

First Flight Tests for a Quadrotor UAV with Tilting Propellers

Markus Ryll, Heinrich H Bülthoff, Paolo Robuffo Giordano

▶ **To cite this version:**

Markus Ryll, Heinrich H Bülthoff, Paolo Robuffo Giordano. First Flight Tests for a Quadrotor UAV with Tilting Propellers. IEEE Int. Conf. on Robotics and Automation, ICRA 2013, May 2013, Karlsruhe, Germany. pp.295-302, 10.1109/ICRA.2013.6630591 . hal-00910823

HAL Id: hal-00910823

https://inria.hal.science/hal-00910823

Submitted on 28 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

First Flight Tests for a Quadrotor UAV with Tilting Propellers

Markus Ryll, Heinrich H. Bülthoff, and Paolo Robuffo Giordano

***Abstract*— In this work we present a novel concept of a quadrotor UAV with tilting propellers. Standard quadrotors are limited in their mobility because of their intrinsic underactuation (only 4 independent control inputs vs. their 6-dof pose in space). The quadrotor prototype discussed in this paper, on the other hand, has the ability to also control the orientation of its 4 propellers, thus making it possible to overcome the aforementioned underactuation and behave as a fully-actuated flying vehicle. We first illustrate the hardware/software specifications of our recently developed prototype, and then report the experimental results of some preliminary, but promising, flight tests which show the capabilities of this new UAV concept.**

I. INTRODUCTION

Common UAVs (Unmanned Aerial Vehicles) are underactuated mechanical systems, i.e., possessing less control inputs than available degrees of freedom (dofs). This is, for instance, the case of helicopters and quadrotor UAVs [1], [2]. For these latter platforms, only the Cartesian position and yaw angle of their body frame w.r.t. an inertial frame can be independently controlled (4 dofs), while the behavior of the remaining roll and pitch angles (2 dofs) is completely determined by the trajectory chosen for the former 4 dofs. Presence of such an underactuation does not only limit the flying ability of quadrotors in free or cluttered space, but it also degrades the possibility of interacting with the environment by exerting desired forces in arbitrary directions. As quadrotor UAVs are being more and more exploited as autonomous flying *service robots* [3], [4], it is important to explore different actuation strategies that can overcome the aforementioned underactuation problem and allow for full motion/force control in all directions in space.

Motivated by these considerations, several possibilities have been proposed in the past literature spanning different concepts: ducted-fan designs [5], tilt-wing mechanisms [6], [7], or tilt-rotor actuations [8], [9]. Along similar lines, previous works [10], [11] also discussed a novel concept for a quadrotor UAV with *actuated* tilting propellers, i.e., with propellers able to rotate around the axes connecting them to the main body frame. This design grants a total of 8 control inputs ($4 + 4$ propeller spinning/tilting velocities), and, as formally shown in [10], makes it possible to obtain complete controllability over the main body 6-dof configuration in $\mathbb{R}^3 \times SO(3)$, thus rendering the quadrotor UAV a fully-actuated flying vehicle.

M. Ryll and P. Robuffo Giordano are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany {markus.ryll,prg}@tuebingen.mpg.de.

H. H. Bülthoff is with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany, and with the Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul, 136-713 Korea hhb@tuebingen.mpg.de.

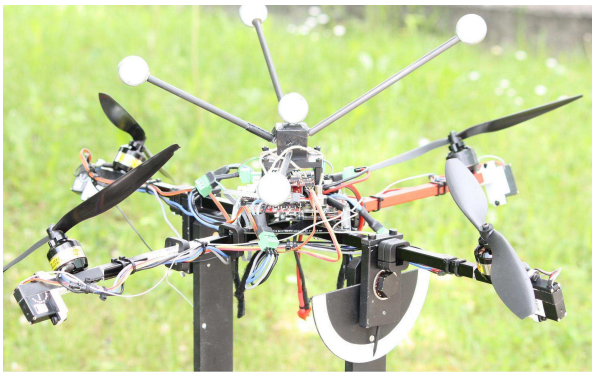


Fig. 1: A picture of the prototype on a testing gimbal

The work in [10] proposed a trajectory tracking controller based on dynamic feedback linearization and meant to fully exploit the actuation capabilities of this new design. The closed-loop tracking performance was, however, only evaluated via numerical simulations, albeit considering a realistic dynamical model. Goal of the present paper is to extend [10] by illustrating the control implementation and trajectory tracking performance of a *real prototype* developed in our group, in particular by reporting the results of several experiments in different flight regimes.

The rest of the paper is organized as follows: Sect. II reviews the modeling assumptions and control design proposed in [10] and upon which this work is based. Section III describes our prototype from the hardware and software points of view and discusses the main 'real-world' discrepancies w.r.t. the modeling assumptions taken in [10]. Finally, Sect. IV presents some experimental results for hovering and trajectory tracking regimes, and Sect. V concludes the paper.

II. REVIEW OF THE DYNAMIC MODELING AND CONTROL DESIGN

For the reader's convenience, in this Section we will briefly summarize the modeling assumptions and control approach proposed in [10] for a quadrotor with tilting propellers (from now on denoted as "*omnicopter*"). Figure 1 shows a picture of the prototype, while Figs. 3–2 and Fig. 5 present CAD and schematic views.

A. Dynamic model

The omnicopter consists of 5 rigid bodies in relative motion among themselves: the main body B and the 4 propeller groups P_i. The propeller groups P_i host the propeller and its associated (spinning) motor as well as the additional motor responsible for the tilting actuation mechanism, see Fig. 3. Let $\mathcal{F}_W : \{\boldsymbol{O}_W;\ \boldsymbol{X}_W,\ \boldsymbol{Y}_W,\ \boldsymbol{Z}_W\}$ be a world

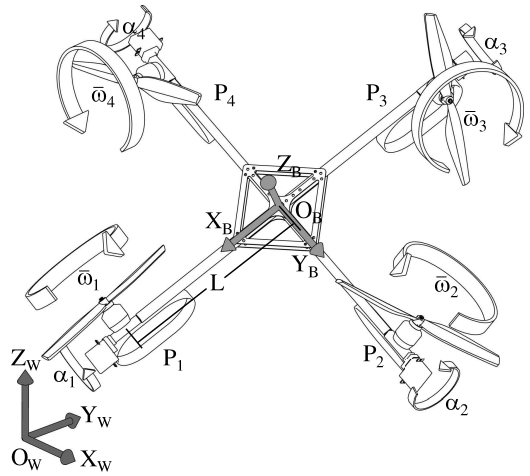


Fig. 2: Schematic view of the omnicopter. The center of mass is assumed to be located at the origin of the body frame. The symbol L represents the length of all propeller arms, $\bar{w}_i$, $i = 1 \dots 4$, the propeller rotation speeds and α_i, $i = 1 \dots 4$, the orientation angles of the propeller groups P_i

inertial frame, $\mathcal{F}_B : \{\boldsymbol{O}_B;\ \boldsymbol{X}_B,\ \boldsymbol{Y}_B,\ \boldsymbol{Z}_B\}$ a moving frame attached to the quadrotor body at its center of mass, and $\mathcal{F}_{P_i} : \{\boldsymbol{O}_{P_i};\ \boldsymbol{X}_{P_i},\ \boldsymbol{Y}_{P_i},\ \boldsymbol{Z}_{P_i}\}$, $i = 1 \dots 4$, the frames associated to the i-th propeller groups, with $\boldsymbol{X}_{P_i}$ representing the tilting actuation axis and $\boldsymbol{Z}_{P_i}$ the propeller actuated spinning (thrust T_i) axis. We also let ${}^W\boldsymbol{R}_B \in SO(3)$ represent the orientation of the body frame w.r.t. the world frame, and ${}^B\boldsymbol{R}_{P_i}(\alpha_i) \in SO(3)$ the orientation of the propeller group P_i w.r.t. the body frame, with $\alpha_i \in \mathbb{R}$ denoting the i-th actuated tilting angle. The omnicopter configuration is then completely determined by the body position $\boldsymbol{p} = {}^W\boldsymbol{O}_B \in \mathbb{R}^3$ and orientation ${}^W\boldsymbol{R}_B$ in the world frame, and by the 4 tilting angles α_i specifying the propeller group orientations w.r.t. the body frame (rotations about $\boldsymbol{X}_{P_i}$).

By employing standard techniques, such as the Newton-Euler procedure [12], it is possible to derive a complete dynamical model of the omnicopter by considering the forces and moments generated by the propeller spinning motion, as well as gyroscopic and inertial effects due to the relative motion of the 5 bodies among themselves. To this end, we let $\boldsymbol{\omega}_B \in \mathbb{R}^3$ be the angular velocity of the quadrotor body B w.r.t. the world frame and expressed in the body frame[1] and $\boldsymbol{\omega}_{P_i}$ be the angular velocity of the propeller group P_i w.r.t. the world frame, i.e.,

$$\boldsymbol{\omega}_{P_i} = {}^B\boldsymbol{R}_{P_i}^T\boldsymbol{\omega}_B + [\dot{\alpha}_i\ 0\ \bar{w}_i]^T,$$

where $\bar{w}_i \in \mathbb{R}$ is the spinning velocity of the propeller about $\boldsymbol{Z}_{P_i}$ and $\dot{\alpha}_i \in \mathbb{R}$ the tilting velocity of the group about $\boldsymbol{X}_{P_i}$, both w.r.t. the body frame. By applying the Euler equations of motion, one has

$$\boldsymbol{\tau}_{P_i} = \boldsymbol{I}_{P_i}\dot{\boldsymbol{\omega}}_{P_i} + \boldsymbol{\omega}_{P_i} \times \boldsymbol{I}_{P_i}\boldsymbol{\omega}_{P_i} - \boldsymbol{\tau}_{ext_i} \tag{1}$$

[1]In the following, we will assume that every quantity is expressed in its own frame, e.g., $\boldsymbol{\omega}_B = {}^B\boldsymbol{\omega}_B$.

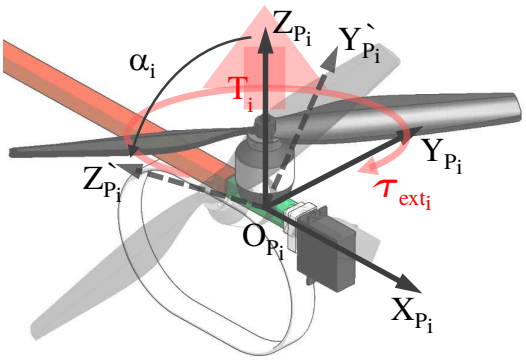


Fig. 3: Close view of the i-th tilting arm where the frame $\mathcal{F}_{P_i}$, the associated propeller thrust T_i, torque τ_{ext_i}, and propeller tilt angle α_i are shown

where $\boldsymbol{I}_{P_i} \in \mathbb{R}^{3\times3}$ is the (constant) symmetric and positive definite Inertia matrix of the propeller group, and $\boldsymbol{\tau}_{ext_i}$ any external torque applied to the propeller. As usually done, we assume that the propeller spinning about $\boldsymbol{Z}_{P_i}$ causes a counter-torque due to air drag and we neglect any additional aerodynamic effect. Therefore, $\boldsymbol{\tau}_{ext_i} = [0\ 0\ -\tau_{D_i}]^T$ with $\tau_{D_i} = k_m\bar{w}_i|\bar{w}_i|$, $k_m > 0$.

As for the dynamics of the main body B, one obtains

$$\boldsymbol{\tau}_B = \boldsymbol{I}_B\dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times \boldsymbol{I}_B\boldsymbol{\omega}_B + \sum_{i=1}^{4} {}^B\boldsymbol{R}_{P_i}\boldsymbol{\tau}_{P_i}, \tag{2}$$

with $\boldsymbol{I}_B \in \mathbb{R}^{3\times3}$ being the (constant) symmetric and positive definite Inertia matrix of B. Here, the torques $\boldsymbol{\tau}_B \in \mathbb{R}^3$ exerted on B are generated by the moments of the four propeller forces $\boldsymbol{T}_{P_i} \in \mathbb{R}^3$ acting at $\boldsymbol{O}_{P_i}$, that is,

$$\boldsymbol{\tau}_B = \sum_{i=1}^{4} ({}^B\boldsymbol{O}_{P_i} \times {}^B\boldsymbol{R}_{P_i}\boldsymbol{T}_{P_i}). \tag{3}$$

Again, we model the propeller forces as acting along the $\boldsymbol{Z}_{P_i}$ axes so that $\boldsymbol{T}_{P_i} = [0\ 0\ T_i]^T$ with $T_i = k_f\bar{w}_i|\bar{w}_i|$, $k_f > 0$, and neglect additional aerodynamic effects.

Finally, concerning the translational dynamics, assuming that the barycenter of each propeller group P_i coincides with $\boldsymbol{O}_{P_i}$ (which holds with a good approximation for the prototype described in Sect. III-A), we have

$$m\ddot{\boldsymbol{p}} = m\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + {}^W\boldsymbol{R}_B \sum_{i=1}^{4} {}^B\boldsymbol{R}_{P_i}\boldsymbol{T}_{P_i} \tag{4}$$

where m is the total mass of the quadrotor and propeller bodies and g the scalar gravity constant.

To conclude, we note that this model has 8 independent inputs: the 4 motor torques actuating the tilting axes $\boldsymbol{X}_{P_i}$, i.e., $\tau_{\alpha_i} = \boldsymbol{\tau}_{P_i}^T\boldsymbol{X}_{P_i} \in \mathbb{R}$, and the 4 motor torques actuating the spinning axes $\boldsymbol{Z}_{P_i}$, i.e., $\tau_{\bar{w}_i} = \boldsymbol{\tau}_{P_i}^T\boldsymbol{Z}_{P_i} \in \mathbb{R}$, with $i = 1 \dots 4$.

Symbols	Definitions
B	omnicopter body
P_i	propeller group
$\mathcal{F}_W$	inertial world frame
$\mathcal{F}_B$	body frame B
$\mathcal{F}_{P_i}$	i-th propeller group frame
$\boldsymbol{p}$	position of B in $\mathcal{F}_W$
${}^W\boldsymbol{R}_B$	rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$
${}^B\boldsymbol{R}_{P_i}$	rotation matrix from $\mathcal{F}_{P_i}$ to $\mathcal{F}_B$
α_i	i-th propeller tilting angle about $\boldsymbol{X}_{P_i}$
$\bar{w}_i$	i-th propeller spinning velocity about $\boldsymbol{Z}_{P_i}$
$\boldsymbol{\omega}_B$	angular velocity of B in $\mathcal{F}_B$
$\boldsymbol{\tau}_{ext_i}$	i-th propeller air drag torque about $\boldsymbol{Z}_{P_i}$
$\boldsymbol{T}_i$	i-th propeller thrust along $\boldsymbol{Z}_{P_i}$
τ_{P_i}	motor torque actuating $\boldsymbol{X}_{P_i}$
$\tau_{\bar{w}_i}$	motor torque actuating $\boldsymbol{Z}_{P_i}$
m	total omnicopter mass
$\boldsymbol{I}_{P_i}$	inertia of the i-th propeller group P_i
$\boldsymbol{I}_B$	inertia of the omnicopter body B
k_f	propeller thrust coefficient
k_m	propeller drag coefficient
L	distance of $\mathcal{F}_{P_i}$ from $\mathcal{F}_B$
g	gravity constant

B. Trajectory Tracking Control

Owing to its actuation system, the omnicopter can exactly track a desired and arbitrary trajectory $(\boldsymbol{p}_d(t),\, \boldsymbol{R}_d(t)) \in \mathbb{R}^3 \times SO(3)$ for the body position $\boldsymbol{p}$ and orientation ${}^W\boldsymbol{R}_B$ taken as output functions, see [10]. We then review here the proposed tracking control scheme. First, we simplify the previous dynamical model by assuming that the motors actuating the spinning and tilting axes can realize given desired speeds $\bar{w}_i$ and $w_{\alpha_i} = \dot{\alpha}_i$ with negligible transients thanks to high-gain low-level loops. This way, w_{α_i} and $\bar{w}_i$ can be considered as 'velocity' inputs in place of the (actual) motor torques. Second, we neglect the gyroscopic effects due to the relative motion among the omnicopter parts, and treat them as disturbances to be rejected by the trajectory controller.

We start defining $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_4]^T \in \mathbb{R}^4$, $\boldsymbol{w}_\alpha = [w_{\alpha_1} \dots w_{\alpha_4}]^T \in \mathbb{R}^4$ and $\boldsymbol{w} = [\bar{w}_1|\bar{w}_1| \dots \bar{w}_4|\bar{w}_4|]^T \in \mathbb{R}^4$, where the quantity $w_i = \bar{w}_i|\bar{w}_i|$ represents the *signed square* of the i-th spinning velocity. The omnicopter dynamical model then reduces to

$$\left\{\begin{array}{rcl} \ddot{\boldsymbol{p}} &=& \left[\begin{array}{c} 0 \\ 0 \\ -g \end{array}\right] + \frac{1}{m}{}^W\boldsymbol{R}_B\boldsymbol{F}(\boldsymbol{\alpha})\boldsymbol{w} \\ \dot{\boldsymbol{\omega}}_B &=& \boldsymbol{I}_B^{-1}\boldsymbol{\tau}(\boldsymbol{\alpha})\boldsymbol{w} \\ \dot{\boldsymbol{\alpha}} &=& \boldsymbol{w}_\alpha \\ {}^W\dot{\boldsymbol{R}}_B &=& {}^W\boldsymbol{R}_B[\boldsymbol{\omega}_B]_\wedge \end{array}\right. \tag{5}$$

with $[\cdot]_\wedge$ being the usual map from $\mathbb{R}^3$ to $so(3)$, and

$$\boldsymbol{F}(\boldsymbol{\alpha}) = \left[\begin{array}{cccc} 0 & -k_f s_2 & 0 & k_f s_4 \\ -k_f s_1 & 0 & k_f s_3 & 0 \\ k_f c_1 & -k_f c_2 & k_f c_3 & -k_f c_4 \end{array}\right],$$

$$\boldsymbol{\tau}(\boldsymbol{\alpha}) = \left[\begin{array}{ccc} 0 & -Lk_f c_2 - k_m s_2 & 0 \\ -Lk_f c_1 + k_m s_1 & 0 & Lk_f c_3 - k_m s_3 \\ -Lk_f s_1 - k_m c_1 & Lk_f s_2 - k_m c_2 & -Lk_f s_3 - k_m c_3 \\ Lk_f c_4 + k_m s_4 & 0 & Lk_f s_4 - k_m c_4 \end{array}\right]^T \tag{6}$$

being the 3×4 input coupling matrixes ($s_i = \sin(\alpha_i)$ and $c_i = \cos(\alpha_i)$).

A direct inversion of (5) by means of a *static* feedback linearization does not yield a satisfactory solution for the aforementioned 6-dof tracking problem. This is due to the lack of a direct coupling between the output accelerations $(\ddot{\boldsymbol{p}},\, \dot{\boldsymbol{\omega}}_B)$ and the tilting inputs $\boldsymbol{w}_\alpha$. However, as explained in [10], one can resort to a *dynamic* feedback linearization scheme for obtaining the sought result. In fact, by differentiating (5) w.r.t. time, one obtains

$$\left[\begin{array}{c} \dddot{\boldsymbol{p}} \\ \ddot{\boldsymbol{\omega}}_B \end{array}\right] = \boldsymbol{A}(\boldsymbol{\alpha}, \boldsymbol{w}) \left[\begin{array}{c} \dot{\boldsymbol{w}} \\ \boldsymbol{w}_\alpha \end{array}\right] + \boldsymbol{b}(\boldsymbol{\alpha},\, \boldsymbol{w},\, \boldsymbol{\omega}_B) \tag{7}$$

where (i) the new input $\dot{\boldsymbol{w}}$ is the dynamic extension of the former (and actual) input $\boldsymbol{w}$, and (ii) input $\boldsymbol{w}_\alpha$ explicitly appears in the output dynamics. Furthermore, it is possible to prove that the 6×8 coupling matrix $\boldsymbol{A}(\boldsymbol{\alpha},\, \boldsymbol{w})$ has *always* rank $\rho_A = \mathrm{rank}(\boldsymbol{A}) = 6$ provided that $w_i \neq 0$, $i = 1 \dots 4$, i.e., that the propeller spinning never stops, see [10].

Therefore, assuming this rank condition holds, system (7) can be inverted as

$$\begin{bmatrix} \dot{\boldsymbol{w}} \\ \boldsymbol{w}_\alpha \end{bmatrix} = \boldsymbol{A}^\dagger \left(\begin{bmatrix} \dddot{\boldsymbol{p}}_r \\ \ddot{\boldsymbol{\omega}}_r \end{bmatrix} - \boldsymbol{b} \right) + (\mathbf{I}_8 - \boldsymbol{A}^\dagger \boldsymbol{A})\boldsymbol{z}, \tag{8}$$

with $\mathbf{I}_N$ being the identity matrix of dimension N and $\boldsymbol{z} \in \mathbb{R}^8$ a vector projected onto the null-space of $\boldsymbol{A}$ (of dimension 2). The tracking problem under consideration is then solved by choosing

$$\dddot{\boldsymbol{p}}_r = \dddot{\boldsymbol{p}}_d + \boldsymbol{K}_{p_1}(\ddot{\boldsymbol{p}}_d - \ddot{\boldsymbol{p}}) + \boldsymbol{K}_{p_2}(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}) + \boldsymbol{K}_{p_3}(\boldsymbol{p}_d - \boldsymbol{p}) \tag{9}$$

and

$$\ddot{\boldsymbol{\omega}}_r = \ddot{\boldsymbol{\omega}}_d + \boldsymbol{K}_{\omega_1}(\dot{\boldsymbol{\omega}}_d - \dot{\boldsymbol{\omega}}_B) + \boldsymbol{K}_{\omega_2}(\boldsymbol{\omega}_d - \boldsymbol{\omega}_B) + \boldsymbol{K}_{\omega_3}\boldsymbol{e}_R \tag{10}$$

where

$$\boldsymbol{e}_R = \frac{1}{2}[{}^W\boldsymbol{R}_B^T\boldsymbol{R}_d - \boldsymbol{R}_d^T{}^W\boldsymbol{R}_B]_\vee, \tag{11}$$

with $[\cdot]_\vee$ being the inverse map from $so(3)$ to $\mathbb{R}^3$, and the (diagonal) positive definite gain matrices $\boldsymbol{K}_{p_1}$, $\boldsymbol{K}_{p_2}$, $\boldsymbol{K}_{p_3}$, and $\boldsymbol{K}_{\omega_1}$, $\boldsymbol{K}_{\omega_2}$, $\boldsymbol{K}_{\omega_3}$ defining Hurwitz polynomials.

Finally, owing to the actuation redundancy (2-dimensional null-space of $\boldsymbol{A}$), one can exploit vector $\boldsymbol{z}$ in (8) in order to fulfill additional tasks not interfering with the main trajectory tracking objective. To this end, we proposed to minimize a cost function $H(\boldsymbol{w})$ penalizing propeller speeds that are either too low (for preventing $\mathrm{rank}(\boldsymbol{A}) < 6$), or too large (for reducing energy consumption). This was achieved by designing $H(\boldsymbol{w}) = \sum_{i=1}^4 h_i(w_i)$ with the individual $h_i(w_i)$ having a global minimum at a given w_{rest} and growing unbounded as $w_i \to w_{min} < w_{rest}$ and as $w_i \to \infty$. Figure 4

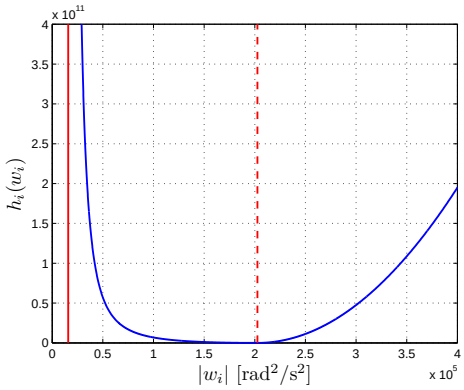


Fig. 4: Example of a function $h_i(w_i)$ with $\bar{w}_{min} = 126$ [rad/s] (solid red line), $\bar{w}_{rest} = 450$ [rad/s] (dashed red line). Note that $h_i(w_i) \to \infty$ as $|w_i| \to w_{min}$ or $|w_i| \to \infty$, and with a minimum at w_{rest} with continuous derivative

shows an example for $\bar{w}_{min} = 126$ [rad/s] and $\bar{w}_{rest} = 450$ [rad/s]. Vector $\boldsymbol{z}$ in (8) is then chosen as

$$\boldsymbol{z} = -k_H \begin{bmatrix} \nabla_{\boldsymbol{w}} H(\boldsymbol{w}) \\ \mathbf{0} \end{bmatrix} \tag{12}$$

with $k_H > 0$ being a suitable gain.

C. Discussion

In view of the next developments, we discuss some remarks about the 'implementability' of the proposed controller (8–12). The controller needs measurement of the position $\boldsymbol{p}$, linear velocity $\dot{\boldsymbol{p}}$ and linear acceleration $\ddot{\boldsymbol{p}}$, of the orientation ${}^W\boldsymbol{R}_B$, angular velocity $\boldsymbol{\omega}_B$ and angular acceleration $\dot{\boldsymbol{\omega}}_B$, and of the tilting angles $\boldsymbol{\alpha}$ and propeller spinning velocities $\boldsymbol{w}$. As it will be explained in Sect. III-B, measurement of the linear position/velocity $(\boldsymbol{p}, \dot{\boldsymbol{p}})$ and of the orientation${}^W\boldsymbol{R}_B$ will be obtained by means of an external visual tracking system, while measurements of the angular velocity $\boldsymbol{\omega}_B$ will be provided by the gyroscopes onboard our prototype. Similarly, direct measurements of $\boldsymbol{\alpha}$ and $\boldsymbol{w}$ will be possible from the low-level motor controllers actuating the spinning and tilting axes.

As for the remaining acceleration measurements $(\ddot{\boldsymbol{p}}, \dot{\boldsymbol{\omega}}_B)$, instead of resorting to numerical differentiation of the corresponding (noisy) velocity quantities, we exploit model (5) and evaluate $(\ddot{\boldsymbol{p}}, \dot{\boldsymbol{\omega}}_B)$ as a function of measured quantities and applied commands. This, of course, requires the additional knowledge of the various model parameters (e.g., mass, inertia, internal geometry), and unavoidably neglects all those effects not captured by (5). The next Sections will however confirm the reasonability of these assumptions for our prototype and the robustness of the proposed controller in coping with these non-idealities.

III. DESCRIPTION OF THE PROTOTYPE AND SYSTEM ARCHITECTURE

A. Prototype

As first prototype developed by our group, we opted for a very low cost solution with all parts available off-the-shelf.

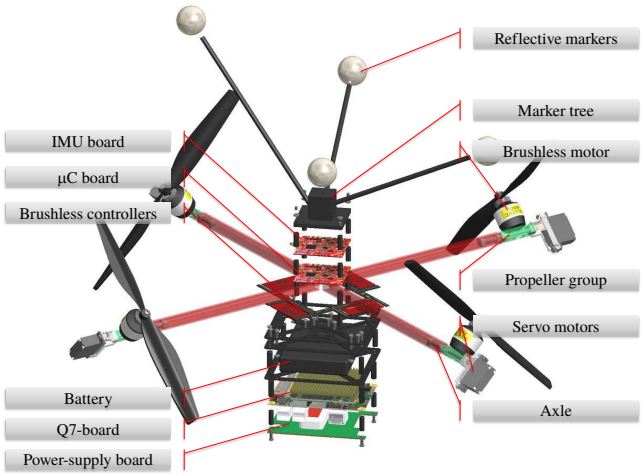


Fig. 5: Exploded view of the various components of the omnicopter. All the important parts are properly labeled

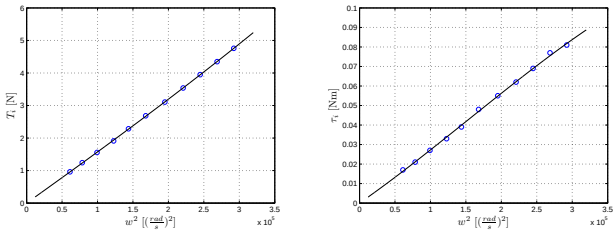
The overall costs including all mechanical and electrical parts and actuators are below 1000 €. The mechanical main frame of the omnicopter is based on the MikroKopter[2] module, including the propeller (EPP1045 CF) and the brushless propeller motors (Roxxy 2827-35). At the end of every arm of the omnicopter body, a rigidly connected axle allows rotation of the propeller groups containing the propeller motor and the servo motor for the tilting actuation (Robbe S3150 Digital), see Fig. 5. This has a maximum torque $\tau_{\alpha_{max}} = 0.37$ Nm and a maximum rotation speed $\dot{\alpha}_{max} = 4.1$ rad/s. The propeller group is designed so that its barycenter is as close as possible to the axle, as assumed by the dynamical model developed in Sect. II-A.

Furthermore, two microcontroller boards are mounted on top of the omnicopter. The first contains the gyroscopes measuring $\boldsymbol{\omega}_B$, and is also in charge of reading the tilting angles α_i of the servo motors and the spinning velocities $\bar{w}_i$ of the propellers. The second microcontroller board sends the desired spinning velocities $\bar{w}_{Des_i}$ to the brushless controller and the desired angles α_{Desi} to the servo motors.

The trajectory tracking controller of Sect. II-B is implemented in Matlab/Simulink and, via the Real-Time Workshop toolbox, is then deployed and executed in real-time on an Intel Atom board (Quadmo747, from now on 'Q7-board') running the Linux Ubuntu10.10 real-time environment. The Q7-board is mounted below the battery and is equipped with a wireless USD-dongle for communication. As only one RS-232 port (TTL level) is available on the Q7-board, the second microcontroller board is connected via one USB-port and USBToSerial converter. The Q7-board is powered by a battery, with the necessary voltage conversion and stabilization performed by a power-supply board containing a 12V DC/DC power converter.

The nominal mass of the full omnicopter is 1.32 Kg. From a high detail CAD model of the body and propeller groups

[2]http://www.mikrokopter.de



(a)

Fig. 6: (a) Dots - measured values of the thrust T_i vs. the signed squared spinning velocity w_i; Line - identified polynomial model (13); (b) Dots - measured values of the torque τ_{D_i} vs. the signed squared spinning velocity w_i; Line - identified polynomial model (14)

we also obtained the following inertia matrixes

$$\boldsymbol{I}_{P_i} = \begin{bmatrix} 8.450e^{-5} & 0 & 0 \\ 0 & 8.450e^{-5} & 0 \\ 0 & 0 & 4.580e^{-5} \end{bmatrix} \left[\mathrm{kg\,m^2}\right]$$

and

$$\boldsymbol{I}_B = \begin{bmatrix} 0.0161 & 0 & 0 \\ 0 & 0.0161 & 0 \\ 0 & 0 & 0.0282 \end{bmatrix} \left[\mathrm{kg\,m^2}\right].$$

In the current setup, the servo motors are limited in their rotation by mechanical end stops in the range of -90 deg $< \alpha_i <$ 90 deg. This limits the rotation of the body frame B to $\pm$55 deg around the roll or pitch axes.

In order to obtain accurate values of k_f and k_m for our motor-propeller combination, we made use of a testbed equipped with a 6-dof torque/force sensor (Nano17-E, see Fig. 7) for identifying the mappings between the propeller spinning velocity and the generated thrust T_i and torque τ_{D_i}. This resulted in the following polynomial models (shown in Fig. 6):

$$T_i = 4.94e^{-18}|w_i|^3 + 9.62e^{-13}|w_i|^2 + 1.56e^{-5}|w_i| \quad (13)$$

and

$$\tau_{D_i} = -5.41e^{-19}|w_i|^3 + 2.50e^{-13}|w_i|^2 - 2.53e^{-7}|w_i| \quad (14)$$

where $w_i = \bar{w}_i|\bar{w}_i|$ as explained before. Controller (8–12) was then implemented by directly exploiting the mappings (13–14) for obtaining (T_i, τ_{D_i}), and by replacing $k_f = \left.\frac{\partial T_i}{\partial w_i}\right|_{w_i}$ and $k_m = \left.\frac{\partial \tau_{D_i}}{\partial w_i}\right|_{w_i}$, both evaluated upon the measured w_i.

B. System architecture

The Q7-board runs a GNU-Linux Ubuntu 10.10 real time OS and executes the Matlab-generated code. The controller runs at 500 Hz and takes as inputs: (i) the desired trajectory $(\boldsymbol{p}_d(t), \boldsymbol{R}_d(t))$ and needed derivatives $(\dot{\boldsymbol{p}}_d(t), \ddot{\boldsymbol{p}}_d(t), \dddot{\boldsymbol{p}}_d(t))$ and $(\boldsymbol{\omega}_d(t), \dot{\boldsymbol{\omega}}_d(t), \ddot{\boldsymbol{\omega}}_d(t))$, (ii) the current position/orientation of the omnicopter $(\boldsymbol{p}, {}^W\boldsymbol{R}_B)$ and its linear/angular velocity $(\dot{\boldsymbol{p}}, \boldsymbol{\omega}_B)$, (iii) the spinning velocities of the propellers w_i, (iv) the tilting angles α_i.

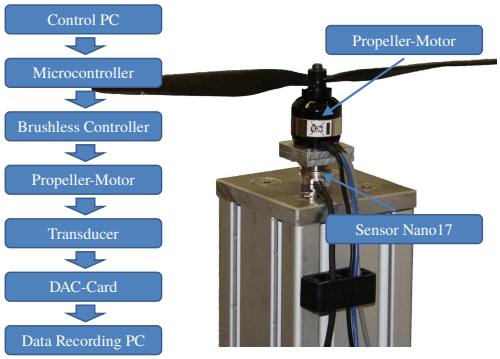


Fig. 7: Left: Scheme of the measurement chain; Right: Motor testbed including Propeller motor combination and Nano17 sensor mounted at a height of 0.45 m

The position $\boldsymbol{p}$ and orientation ${}^W\boldsymbol{R}_B$ of the omnicopter are directly obtained from an external motion caption system[3] (MoCap) at 240 Hz. A marker tree consisting of five infra red markers is mounted on top of the omnicopter for this purpose. Knowing $\boldsymbol{p}$, the linear velocity $\dot{\boldsymbol{p}}$ is then obtained via numerical differentiation, while the angular velocity $\boldsymbol{\omega}_B$ is measured by the onboard IMU (3 ADXRS610 gyroscopes).

Due to performance reasons (bottleneck in serial communication), the sending of the desired motor speeds and tilting angles, and the reading of the IMU-data, of the actual spinning velocities, and of tilting angles is split among two communication channels and two microcontrollers (called, from now on, 'μC-Board' and 'IMU-Board'). The desired motor spinning velocities w_{Des_i} are sent from the Q7-board to the μC-Board via a serial connection at the frequency of 250 Hz and 8 bit resolution, and from the μC-Board to the brushless controllers via I^{2}C-bus at again 250 Hz (see Fig. 8). The brushless controllers implement a PID-controller for regulating the spinning velocity. The desired tilting angles α_{Des_i} are sent from the Q7-board to the μC-Board via the same serial connection at a frequency of 55 Hz and 10 bit resolution, and from the μC-Board to the servo motors via PWM (signal length 15 ms). We note that the trajectory tracking controller described in Sect II-B assumes availability of the *tilting velocities* w_{α_i}, see (8), while the current architecture only allows for sending desired angles commands α_{Des_i}. This is addressed by numerically integrating over time the controller commands, that is, by implementing $\alpha_{Des_i} = \int_{t_0}^{t} w_{\alpha_i}(\tau)\mathrm{d}\tau$.

The IMU-Board reads the current angles α_i of the propeller groups P_i by a direct connection between the servo motor potentiometer and the A/D-converter of the microcontroller (10 bit resolution at 250 Hz). It also retrieves the current spinning velocities $\bar{w}_i$ of the propellers via the I^{2}C-Bus (8 bit resolution and 250 Hz). The gyro data are read at 250 Hz and converted with 10 bit resolution. Finally, the values of α_i, $\bar{w}_i$ and of the gyro data are transmitted from the IMU-Board to the Q7-board via the RS232-port at 250 Hz. All values of the controller can be monitored on a

[3] http://www.vicon.com/products/bonita.html

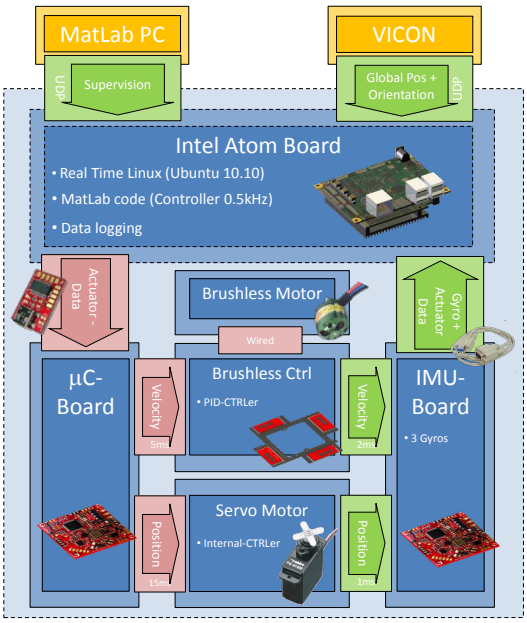


Fig. 8: Overview of the omnicopter architecture including update rates and delays

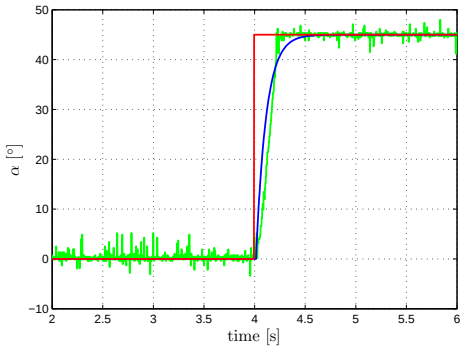


Fig. 9: Modeling of the Servomotor. Behavior of the real servo motor (green) and the model (blue) following a step input (red) of 45° after compensating for the (known) transport delay $T = 18$ ms

remote Windows PC which mirrors the running controller in real time using the matlab/simulink "external mode". This simplifies the development as most of the gains and settings can be changed online during flight tests.

The communication architecture for the tilting angles α_{Des_i} (in particular, the PWM modulation) unfortunately introduces a non-negligible roundtrip delay of about 18 ms form sent command to read values. We experimentally found this delay to significantly degrade the closed-loop performance of the controller, and therefore propose in the next Sect. III-C a simple prediction scheme for mitigating its adverse effects.

C. Coping with the non-idealities of the servo motors

The i-th servo motor for the tilting angles can be approximately modeled as a linear transfer function $G(s)$ with, in series, a transport delay of $T = 18$ ms, that is, as the delayed linear system $\alpha_i(s) = G(s)e^{-Ts}\alpha_{des_i}(s)$. A model of the undelayed $G(s)$ was experimentally obtained by measuring the step response (Fig. 9) of the servo motors while having

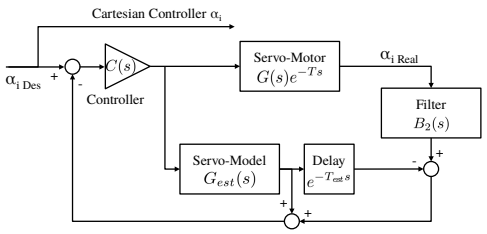


Fig. 10: Scheme of the Smith predictor for α_i including the controller $C(s)$, the servo motor $G(s)e^{-Ts}$, the model of the servo motor $G_{est}(s)e^{-Ts}$, and the Butterworth filter $B_2(s)$

the propellers spinning at $\bar{w}_i = 420$ rad/s (the velocity corresponding to hovering), and by compensating offline for the known delay T. This resulted into the estimated transfer function

$$G_{est}(s) = \frac{0.4s + 6}{0.06s^2 + s + 6}. \tag{15}$$

The performance degradation of the cartesian trajectory controller (8–12) can then be ascribed to two main effects, namely presence of the transport delay T and slow dynamic response of $G_{est}(s)$ to fast changing inputs. In order to mitigate these shortcomings, we resorted to the following simple strategy (see Fig. 10): instead of feeding back the measured (i.e., delayed) angles α_i to the cartesian controller (8–12), we replaced them with the (undelayed) desired angles α_{des_i}. In parallel, we aimed at improving the servo motor performance (i.e., making $G_{est}(s)$ more responsive) via a Smith predictor scheme [13]. In fact, as well-known from classical control theory, the Smith predictor is an effective tool for coping with *known* delays affecting *known* stable linear systems. In our case, an additional outer PID controller $C(s)$ plugged into the Smith predictor loop, as shown in Fig. 10, allowed to improve the rising time of the servo controller.

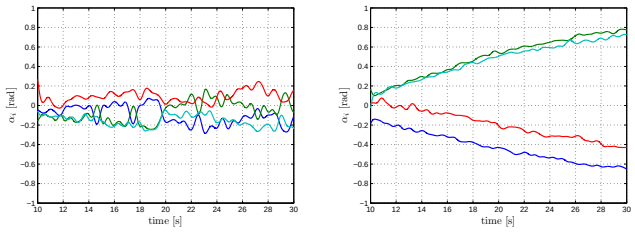
Finally, since we found the measured angles α_i to be affected by significant noise, we filtered their readings with a 2nd order Butterworth filter with a cutoff frequency of 20Hz. The location of this cutoff frequency was experimentally determined by analyzing offline the power spectrum of the angles α_i recorded during a hovering flight of 40 s.

IV. EXPERIMENTAL RESULTS

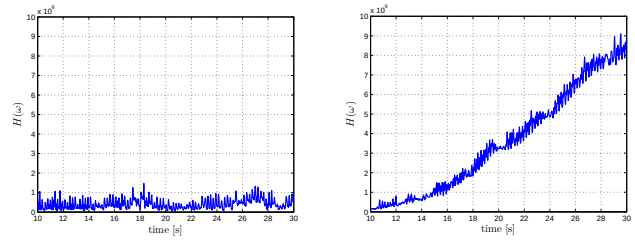
In this section we will present results from three experiments run with our prototype and aimed at validating our modeling and control approach. The first experiment is a hovering task meant to show the performance of the controller in the simplest scenario, and also to highlight the importance of having included the null-space optimization term (12) in the control strategy. The other two experiments are intended to show the performance in tracking position/orientation trajectories which would be unfeasible for standard quadrotor UAVs: a circular trajectory with constant (horizontal) attitude and a rotation on the spot.

A. Hovering on the spot

In the first experiment, we show the importance of having included the minimization of the cost function $H(\boldsymbol{w})$ in the

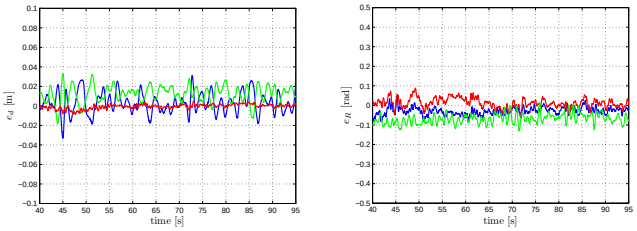


(a) (b)



(c) (d)

Fig. 11: Results for hovering on spot with (i) and without (ii) including the null-space term (12): (a) α_i for case (i) while hovering; (b) α_i for case (ii) while hovering; (c) $H(\boldsymbol{w})$ for case (i) while hovering; (d) $H(\boldsymbol{w})$ for case (ii) while hovering



(a) (b)

Fig. 12: Tracking error while hovering: (a) Position tracking error e_p; (b) Orientation tracking error e_R

proposed controller. To this end, we report the results of a simple hovering on the spot by (i) including and (ii) not including the null-space optimization term (12). The quadrotor starts from the initial state of $\boldsymbol{p}(t_0) = \mathbf{0}$, $\dot{\boldsymbol{p}}(t_0) = \mathbf{0}$, ${}^W\boldsymbol{R}_B(t_0) = \mathbf{I}_3$, $\boldsymbol{\omega}_B(t_0) = \mathbf{0}$, $\boldsymbol{\alpha}(t_0) = \mathbf{0}$, and $\boldsymbol{w}(t_0) = w_{rest}$, and is commanded to stay still while maintaining the desired attitude $\boldsymbol{R}_d = \mathbf{I}_3$. We set the gains in (9) and (10) to $\boldsymbol{K}_{p_1} = 0.48\mathbf{I}_3$, $\boldsymbol{K}_{\omega_1} = 30\mathbf{I}_3$, $\boldsymbol{K}_{p_2} = 0.48\mathbf{I}_3$, $\boldsymbol{K}_{\omega_2} = 300\mathbf{I}_3$, $\boldsymbol{K}_{p_3} = 0.48\mathbf{I}_3$, $\boldsymbol{K}_{\omega_3} = 1000\mathbf{I}_3$. Figures 11(a)–(c) report the results for case (i): the angles α_i stay close to 0 rad over time, as expected for such a hovering maneuver, and $H(\boldsymbol{w})$ keeps a constant and low value as the propellers spin with a speed close to the allowed minimum. In case (ii), however, the situation looks completely different: the lack of any minimization action of $H(\boldsymbol{w})$, coupled with presence of noise and non-idealities, makes the angles α_i to eventually diverge over time from their (expected) vertical direction and, accordingly, the value of $H(\boldsymbol{w})$ to increase as the propellers need to accelerate in order to keep the quadrotor still in place (Figs. 11(b)–(d)).

Finally, Figs. 12(a–b) show the position error $\boldsymbol{e}_p = \boldsymbol{p}_d - \boldsymbol{p}$ and orientation error $\boldsymbol{e}_R$ during the experiment. The average position tracking error is about 0.017 m with a maximum of 0.047 m. The maximum rotation errors are -0.082 rad (roll), -0.131 rad (pitch), and 0.089 rad (yaw).

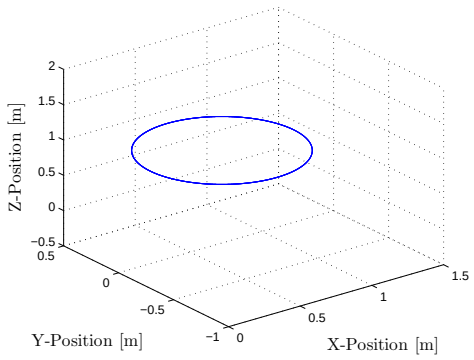
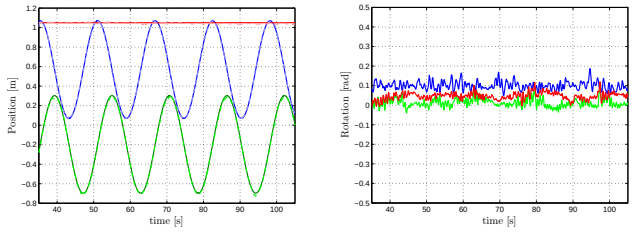


Fig. 13: Desired horizontal circular trajectory $\boldsymbol{p}_d(t)$ at a height of $z = 1.05$ m



(a) (b)

Fig. 14: (a) Desired (solid) and measured (dashed) position in x (blue), y (green) and z (red); (b) Measured orientation in roll (red), pitch (green) and yaw (blue)

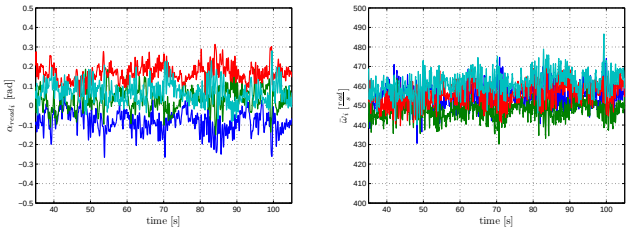
B. Circular trajectory

In this experiment we demonstrate the ability of the omnicopter to follow an arbitrary trajectory in space while keeping a desired orientation (constant in this case). As explained, this would be unfeasible for a standard quadrotor UAV. The chosen desired trajectory is a horizontal circle with diameter of 1 m and lying at a height of $z = 1.05$ m from ground, see Fig. 13. The quadrotor is commanded to travel along the path with a constant speed of 0.2 m/s while keeping the main body parallel w.r.t. the ground. We set the gains in (9) and (10) to the same values as in Sect. IV-A.

Figure 14(a) shows the desired (solid) and real (dashed) position of the omnicopter while following the trajectory (the two plots are almost coincident), while Fig. 14(b) reports the orientation error $\boldsymbol{e}_R$ (the desired orientation during the trajectory was set to $\boldsymbol{R}_d = \mathbf{I}_3$). The maximum position error $\max(\|\boldsymbol{e}_p(t)\|)$ while following the path was approximately 4.2 cm, with $\text{avg}(\|\boldsymbol{e}_p(t)\|) \approx 1.6$ cm. The maximum orientation errors were 0.12 rad for roll, 0.08 rad for pitch and 0.18 rad for yaw. Figures 15(a)-(b) show the behavior of the tilting angles α_i, and of the motor spinning velocities $\bar{w}_i$ which kept close to $\bar{w}_{rest}$=450 rad/s as expected.

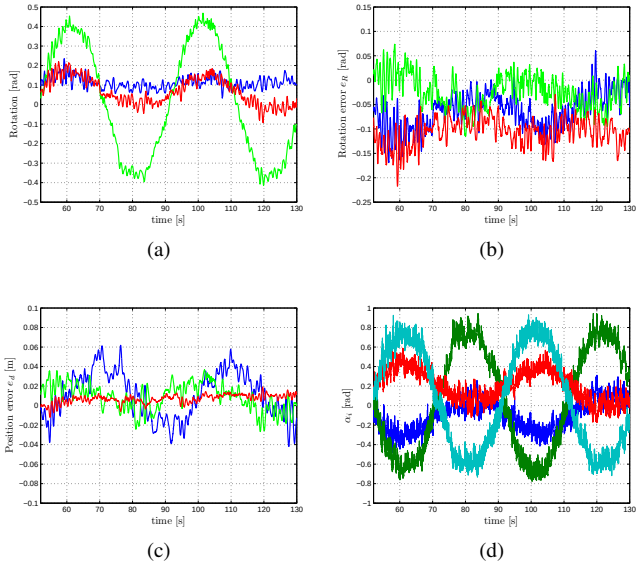
C. Rotation on spot

In the last experiment we demonstrate the tracking abilities in following a given orientation profile $\boldsymbol{R}_d(t)$ while keeping the same position in space. Again, this maneuver is clearly unfeasible for a standard quadrotor UAV. The



(a) (b)

Fig. 15: Behavior of the titling angles α_i (a) and of the propeller speeds $\bar{w}_i$ (b) while performing the horizontal circular trajectory



(a) (b)

(c) (d)

Fig. 16: Rotation on the spot around the $\boldsymbol{Y}_B$-Axis: (a) Orientation of the main body B; (b) and (c) orientation and position error vectors; (d) behavior of the tilting angles α_i

followed trajectory is a sinusoidal rotation around the $\boldsymbol{Y}_B$-axis (pitching), i.e., $\boldsymbol{R}_d(t) = \boldsymbol{R}_Y(\theta(t))$ with $\theta_{max} = 0.436\ rad$ and $\dot{\theta}_{max} = 0.07$ rad/s. The initial conditions were set to hovering ($\boldsymbol{p}(t_0) = \boldsymbol{0}$, $\dot{\boldsymbol{p}}(t_0) = \boldsymbol{0}$, ${}^W\boldsymbol{R}_B(t_0) = \mathbf{I}_3$, $\boldsymbol{\omega}_B(t_0) = \boldsymbol{0}$, $\boldsymbol{\alpha}(t_0) = \boldsymbol{0}$, and $\boldsymbol{w}(t_0) = w_{rest}$) and the controller gains as in Sect. IV-A.

Figures 16 (a-d) show the results of the flight. In particular, Fig. 16(a) reports the quadrotor orientation during flight (blue - roll, green - pitch, red - yaw), and Fig. 16(b) the orientation tracking error $\boldsymbol{e}_R(t)$. The position tracking error $\boldsymbol{e}_p(t)$ is shown in Fig. 16(c), with a maximum of $\max(\|\boldsymbol{e}_p(t)\|) = 0.062$ m. Finally, Fig. 16(d) depicts the behavior of the tilting angles $\alpha_i(t)$ during the maneuver, and Fig. 17 the behavior of $H(\boldsymbol{w})$ during flight. As clear from the plots, this experiment

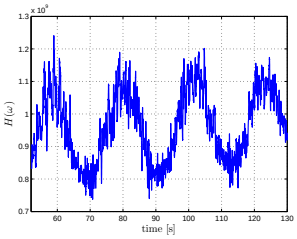


Fig. 17: Behavior of $H(\boldsymbol{w})$ while rotating on the spot

involving a rotation on the spot still confirms the capabilities of the omnicopter and the robustness of the proposed control strategy in coping with all the non-idealities of real-world conditions. The interested reader can also appreciated the reported maneuvers in the video attached to this paper.

V. CONCLUSIONS

In this paper, we have addressed the hardware/software design and control implementation for a recently-developed prototype of a novel quadrotor UAV with tilting propellers — the 'omnicopter'. Contrarily to standard quadrotors, the omnicopter design allows to actively rotate the 4 propellers about the axes connecting them to the main body. This makes it possible to obtain full controllability over the 6-dof body pose in space, thus overcoming the underactuation hindering standard quadrotor UAVs. The reported experiments, although preliminary, have clearly shown good potential of the UAV in various experiments. After having obtained these promising results, confirming the validity of our design, our next step is to build an improved prototype with a better actuation system. This will allow, on one side, to gain a higher tracking accuracy, and, on the other side, to fully exploit the omnicopter 6-dof motion capabilities also in interaction tasks with the environment.

REFERENCES

[1] S. Bouabdallah, M. Becker, and R. Siegwart, "Autonomous miniature flying robots: Coming soon!" *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 88–98, 2007.

[2] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[3] EU Collaborative Project ICT-248669 AIRobots, "www.airobots.eu."

[4] EU Collaborative Project ICT-287617 ARCAS, "www.arcas-project.eu."

[5] R. Naldi, L. Gentili, L. Marconi, and A. Sala, "Design and experimental validation of a nonlinear control law for a ducted-fan miniature aerial vehicle," *Control Engineering Practice*, vol. 18, no. 7, pp. 747–760, 2010.

[6] K. T. Oner, E. Cetinsoy, M. Unel, M. F. Aksit, I. Kandemir, and K. Gulez, "Dynamic model and control of a new quadrotor unmanned aerial vehicle with tilt-wing mechanism," in *Proc. of the 2008 World Academy of Science, Engineering and Technology*, 2008, pp. 58–63.

[7] K. T. Oner, E. Cetinsoy, E. Sirimoglu, C. Hancer, T. Ayken, and M. Unel, "LQR and SMC stabilization of a new unmanned aerial vehicle," in *Proc. of the 2009 World Academy of Science, Engineering and Technology*, 2009, pp. 554–559.

[8] F. Kendoul, I. Fantoni, and R. Lozano, "Modeling and control of a small autonomous aircraft having two tilting rotors," *IEEE Trans. on Robotics*, vol. 22, no. 6, pp. 1297–1302, 2006.

[9] A. Sanchez, J. Escareño, O. Garcia, and R. Lozano, "Autonomous hovering of a noncyclic tiltrotor UAV: Modeling, control and implementation," in *Proc. of the 17th IFAC Wold Congress*, 2008, pp. 803–808.

[10] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Modeling and Control of a Quadrotor UAV with Tilting Propellers," in *2012 IEEE Int. Conf. on Robotics and Automation*, 2012.

[11] R. Falconi and C. Melchiorri, "Dynamic Model and Control of an Over-Actuated Quadrotor UAV," in *Proc. of the 10th IFAC Symposium on Robotic Control*, 2012.

[12] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Oldenburg, 2004.

[13] W. A. Wolovich, *Automatic Control Systems*. Oxford University Press, 1994.