\mathbf{x}}^w)\right] - \mathbf{F}^{-1}\dot{\mathbf{x}}^w = \mathbf{0}, \tag{7}$$

or, after some manipulation,

$$\dot{\tilde{\mathbf{x}}}^w + \boldsymbol{\kappa}_1 \tanh(\boldsymbol{\kappa}_2 \tilde{\mathbf{x}}^w) = \mathbf{0}. \tag{8}$$

Let us consider the Lyapunov candidate function $V = \frac{1}{2}\tilde{\mathbf{x}}^{wT}\tilde{\mathbf{x}}^w > 0$, whose first time derivative is $\dot{V} = -\tilde{\mathbf{x}}^{wT}\boldsymbol{\kappa}_1 \tanh(\boldsymbol{\kappa}_2 \tilde{\mathbf{x}}^w)$. As $\boldsymbol{\kappa}_1$ and $\boldsymbol{\kappa}_2$ are positive definite diagonal matrices, $\dot{V}$ is negative, since the hyperbolic tangent is an odd function. This means that $\tilde{\mathbf{x}}(t) \to \mathbf{0}$ as $t \to \infty$, or, alternatively, the closed-loop control system is asymptotically stable.

3.3 Adaptive Dynamic Controller

To design the adaptive dynamic controller, the simplified dynamic model (5) is expressed as

$$\mathbf{u}^b = \mathbf{D}\ddot{\mathbf{x}}^b + \boldsymbol{\eta}, \tag{9}$$

or

$$\underbrace{\begin{bmatrix} u_{\dot{x}} \\ u_{\dot{y}} \\ u_{\dot{z}} \\ u_{\dot{\psi}} \end{bmatrix}}_{\mathbf{u}^b} = \underbrace{\begin{bmatrix} \xi_1 & 0 & 0 & 0 \\ 0 & \xi_2 & 0 & 0 \\ 0 & 0 & \xi_3 & 0 \\ 0 & 0 & 0 & \xi_4 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} \ddot{x}^b \\ \ddot{x}^b \\ \ddot{z}^w \\ \ddot{\psi}^b \end{bmatrix}}_{\ddot{\mathbf{x}}^b}$$

$$+ \underbrace{\begin{bmatrix} \dot{x}^b & 0 & 0 & 0 \\ 0 & \dot{y}^b & 0 & 0 \\ 0 & 0 & \dot{z}^b & 0 \\ 0 & 0 & 0 & \dot{\psi}^b \end{bmatrix} \begin{bmatrix} \xi_5 \\ \xi_6 \\ \xi_7 \\ \xi_8 \end{bmatrix}}_{\boldsymbol{\eta}}.$$

Then the control law

$$\mathbf{u}_D^b = \mathbf{D}\boldsymbol{\sigma} + \boldsymbol{\eta}, \tag{10}$$

based on the inverse dynamics of the vehicle, is proposed, where

$$\boldsymbol{\sigma} = [\sigma_1\ \sigma_2\ \sigma_3\ \sigma_4]^T = \dot{\mathbf{u}}_k^b + \mathbf{K}_d \dot{\tilde{\mathbf{x}}}^b, \tag{11}$$

with $\mathbf{K}_d = diag[k_{dx}\ k_{dy}\ k_{dz}\ k_{d\psi}] > \mathbf{0}$, $\mathbf{u}_k^b$ being the kinematic control actions obtained from Eq. 6, and $\dot{\tilde{\mathbf{x}}}^b = \mathbf{u}_k^b - \dot{\mathbf{x}}^b$. Then, Eq. 9 can be written as

$$\mathbf{u}_D^b = \mathbf{G}\boldsymbol{\xi}, \tag{12}$$

where

$$\mathbf{G} = \begin{bmatrix} g_{11} & 0 & 0 & 0 & g_{15} & 0 & 0 & 0 \\ 0 & g_{22} & 0 & 0 & 0 & g_{26} & 0 & 0 \\ 0 & 0 & g_{33} & 0 & 0 & 0 & g_{37} & 0 \\ 0 & 0 & 0 & g_{44} & 0 & 0 & 0 & g_{48} \end{bmatrix} \tag{13}$$

and

$$\begin{aligned} g_{11} &= \dot{u}_{k\dot{x}} + k_{dx}[u_{k\dot{x}} - \dot{x}^b], \\ g_{15} &= \dot{x}^b, \\ g_{22} &= \dot{u}_{k\dot{y}} + k_{dy}[u_{k\dot{y}} - \dot{y}^b], \\ g_{26} &= \dot{y}^b, \\ g_{33} &= \dot{u}_{k\dot{z}} + k_{dz}[u_{k\dot{z}} - \dot{z}_b], \\ g_{37} &= \dot{z}^b, \\ g_{44} &= \dot{u}_{k\dot{\psi}} + k_{d\psi}[u_{k\dot{\psi}} - \dot{\psi}^b], \text{ and} \\ g_{48} &= \dot{\psi}^b. \end{aligned} \tag{14}$$

Usually there are uncertainties in the values of the parameters used to perform the calculation of the control actions, as well non modeled dynamic effects. To overcome such problems and get better control performance, a good option is to adapt the values of the parameters used in the controller to reduce the control errors. Thus, considering uncertainties in the parameters of the UAV it results that

$$\mathbf{u}_D^b = \mathbf{G}\hat{\boldsymbol{\xi}} = \mathbf{G}\boldsymbol{\xi} + \mathbf{G}\tilde{\boldsymbol{\xi}} = \mathbf{D}\boldsymbol{\sigma} + \boldsymbol{\eta} + \mathbf{G}\tilde{\boldsymbol{\xi}}, \tag{15}$$

where $\boldsymbol{\xi}$, $\hat{\boldsymbol{\xi}}$ and $\tilde{\boldsymbol{\xi}} = \hat{\boldsymbol{\xi}} - \boldsymbol{\xi}$ are, respectively, the vectors of real parameters (unknown), the estimated ones (identified) and the corresponding error.

3.4 Stability Analysis

This stability analysis is also based on the theory of Lyapunov. First of all, from Eqs. 9 and 15 one gets the closed-loop equation of the control system, given by

$$\mathbf{DR^b} + \eta = \mathbf{D}\boldsymbol{\sigma} + \boldsymbol{\eta} + \mathbf{G}\tilde{\boldsymbol{\xi}}, \tag{16}$$

which can be written as

$$\mathbf{D}(\boldsymbol{\sigma} - \ddot{\mathbf{x}}^b) = -\mathbf{G}\tilde{\boldsymbol{\xi}}. \tag{17}$$

Recalling (11) it comes that

$$\boldsymbol{\sigma} - \ddot{\mathbf{x}}^b = \ddot{\tilde{\mathbf{x}}}^b + \mathbf{K}_d \dot{\tilde{\mathbf{x}}}^b, \tag{18}$$

where $\dot{\tilde{\mathbf{x}}}^b = \mathbf{u}_k^b - \dot{\mathbf{x}}^b$ and $\mathbf{K}_d$ is a positive definite diagonal matrix. Therefore, (17) can be written as

$$\mathbf{D}(\ddot{\tilde{\mathbf{x}}}^b + \mathbf{K}_d \dot{\tilde{\mathbf{x}}}^b) = -\mathbf{G}\tilde{\boldsymbol{\xi}}, \tag{19}$$

from where

$$\ddot{\tilde{\mathbf{x}}}^b = -\mathbf{D}^{-1}\mathbf{G}\tilde{\boldsymbol{\xi}} - \mathbf{K}_d \dot{\tilde{\mathbf{x}}}^b. \tag{20}$$

Now, let

$$V = \frac{1}{2}\dot{\tilde{\mathbf{x}}}^{bT}\mathbf{D}\dot{\tilde{\mathbf{x}}}^b + \frac{1}{2}\tilde{\boldsymbol{\xi}}^T\boldsymbol{\gamma}\tilde{\boldsymbol{\xi}}, \tag{21}$$

be a Lyapunov candidate function, in which $\boldsymbol{\gamma} \in \mathbb{R}^{8\times 8}$ is a positive definite diagonal matrix. By using $\dot{\tilde{\boldsymbol{\xi}}} = \dot{\hat{\boldsymbol{\xi}}}$, because the vector of real parameters is assumed as a constant one, which means that $\dot{\boldsymbol{\xi}} = 0$, the time derivative of the Lyapunov candidate function is

$$\dot{V} = -\dot{\tilde{\mathbf{x}}}^{bT}\mathbf{DK}_d\dot{\tilde{\mathbf{x}}}^{b} - \dot{\tilde{\mathbf{x}}}^{bT}\mathbf{G}\tilde{\boldsymbol{\xi}} + \tilde{\boldsymbol{\xi}}^{T}\boldsymbol{\gamma}\dot{\hat{\boldsymbol{\xi}}}. \tag{22}$$

Considering the parameter adaptation law

$$\dot{\hat{\boldsymbol{\xi}}} = \boldsymbol{\gamma}^{-1}\mathbf{G}^{T}\dot{\tilde{\mathbf{x}}}^{b} \tag{23}$$

for the proposed dynamic controller, and introducing it in Eq. 22 one gets, after some manipulation,

$$\dot{V} = -\dot{\tilde{\mathbf{x}}}^{bT}\mathbf{DK}_d\dot{\tilde{\mathbf{x}}}^{b} \leq 0, \tag{24}$$

which allows concluding that $\dot{\tilde{\mathbf{x}}}^{b}$ and $\tilde{\boldsymbol{\xi}}$ are bounded.

Now, integrating (24) over the time interval $[0, T]$ one gets

$$V(T) - V(0) = -\int_0^T \dot{\tilde{\mathbf{x}}}^{bT}\mathbf{DK}_d\dot{\tilde{\mathbf{x}}}^{b}dt. \tag{25}$$

Since $V(T) > 0$ because $V(t)$ is a Lyapunov candidate function, Eq. 25 becomes

$$V(0) \geq \int_0^T \dot{\tilde{\mathbf{x}}}^{bT}\mathbf{DK}_d\dot{\tilde{\mathbf{x}}}^{b}dt. \tag{26}$$

Because $\mathbf{DK}_d$ is a symmetric positive definite matrix whose minimum and maximum eigenvalues are, respectively, $\lambda_{min}(\mathbf{DK}_d)$ and $\lambda_{max}(\mathbf{DK}_d)$, one can write

$$\lambda_{min}(\mathbf{DK}_d)||\dot{\tilde{\mathbf{x}}}^{b}||^2 \leq \dot{\tilde{\mathbf{x}}}^{bT}\mathbf{DK}_d\dot{\tilde{\mathbf{x}}}^{b} \leq \lambda_{max}(\mathbf{DK}_d)||\dot{\tilde{\mathbf{x}}}^{b}||^2. \tag{27}$$

Finally, from Eqs. 26 and 27 one gets

$$\int_0^T ||\dot{\tilde{\mathbf{x}}}^{bT}||^2 dt \leq \frac{V(0)}{\lambda_{min}(\mathbf{DK}_d)}, \ \forall T. \tag{28}$$

As a result, $\dot{\tilde{\mathbf{x}}}^{b}$ is square-integrable, and then, as $\dot{\tilde{\mathbf{x}}}^{b}$, $\tilde{\boldsymbol{\xi}}$ and $\mathbf{G}$ are bounded, according to the expression $\ddot{\tilde{\mathbf{x}}}^{b} = -\mathbf{D}^{-1}\mathbf{G}\tilde{\boldsymbol{\xi}} - \mathbf{K}_d\dot{\tilde{\mathbf{x}}}^{b}$, it implies that $\ddot{\tilde{\mathbf{x}}}^{b}$ is also bounded. Hence, according to the Barbalat lemma (see [10]), one can conclude that $\dot{\tilde{\mathbf{x}}}(t)^b \rightarrow 0$ with $t \rightarrow \infty$, which means that the velocity control errors converge to zero. Now, revisiting Section 3.2 and relaxing the perfect velocity tracking assumption, that is $\mathbf{u}_k^b - \dot{\mathbf{x}}^b = \boldsymbol{\epsilon} \neq \mathbf{0}$, Eq. 8 can now be written as

$$\dot{\tilde{\mathbf{x}}}^{w} + \boldsymbol{\kappa}_1\tanh(\boldsymbol{\kappa}_2\tilde{\mathbf{x}}^{w}) = \boldsymbol{\delta} \tag{29}$$

with $\boldsymbol{\delta} = \mathbf{F}(\mathbf{x}^b)\boldsymbol{\epsilon}$. Considering the Lyapunov candidate function $P = \frac{1}{2}\tilde{\mathbf{x}}^{wT}\tilde{\mathbf{x}}^{w} > 0$, its first time derivative will be, taking (29) into account,

$$\dot{P} = \tilde{\mathbf{x}}^{wT}\boldsymbol{\delta} - \tilde{\mathbf{x}}^{wT}\boldsymbol{\kappa}_1\tanh(\boldsymbol{\kappa}_2\tilde{\mathbf{x}}^{w}). \tag{30}$$

To guarantee the stability of the closed-loop system, such a derivative should be negative, or $\dot{P} < 0$ for all $\tilde{\mathbf{x}}^{w}$. Such a condition is fulfilled keeping $|\tilde{\mathbf{x}}^{wT}\boldsymbol{\delta}| < |\tilde{\mathbf{x}}^{wT}\boldsymbol{\kappa}_1\tanh(\boldsymbol{\kappa}_2\tilde{\mathbf{x}}^{w})|$. For large values of the posture error $\tilde{\mathbf{x}}^{w}$, $\boldsymbol{\kappa}_1\tanh(\boldsymbol{\kappa}_2\tilde{\mathbf{x}}^{w})$ can be approximated by $\boldsymbol{\kappa}_1\text{sgn}\ \tilde{\mathbf{x}}^{w}$, resulting that $\dot{P}$ will be negative whenever one guarantees that $\lambda_{min}(\boldsymbol{\kappa}_1) > ||\boldsymbol{\delta}||$, what can be obtained by choosing large gains in $\boldsymbol{\kappa}_1$. On the other hand, for small values of $\tilde{\mathbf{x}}^{w}$, $\boldsymbol{\kappa}_1\tanh(\boldsymbol{\kappa}_2\tilde{\mathbf{x}}^{w}) \approx \boldsymbol{\kappa}_1\boldsymbol{\kappa}_2\tilde{\mathbf{x}}^{w}$, and Eq. 29 can be rewritten as $\dot{\tilde{\mathbf{x}}}^{w} + \boldsymbol{\kappa}_1\boldsymbol{\kappa}_2\tilde{\mathbf{x}}^{w} = \boldsymbol{\delta}$. The result is that $\tilde{\mathbf{x}}^{w}$ is limited to $\|\tilde{\mathbf{x}}^{w}\| \leq \dfrac{\|\boldsymbol{\delta}\|}{\lambda_{min}(\boldsymbol{\kappa}_1\boldsymbol{\kappa}_2)}$.

As it was proved from the dynamic compensator, the velocity error $\boldsymbol{\epsilon}(t)$ tends to zero asymptotically, and thus $\boldsymbol{\delta}(t)$ also does that. Therefore, the closed-loop control system based on the proposed adaptive controller is asymptotically stable.

4 Simulation Results

In order to get a first evaluation of the significance of the proposed adaptive dynamic compensator to reduce control errors, the system performance is now evaluated via simulation and the instantaneous absolute value of the tracking error $e(t) = \sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2 + \tilde{\psi}^2}$ is analyzed. It is considered T = 420s of simulation, in which the UAV should follow an 8-shaped trajectory. Four simulations were carried out, each one considering a trajectory set to different values of velocity, through varying ω in

$$\mathbf{x}_d = \begin{bmatrix} \rho sin(wt) + 0.1 \\ 1.25\rho cos(0.5wt) - 0.2 \\ 1.0 + 0.5sin(wt) \\ \frac{\pi}{6}sin(wt) \end{bmatrix}, \tag{31}$$

while keeping $\rho = 1.75$.

The simulations carried out correspond to four cases. In the first one just the kinematic controller was enabled,

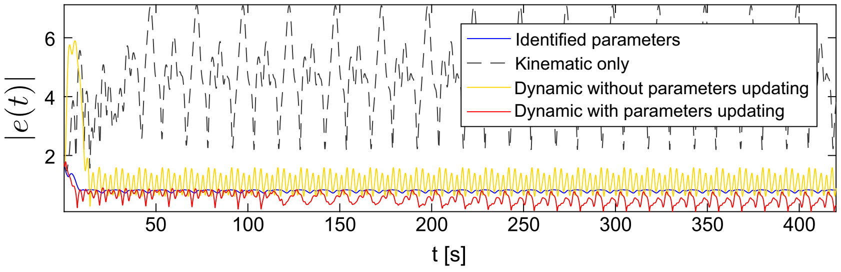


Fig. 3 Evolution of $|e(t)|$ for 420s simulations for the controller using identified parameters, just the kinematic controller, and the dynamic controller with and without parameter updating

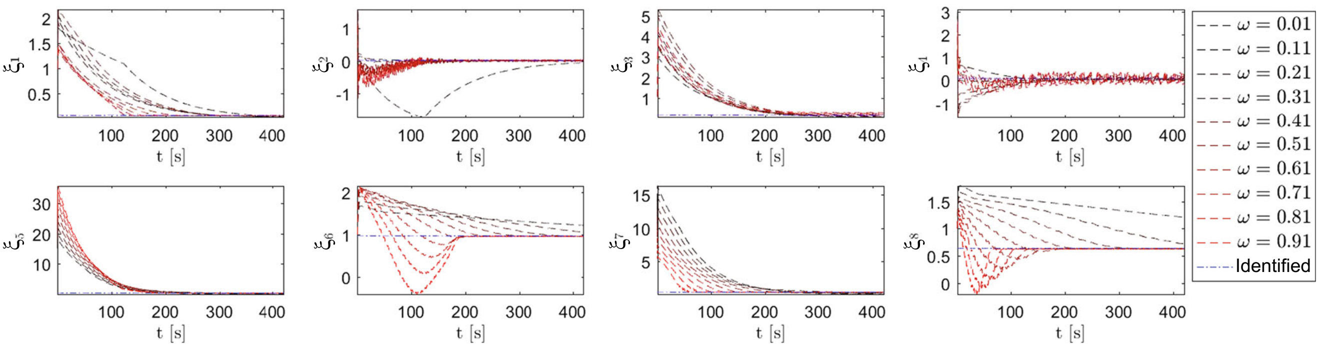


Fig. 4 Evolution of the dynamic model parameters for an eight-shaped trajectory for different values of ω

i.e., the UAV receives the reference commands directly from the kinematic controller of Eq. 6. In the second case the dynamic compensation was activated, but without enabling the parameter updating. The third simulation corresponds to the case in which the dynamic compensator and the parameter updating are both enabled since the beginning of the simulation. Finally, the last simulation included the dynamic compensation activated, but with previously identified parameter and with the parameter updating disabled. Notice that this case corresponds to a case in which the dynamic model was well identified, and does not allow any dynamic change (mass variation during load transportation, for instance).

Figure 3 shows the evolution of the absolute value of the error during the simulation, considering $\omega = 1 rad/s$. It is interesting to note that the errors of the adaptive controller are smaller than those for the model with identified parameters (without parameters updating), which shows the effective contribution of the adaptive controller to improve the performance of the UAV during the tracking of a demanding trajectory. For simulations regarding lower values of ω, however, this is not the case.

Another point deserving to be mentioned, still regarding Figure 3, is the case corresponding to the dynamic compensator with parameters updating. In such a case the parameters were all initialized with the numerical value 1, what means a bad initialization, compared to the identified parameters.

It is also interesting to note that the different simulations are associated to different values of the velocity ω of the trajectory, not to different controller gains. Thus, it is observed that to change the controller gains is a good option, for different trajectories. In this way, even the results with the identified parameters turn out to be worse when the dynamics of the trajectories increases, that is, when ω is increased. Figure 3 illustrates this analysis, showing that the absolute value of the error is lower in the case of the adaptive dynamic compensator than in the case of the dynamic compensator without adaptation, for $\omega = 1 rad/s$. In addition, Fig. 4 shows that the system parameters converge to a fixed value for different trajectories (different values of ω), starting from arbitrary values. The conclusion is that when the dynamic compensator and the parameters updating law are both enabled, the initial values of the model parameters can be arbitrarily chosen (in this work the values adopted where $\xi_1 = \cdots = \xi_8 = 1$). For the simulation using the controller with the identified parameters, the parameters adopted were those identified in [19, 20].

To illustrate a little more the performance of the control structure proposed in this work, the simulated tracking of the eight-shaped trajectory correspondent to $\omega = 1.0 rad/s$, recovered from the data collected along the simulation, is shown in Fig. 5. The two cases corresponding to the model with the identified parameters and the model with unknown parameters initialized with the value 1 and the parameters updating law enabled are illustrated.

Figure 6 shows the evolution of the control errors during the simulation with $\omega = 1.0 rad/s$. It is interesting to note the gradual reduction of the errors caused by the dynamic controller with parameters updating. It is important to observe that there are stationary errors in the model

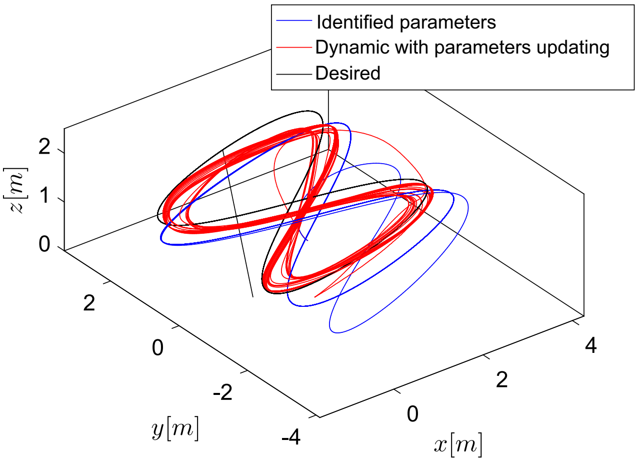
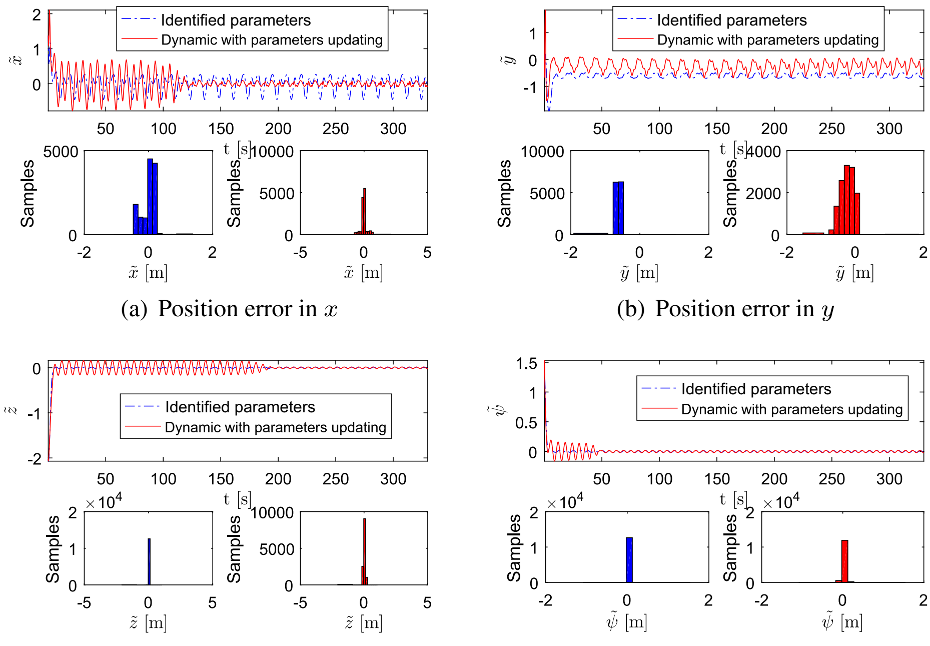


Fig. 5 Simulation with the UAV tracking an Eight-shaped trajectory for which $\omega = 1 rad/s$

Fig. 6 UAV position and orientation errors during the simulation correspondent to $\omega = 1rad/s$



(a) Position error in x

(b) Position error in y

(c) Position error in z

(d) Orientation error

Fig. 7 Position of the UAV during the experiment

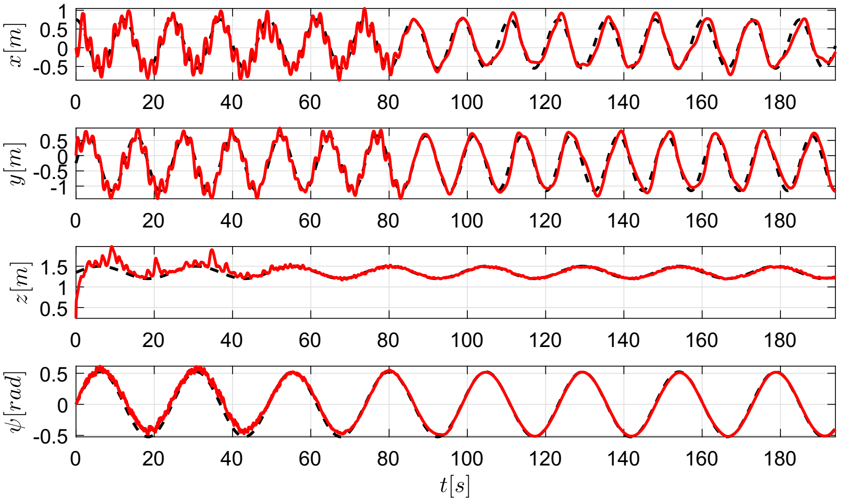
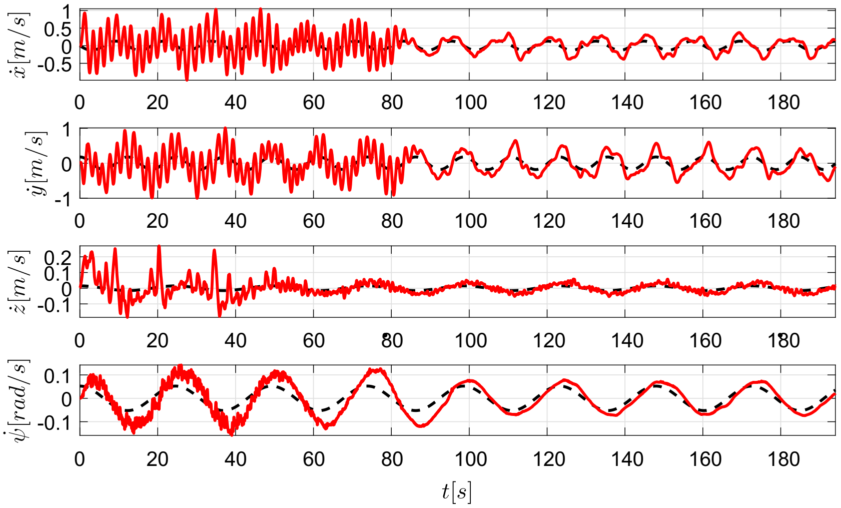


Fig. 8 Velocities of the UAV during the experiment



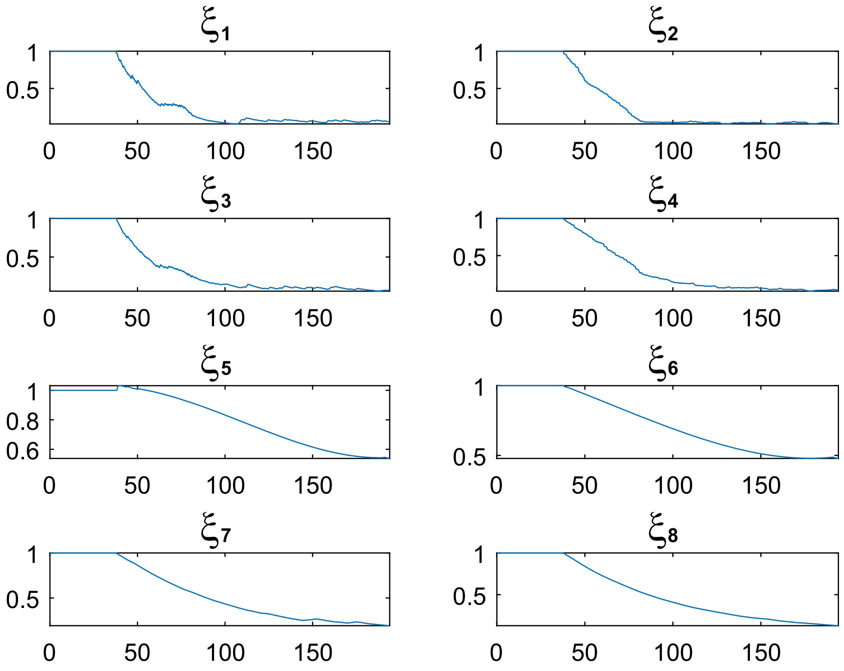


Fig. 9 Evolution of the parameters of the dynamic model along time

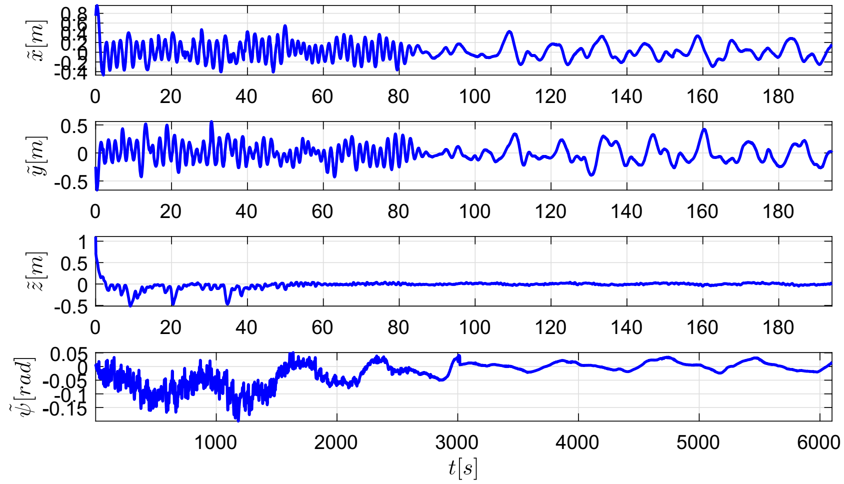


Fig. 10 Position errors with respect to the trajectory

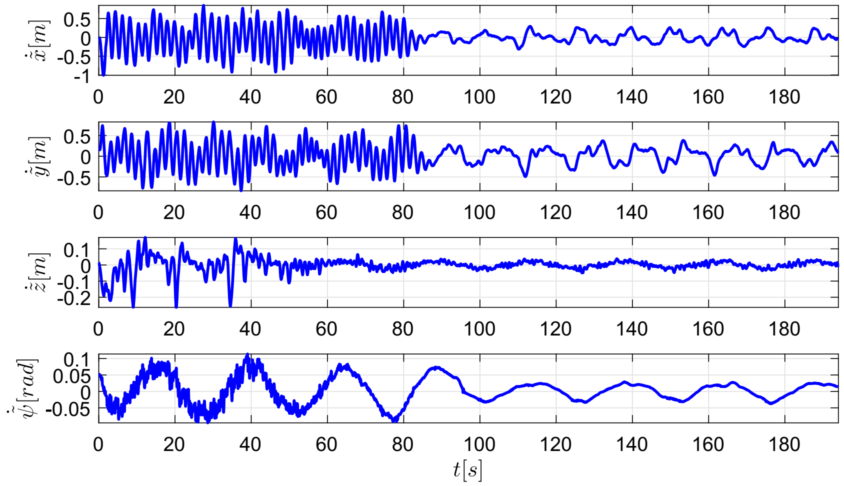


Fig. 11 Velocity errors with respect to the velocity of the trajectory

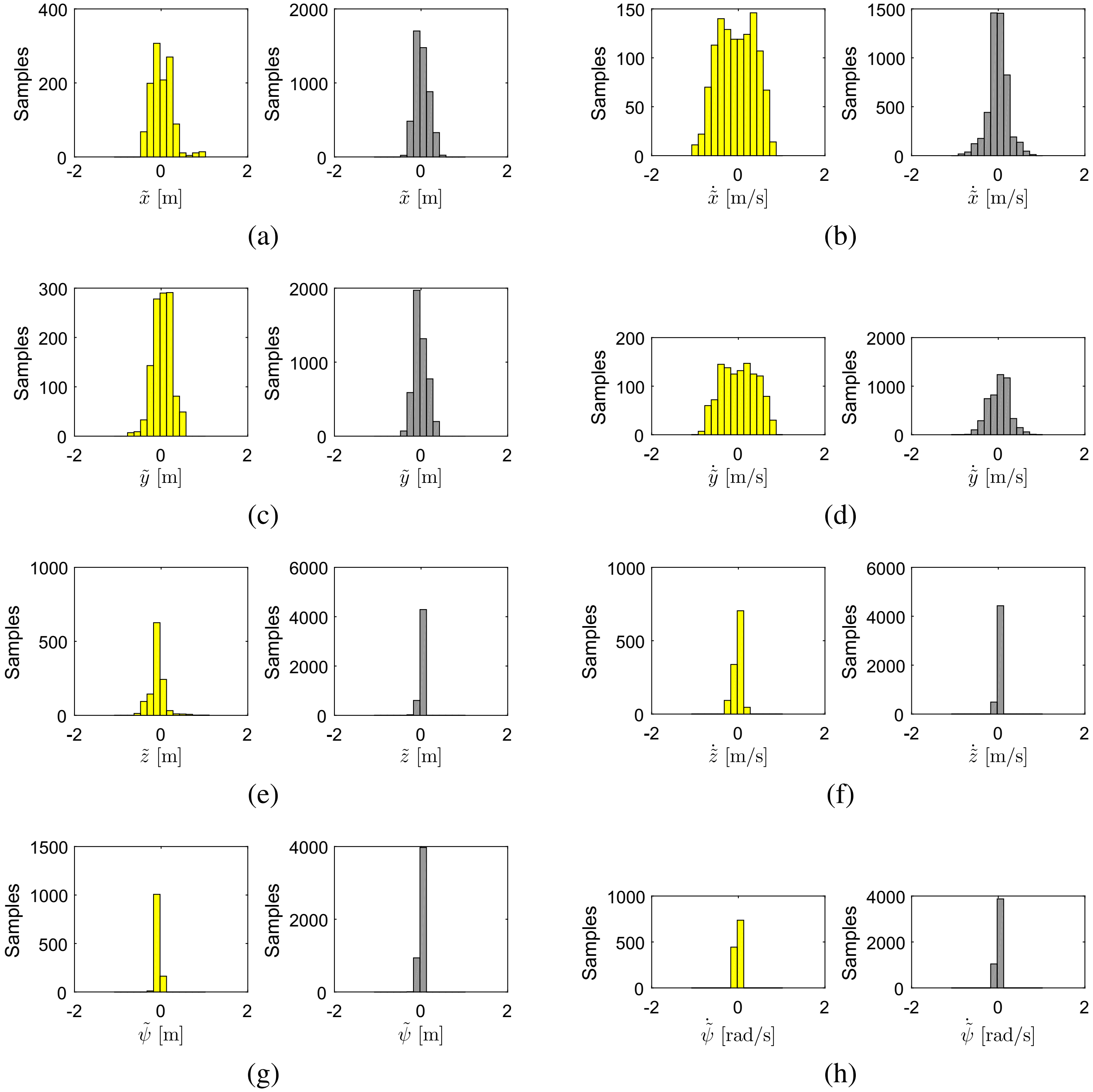


Fig. 12 Distribution of the errors obtained along the experiment. In yellow are the data obtained without activating the parameter updating law ($\dot{\hat{\boldsymbol{\xi}}} = 0$). As for the gray data, they were obtained with the parameter updating law active ($\dot{\hat{\boldsymbol{\xi}}} = \gamma^{-1}\mathbf{G}^T\dot{\tilde{x}}^b$). Parts (**a**), (**c**), (**e**) and (**g**) show the errors in $x, y, z,$ and ψ, whereas parts (**b**), (**d**), (**f**) and (**h**) refer to the velocity errors

with identified parameters (blue line), which are reduced when updating such parameters. Thus, one can conclude that the user could initialize the model parameters with any arbitrary value, not needing to identify the model previously. Moreover, even when the model parameters are available (after identifying the model parameters), the dynamic controller with parameters updating here proposed allows reducing the increased tracking errors when the trajectory is characterized by greater values of ω.

5 Experimental Results

To complete the validation of the controller presented in Section 2, a real experiment consisting in tracking an eight-shaped trajectory was run, considering a real UAV (the AR.Drone 2.0 of Parrot) and $\omega = 1 rad/s$. The experiment starts without a previous knowledge of the parameters of the UAV dynamic model. In fact, the parameters were initialized as $[\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5\ \xi_6\ \xi_7\ \xi_8]^T = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$.

The results hereinafter presented correspond to the data obtained using the adaptive dynamic controller together with the method of sensorial data fusion involving the data coming from the UAV internal sensors and from a depth camera *Xtion Pro Live*, of ASUS. The data fusion method, the data capture system and the technique adopted to detect the UAV in the environment from the depth images are those previously presented in [17–19].

Figures 7 and 8 illustrate the positions and velocities of the UAV during the trajectory tracking. There one can see that both the position and velocity errors are closer to zero when the parameter updating is on. Actually, in Fig. 8 one can see that the desired velocities are reached ($\dot{x} \rightarrow \dot{x}_d, \dot{y} \rightarrow \dot{y}_d, \dot{z} \rightarrow \dot{z}_d$), whereas in Fig. 7 one can check that the position errors get close to zero, thus meaning that $(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{\psi}) \rightarrow 0$ and $(\dot{\tilde{x}}, \dot{\tilde{y}}, \dot{\tilde{z}}, \dot{\tilde{\psi}}) \rightarrow 0$, approximately.

It is important to emphasize that the initial values of the dynamic parameters are all set to 1. The controller used is the adaptive dynamic one, but in the very beginning the parameter updating law of Eq. 23 is not active (this means that $\dot{\hat{\boldsymbol{\xi}}} = 0$). Only after 37.64 seconds of flight such an updating law is activated. The evolution of the parameters along time is shown in Fig. 9, where one can see that each parameter is initialized with the value 1 and only after 37.64 seconds they start varying. As a consequence, one can see that the tracking errors are reduced when the parameters are updated.

To make easier to visualize the decrease in the control errors, Figs. 10 and 11 present the evolution of the position and velocity errors, respectively, along time.

As one can see from the results presented, the adaptive control system here proposed is able to decrease the errors in trajectory tracking caused by a bad identification of the model parameters or a non modeled effect. The plots of error distribution presented in Fig. 12 effectively show that the variance and the dispersion of the errors decrease, getting close to 0, when the parameter updating law is active.

Finally, in the url https://youtu.be/HEw0JTJor80 the interested reader can find a video of the experiment presented in this section.

To conclude this section, notice that the adaptation law here proposed was validated under very realistic considerations.

6 Conclusion

In this work an adaptive dynamic controller was designed for a UAV in order to compensate for parametrization errors or non modeled dynamic effects. Although the model used to represent the UAV is simpler than others presented in the literature, the results here presented allows to claim that the proposed adaptive controller is able to deal with uncertainties in the model parameters or non modeled effects. Moreover, the model parameters have been initialized with a default initial value and the adaptive control system, in addition to reducing control errors, has obtained a good estimate of the model parameters (which can be an initial guess for other experiments, for instance). As a result of the theoretical analysis, simulations and experiments presented in the paper one can conclude that the strategy of adapting the controller improves the performance of the UAV during the task accomplishment, mainly in situations in which the UAV dynamics is strongly excited. Moreover, using the parameters updating law it is not necessary a precise model identification before using the UAV in trajectory tracking tasks. In the simulations and experiments here presented, for instance, the initial values of the parameters were set to 1, without any identification. Unlike other works that execute several maneuvers and movements to carry out the identification of the model parameters before starting the task of interest, this work proposes a system that simultaneously performs navigation and parameter identification. To synthesize the conclusions, the results here presented allow claiming that the proposed algorithm works reducing control errors and improving navigation performance.

Acknowledgments The authors thank CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, an agency of the Brazilian Ministry of Science and Technology to support scientific and technological development –, FAPES – Fundação de Amparo à Pesquisa e Inovação do Espírito Santo, the agency of the State of Espírito Santo that supports scientific and technological development –, for the financial support to this work. They also thank CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, an agency of the Brazilian Ministry of Education to support high education –, for the scholarship granted to Mr. Santos, the Federal Institute of Espírito Santo, the Federal University of Espírito Santo and the Institute of Automatics of the National University of San Juan, Argentine, and CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas), Argentina for supporting the development of this research. A short version of this paper was presented in ICUAS 2017.

References

1. Achtelik, M., Bierling, T., Wang, J., Höcht, L., Holzapfel, F.: Adaptive Control of a Quadcopter in the Presence of Large/Complete Parameter Uncertainties. In: Infotech@Aerospace 2011. American Institute of Aeronautics and Astronautics, St. Louis, MI, USA (2011). https://doi.org/10.2514/6.2011-1485
2. Alvarenga, J., Vitzilaios, N.I., Valavanis, K.P., Rutherford, M.J.: Survey of unmanned helicopter model-based navigation and control techniques. J. Intell. Robot. Syst. **80**(1), 87–138 (2015)
3. Bouadi, H., Cunha, S.S., Drouin, A., Mora-Camino, F.: Adaptive Sliding Mode Control for Quadrotor Attitude Stabilization and Altitude Tracking. In: 2011 IEEE 12Th International Symposium on Computational Intelligence and Informatics (CINTI), pp. 449–455 (2011). https://doi.org/10.1109/CINTI.2011.6108547
4. Efe, M.O.: Robust Low Altitude Behavior Control of a Quadrotor Rotorcraft through Sliding Modes. In: 2007 Mediterranean

Conference on Control Automation, pp. 1-6, Athens, Greece (2007). https://doi.org/10.1109/MED.2007.4433755

5. Emran, B., Yesildirek, A.: Robust nonlinear composite adaptive control of quadrotor. Int. J. Digital Inform. Wirel. Commun. **4**, 213–225 (2014)
6. Felix, M.C.: A Two Level Non Linear Inverse Control Structure for Rotorcraft Trajectory Tracking. In: 2007 Chinese Control Conference, pp. 321–325 (2007). https://doi.org/10.1109/CHICC.2006.4347095
7. Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.M., Hirzinger, G., Rus, D.: Energy-Efficient Autonomous Four-Rotor Flying Robot Controlled at 1 Khz. In: Proceedings 2007 IEEE International Conference on Robotics and Automation, pp. 361–366 (2007). https://doi.org/10.1109/ROBOT.2007.363813
8. Hatamleh, K.S., Ma, O., Paz, R.: A uav model parameter identification method: a simulation study. Int. J. Inform. Acquis. **06**(04), 225–238 (2009)
9. Hoffer, N.V., Coopmans, C., Jensen, A.M., Chen, Y.: A survey and categorization of small low-cost unmanned aerial vehicle system identification. J. Intell. Robot. Syst. **74**(1-2), 129–145 (2014)
10. Khalil, H.K.: Nonlinear Control. Pearson, London (2015)
11. Krajník, T., Vonásek, V., Fiser, D., Faigl, J.: Ar-drone as a robotic platform for research and education. In: International Conference on Research and Education in Robotics - EUROBOT 2011. Prague, Czech Republic (2011). https://doi.org/10.1007/978-3-642-21975-7_16
12. Liu, H., Xi, J., Zhong, Y.: Robust attitude stabilization for nonlinear quadrotor systems with uncertainties and delays. IEEE Trans. Ind. Electron. **64**(7), 5585–5594 (2017)
13. Madani, T., Benallegue, A.: Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique. In: Proceedings of the 45Th IEEE Conference on Decision and Control, pp. 1515–1520 (2006). https://doi.org/10.1109/CDC.2006.377548
14. Martins, F.N., Celeste, W.C., Carelli, R., Sarcinelli-Filho, M., Bastos-Filho, T.F.: An adaptive dynamic controller for autonomous mobile robot trajectory tracking. Control. Eng. Pract. **16**(11), 1354–1363 (2008)
15. Mohammadi, M., Shahri, A.M.: Adaptive nonlinear stabilization control for a quadrotor uav: theory, simulation and experimentation. J. Intell. Robot. Syst. **72**(1), 105–122 (2013)
16. Santana, L.V., Brandão, A.S., Sarcinelli-Filho, M.: Navigation and cooperative control using the ar.drone quadrotor. J. Intell. Robot. Syst. **84**(1–4), 327–350 (2016). https://doi.org/10.1007/s10846-016-0355-y
17. Santos, M., Santana, L., Brandao, A., Sarcinelli-Filho, M.: Uav Obstacle Avoidance Using Rgb-D System. In: 2015 International Conference On Unmanned Aircraft Systems (ICUAS), pp. 312–319 (2015). https://doi.org/10.1109/ICUAS.2015.7152305
18. Santos, M., Santana, L., Martins, M., Brandao, A., Sarcinelli-Filho, M.: Estimating and Controlling Uav Position Using Rgb-D/Imu Data Fusion with Decentralized Information/Kalman Filter. In: 2015 IEEE International Conference on Industrial Technology (ICIT), pp. 232–239 (2015). https://doi.org/10.1109/ICIT.2015.7125104
19. Santos, M.C., Santana, L.V., Brandão, A.S., Sarcinelli-Filho, M., Carelli, R.: Indoor low-cost localization system for controlling aerial robots. Control. Eng. Pract. **61**, 93–111 (2017)
20. Santos, M.C.P., Rosales, C.D., Sarcinelli-Filho, M., Carelli, R.: A novel null-space-based uav trajectory tracking controller with collision avoidance. IEEE/ASME Trans. Mechatron. **22**(6), 2543–2553 (2017)
21. Santoso, F., Garratt, M.A., Anavatti, S.G.: Adaptive Neuro-Fuzzy Inference System Identification for the Dynamics of the Ar.Drone Quadcopter. In: 2016 International Conference on Sustainable Energy Engineering and Application (ICSEEA), pp. 55–60 (2016). https://doi.org/10.1109/ICSEEA.2016.7873567
22. Sastry, S., Bodson, M.: Adaptive Control: Stability, Convergence and Robustness. Courier Corporation, North Chelmsford (2011)
23. Zha, C., Ding, X., Yu, Y., Wang, X.: Quaternion-based nonlinear trajectory tracking control of a quadrotor unmanned aerial vehicle. Chinese J. Mech. Eng. **30**(1), 77–92 (2017)

Milton Cesar Paes Santos received the B.S. degree in Computer Engineering from Federal University of Espírito Santo, Brazil, in 2011, and the M. Sc. degree in Electrical Engineering, also from Federal University of Espírito Santo, Brazil, in 2013. He is currently working towards the Ph.D. degree at the Department of Electrical Engineering, Federal University of Espírito Santo. Since 2013, he is with the Federal Institute of Espírito Santo, Santa Teresa, as a D-301 associate professor. His research interest is computer vision applied to mobile robotics and the design of computational systems for control applications with aerial robots.

Claudio Darío Rosales studied Electronic Engineering at UNSJ (Argentine) obtaining the degree Electronic Engineering in 2009, and the PhD in Control Systems from the National University of San Juan, in 2014. He is a postdoctoral researcher of the Council for Scientific and Technological Research, Argentina, since 2015. His main interests included algorithms for multi-robots systems, nonlinear control and aerial robotic.

Jorge Antonio Sarapura was born in Orán, Salta, Argentina. He graduated with honours in Electronic Engineering from the National University of Tucumán, Argentina, and obtained a Master's degree in Control Systems Engineering from the National University of San Juan (UNSJ). He is a teaching assistant at the National University of San Juan and a Ph.D. student. His research interests are in robotics, artificial vision applied to automatic control and system identification.

Mário Sarcinelli-Filho received the B.S. degree in Electrical Engineering from Federal University of Espírito Santo, Brazil, in 1979, and the M. Sc. and Ph. D. degrees, both in Electrical Engineering, from Federal University of Rio de Janeiro, Brazil, in 1983 and 1990, respectively. He is currently a full professor at the Department of Electrical Engineering, Federal University of Espírito Santo, Brazil, and a researcher of the Brazilian National Council for Scientific and Technological Development (CNPq). His research interests are control of unmanned aerial vehicles, including formations of unmanned aerial vehicles, coordinated control of mobile robots, mobile robot navigation, signal and image processing, and sensing systems. He has coauthored more than 40 journal papers, more than 300 conference papers, and 14 book chapters, and advised 17 PhD students and 19 MSc students.

Ricardo Carelli (M'76 - SM'98) was born in San Juan, Argentina. He graduated in Engineering from the National University of San Juan, Argentina, and obtained a Ph.D degree in Electrical Engineering from the National University of Mexico (UNAM). He is full professor at the National University of San Juan and senior researcher of the National Council for Scientific and Technical Research (CONICET, Argentina). Prof. Carelli is Director of the Instituto de Automática, National University of San Juan (Argentina). His research interests are on Robotics, Manufacturing Systems, Adaptive Control and Artificial Intelligence Applied to Automatic Control. Prof. Carelli is a senior member of IEEE and a member of AADECA-IFAC.