Hardware and Software Architecture for Nonlinear Control of Multirotor Helicopters

Sammy Omari, Minh-Duc Hua, Guillaume Ducard, *Member, IEEE*, and Tarek Hamel, *Member, IEEE*

***Abstract*—This paper presents the design and implementation of a nonlinear control scheme for multirotor helicopters that takes first-order drag effects into account explicitly. A dynamic model including the blade flapping and induced drag forces is provided and a hierarchical nonlinear controller is presented. This controller is designed for both high-precision flights as well as robustness against model uncertainties and external disturbances. This is achieved by using saturated integrators with fast desaturation properties. The implementation of the controller on the flybox hexacopter platform is described. The hardware and software architecture of this UAV is discussed, and useful hints and insights gained during its design process are presented. Finally, experimental results and videos are reported to demonstrate the successful implementation and the performance of the overall system.**

***Index Terms*—Hardware and software architecture, hierarchical nonlinear control, multirotor helicopter.**

I. Introduction

During the last decade, a tremendous interest for aerial robots and especially for multirotor helicopters, with vertical takeoff and landing (VTOL) capabilities, has been seen worldwide both in academia and industry. Early work in the control of VTOL vehicles was based on helicopter platforms in the late nineties [34], [38]. The first scientific works on quadrotor platforms were undertaken ten years ago [1], [10]. Thanks to their small size, simple mechanics, high performance and availability, they quickly became the universal testbed for indoors aerial robotics research [3], [4], [7], [18], [19], [29], [32], [39], [40]. Some recent control algorithms developed for quadrotors have led to impressive performance allowing the vehicle to execute aggressive maneuvers [20], [26], [36]. These previous results were obtained when the vehicle's position, orientation, and velocity are measured precisely at high-frequency, using an external 3-D tracking system. All recent examples of the high-performance quadrotor control are obtained in indoor environment and rely on full-state measurements obtained from an external 3-D tracking system, limiting their applicability to experimental flight areas properly equipped. Results found in the literature for quadrotor platforms using onboard sensors show far less aggressive maneuvers [2], [7], [11], [28]. This is related to the increased weight of the platform due to onboard position sensors as well as low onboard processing power for pose estimation of the UAV.

More recently, other multirotor platforms, based on the quadrotor principle, such as hexacopters have emerged. They have increasingly attracted the attention of the aerial robotics community for their higher payload or fault-tolerance capabilities [6], [35]. Multirotor helicopters are used to perform advanced missions that require HLs of autonomy. One motivating example is the use of small-scale multirotor helicopters for the inspection of industrial plants. Such inspection tasks require the implementation of a complex control architecture to enable safe and high-precision position controlled flight in close vicinity to structures. Therefore, an appropriate hardware and software architecture is of primary importance to provide the robot with the autonomy required by the missions. While state estimation and control aspects have been widely discussed [2], [13], [21], [23], [24], the hardware design as well as the implementation of control schemes for VTOL drones are only scarcely documented [7], [8], [27]. In contrast, this paper presents a holistic framework for high-performance flight of UAVs that incorporates the design of a nonlinear flight controller but also discusses the necessary mechanical, electrical, and software framework.

In early works on modeling of multirotor helicopters [10], the interaction between the propellers and the vehicle's dynamics was often neglected, in particular, the blade flapping and induced drag dynamics. In fact, these aerodynamic phenomena have been thoroughly studied in the literature for classical helicopters with large rotors (e.g., [30]). Only recently these aerodynamic drag forces have been taken into account in the modeling of small scale quadrotors [18], [25], [29]. So far, the flapping dynamics have been used for improving the estimation of the translational velocity and orientation [2], [22]. Inspired by those works, we derive a nonlinear control scheme that takes into account and compensates for these first-order aerodynamic drag forces. Thanks to its cascade structure, the proposed control scheme is simple to implement and tune, with a real-time computational effort comparable to the one required for linear controllers.

To summarize, the key points addressed in this paper are: 1) the presentation of a hierarchical nonlinear control approach, respectful of the helicopter's motion and allowing for a large domain of operation; 2) the inclusion of first-order aerodynamic

Manuscript received September 7, 2012; revised January 10, 2013 and April 25, 2013; accepted June 6, 2013. Date of publication August 8, 2013; date of current version December 11, 2013. Recommended by Technical Editor H. Gao. This work was support by the ECHORD TUAV project.

S. Omari is with the Autonomous System Laboratory of ETH Zurich, 8092 Zurich, Switzerland, and also with Skybotix, 8051 Zurich, Switzerland (e-mail: omaris@ethz.ch).

M.-D. Hua is with the ISIR CNRS-UPMC, 75252 Paris Cedex 05, France (e-mail: hua@isir.upmc.fr).

G. Ducard and T. Hamel are with the I3S UNS-CNRS, 06903 Sophia Antipolis, France (e-mail: ducard@i3s.unice.fr; thamel@i3s.unice.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2013.2274558

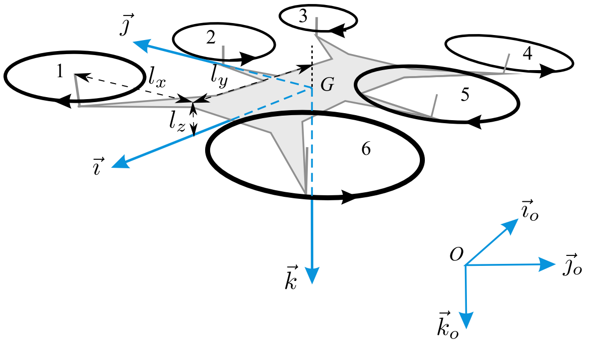


Fig. 1. Schematic representation of a six-rotor UAV platform. Inertial frame $\mathcal{I}$ with origin O and the body-fixed frame $\mathcal{B}$ with origin G.

drag forces due to blade flapping and induced drag in the control design; 3) the addition of effective antiwindup integrators into the control laws in order to enhance the robustness against unmodeled dynamics and aerodynamic disturbances; and 4) the description and implementation of the hardware and software architecture used aboard the flybox hexacopter.

This paper is organized as follows. Section II presents the dynamic model of a generic multirotor VTOL vehicle, including a description of the blade flapping and induced drag forces. In Section III, a generic nonlinear hierarchical control strategy is described for a large class of VTOL vehicles. In Section IV, the experimental framework is presented. The overall hardware architecture of the flybox is detailed and the implementation aspects of the control approach are discussed. Section V provides the experimental results which show the performance of the control scheme. Finally, conclusions are given in Section VI.

II. System Modeling

This section presents the mathematical flight model used throughout this study. The most basic multirotor helicopter configuration consists of a rigid airframe with two pairs of counter-rotating rigid propellers attached to it. The control of this platform is achieved by varying the rotational speed of the rotors. While such a four-rotor configuration already allows for full actuation of the vehicle's attitude, this approach can be easily extended to six- or eight-rotor configurations. In general, the configuration can be scaled up to an arbitrary number of rotors, however, the configuration should always consist of a multiple of counter-rotating rotor pairs for torque balancing reasons. In Fig. 1, a schematic of the flybox hexacopter, described in Section IV, is depicted.

The vehicle's center of mass (CoM) is denoted as G, its mass m, and its inertia matrix $\mathbf{J}$. Let $\mathcal{I} = \{O; \vec{\imath}_o, \vec{\jmath}_o, \vec{k}_o\}$ and $\mathcal{B} = \{G; \vec{\imath}, \vec{\jmath}, \vec{k}\}$ denote the inertial frame (i.e., world frame) and the frame attached to the vehicle, respectively. Let $\boldsymbol{\xi} := [x\ y\ z]^\top \in \mathbb{R}^3$ denote the position of the vehicle's CoM expressed in $\mathcal{I}$. The rotation matrix representing the orientation of the frame $\mathcal{B}$ relatively to the frame $\mathcal{I}$ is $\mathbf{R} \in SO(3)$. The vehicle's velocity and the wind velocity are both expressed in the frame $\mathcal{I}$ are denoted as $\dot{\boldsymbol{\xi}} \in \mathbb{R}^3$ and $\dot{\boldsymbol{\xi}}_w \in \mathbb{R}^3$, respectively. Let $\boldsymbol{\omega} \in \mathbb{R}^3$ be the angular velocity of the frame $\mathcal{B}$ expressed in $\mathcal{B}$. The canonical basis of $\mathbb{R}^3$ is denoted $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Let $\mathbf{d}_i = [d_{1,i}\ d_{2,i}\ d_{3,i}]^\top \in \mathbb{R}^3$ be the position of the ith rotor expressed in the body-fixed frame $\mathcal{B}$. We define $\mathbf{d}_i^\perp = [d_{1,i}\ d_{2,i}\ 0]^\top \in \mathbb{R}^3$ as the component of $\mathbf{d}_i$ perpendicular to $\mathbf{e}_3$. Let the thrust direction of all rotors be parallel to $\mathbf{e}_3$ in $\mathcal{B}$. The notation $_\times$ represents the skew-symmetric matrix associated with the cross product, i.e., $\mathbf{u}_\times\mathbf{v} = \mathbf{u} \times \mathbf{v}\, \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^3$. The Euclidean norm in $\mathbb{R}^n$ is denoted as $|\cdot|$.

A. Dynamic Model of the Vehicle

Following the model proposed in [10], the ith rotor, turning at ϖ_i, generates a thrust force $\mathbf{F}_{t,i} = c_T \varpi_i^2 \mathbf{e}_3$ and an aerodynamic torque $\mathbf{Q}_i = \lambda_i c_Q \varpi_i^2 \mathbf{e}_3$ with the aerodynamic constants c_T and c_Q and $\lambda_i = \{-1, 1\}$, depending on the direction of rotation of the rotor (cw: $\lambda_i = 1$, ccw: $\lambda_i = -1$). Additionally, each rotor imposes a drag force $\mathbf{F}_{d,i}$ on the vehicle due to blade flapping and induced drag, as explained in Section II-B. The remaining aerodynamic forces and torques (mostly due to drag by the fuselage) are summed up in a vector $\mathbf{F}_{\text{aero}} \in \mathbb{R}^3$ and $\boldsymbol{\Gamma}_{\text{aero}} \in \mathbb{R}^3$, respectively. The vehicle is subject to gravity $mg\mathbf{e}_3$.

Applying the Newton–Euler formalism, one obtains the following equations of motion of the vehicle [14]:

$$\begin{cases} m\ddot{\boldsymbol{\xi}} = \mathbf{R}\sum_i \mathbf{F}_i + mg\mathbf{e}_3 + \mathbf{F}_{\text{aero}} & \text{(1a)} \\ \dot{\mathbf{R}} \quad = \mathbf{R}\boldsymbol{\omega}_\times & \text{(1b)} \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times \mathbf{J}\boldsymbol{\omega} + \sum_i (\mathbf{Q}_i + \mathbf{d}_i \times \mathbf{F}_i) + \boldsymbol{\Gamma}_{\text{aero}} & \text{(1c)} \end{cases}$$

where $\mathbf{F}_i$ is the sum of thrust and drag force generated by each rotor.

B. Rotor Aerodynamics

Blade flapping and induced drag are of significant importance for understanding the natural stability of multirotor systems. These forces require special attention since they act on the rotor plane, and thus, affect the underactuated translational dynamics of the UAV. Aerodynamic drag due to blade flapping affects the UAV in forward flight. More precisely, when the UAV is in forward flight, the advancing rotor blade has a higher tip velocity and will, therefore, generate more lift than the retreating blade. Since the rotor blades are not completely rigid, the blade flaps. As well explained in [30], the spinning rotor with high angular momentum acts like a gyroscope and the *phase lag* phenomenon occurs. This phenomenon is the difference in phase (approximately 90°) between the position of the greatest tip velocity of a rotor blade and the position of its greatest upward flap. The flapping movement of the blade remains in an equilibrium in a constant speed forward flight. In this equilibrium, the rotor will have a constant flapping angle away from the forward velocity of the UAV. Since the thrust of the rotor is perpendicular to the tilted rotor plane, the horizontal component of the thrust can be seen as a damping force counteracting the current forward velocity, as depicted in Fig. 2.

For any kind of airfoil generating lift, there is an associated induced drag proportional to the lift that it produces, as depicted in Fig. 3. This is due to the fact, that the lift generated by the

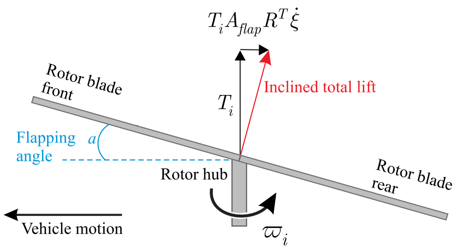


Fig. 2. Aerodynamic drag due to blade flapping. Schematic inspired by [22].

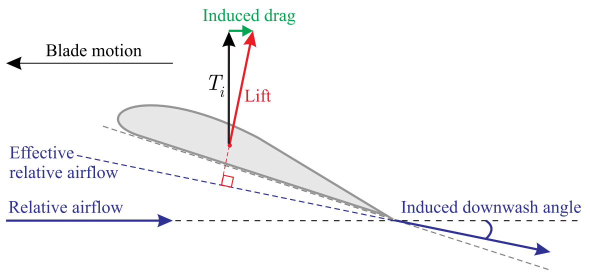


Fig. 3. Aerodynamic force due to induced drag.

airfoil is tilted backwards by an angle corresponding to the induced downwash angle. During hovering, the forces are equally distributed and are responsible for the aerodynamic torque $\mathbf{Q}_i$. However, in forward flight, the advancing blade experiences more lift, and therefore, generates more induced drag as the retreating one. This results in a net force that is opposing the current forward velocity.

We can write the blade flapping dynamics as instantaneous functions of the vehicles velocity since the blade flapping dynamics are very fast compared to the vehicle dynamics. To model the effect of blade flapping on the vehicle, we first compute the advance ratio μ_{ri} and azimuthal direction ψ_{ri} of each rotor

$$\mu_{ri} = \frac{|\mathbf{v}_{ri,1,2}|}{\varpi_i r}, \quad \psi_{ri} = \arctan 2(v_{ri,2}, v_{ri,1})$$

where $\mathbf{v}_{ri}$ is the linear velocity that the ith rotor hub experiences given by $\mathbf{v}_{ri} = \mathbf{R}^\top(\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_w) + \boldsymbol{\omega} \times \mathbf{d}_i$, and r denotes the radius of the propeller.

Inspired by [22] and [29] the lateral and longitudinal flapping angles a_i and b_i can be computed as

$$\begin{bmatrix} a_i \\ b_i \end{bmatrix} = \begin{bmatrix} C\psi_{ri} & -S\psi_{ri} \\ S\psi_{ri} & C\psi_{ri} \end{bmatrix} \begin{bmatrix} a_{si} \\ b_{si} \end{bmatrix}$$

with

$$a_{si} = \frac{c_a \mu_{ri}}{1 - \frac{\mu_{ri}^2}{2}}, \quad b_{si} = \frac{c_b \mu_{ri}}{1 + \frac{\mu_{ri}^2}{2}}$$

and c_a and c_b denoting some positive parameters. From here, one deduces that

$$\begin{bmatrix} a_i \\ b_i \end{bmatrix} = \begin{bmatrix} \frac{c_a}{1 - \frac{\mu_{ri}^2}{2}} & -\frac{c_b}{1 + \frac{\mu_{ri}^2}{2}} \\ \frac{c_b}{1 + \frac{\mu_{ri}^2}{2}} & \frac{c_a}{1 - \frac{\mu_{ri}^2}{2}} \end{bmatrix} \mathbf{v}_{ri,1,2}$$
$$\approx \begin{bmatrix} c_a & -c_b \\ c_b & c_a \end{bmatrix} \mathbf{v}_{ri,1,2}$$

where the last approximation is justified by the fact that the vehicle's velocity is "small" compared to the propeller tip velocity so that the term $\mu_{ri}^2/2$ can be neglected compared to 1. From here, one verifies that

$$\sum_i (\mathbf{F}_{t,i} + \mathbf{F}_i^{\text{flap.}}) = -T\mathbf{e}_3 - c_T \mathbf{A}_{\text{flap.}} \sum_i \varpi_i^2 \mathbf{v}_{ri}$$
$$= -T\mathbf{e}_3 - T\mathbf{A}_{\text{flap.}} (\mathbf{R}^\top(\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_w) - d_{3,i}\boldsymbol{\omega} \times \mathbf{e}_3)$$
$$+ \mathbf{A}_{\text{flap.}}\, \boldsymbol{\omega} \times \sum_i T_i \mathbf{d}_i^\perp \tag{2}$$

with the thrust magnitude

$$T = c_T \sum_i \varpi_i^2 \tag{3}$$

and the flapping matrix

$$\mathbf{A}_{\text{flap.}} := \begin{bmatrix} c_a & -c_b & 0 \\ c_b & c_a & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

We can further simplify (2) to

$$\sum_i (\mathbf{F}_{t,i} + \mathbf{F}_i^{\text{flap.}}) \approx -T\mathbf{e}_3 - T\mathbf{A}_{\text{flap.}} \mathbf{R}^\top \dot{\boldsymbol{\xi}}$$

by assuming that the wind speed $\dot{\boldsymbol{\xi}}_w$ is negligible, the vertical distance of the rotor plane to the CoM is very small ($d_{3,i}\boldsymbol{\omega} \times \mathbf{e}_3 \approx 0$) and that the yaw angular rates of the vehicle are negligible ($\mathbf{A}_{\text{flap.}}\, \boldsymbol{\omega} \times \sum_i T_i \mathbf{d}_i^\perp \approx 0$).

The drag force caused by the induced drag in forward flight of the rotor i can be modeled by $\mathbf{F}_i^{i.d.} \approx -T_i \mathbf{A}_{i.d.} \mathbf{v}_{ri} \approx -T_i \mathbf{A}_{i.d.} \mathbf{R}^\top \dot{\boldsymbol{\xi}}$, with $\mathbf{A}_{i.d.} = \text{diag}(c_{dx}, c_{dy}, 0)$ where c_{dx}, c_{dy} are the induced drag coefficients [22]. Finally, we sum up both drag forces in a lumped parameter model as

$$\sum_i \mathbf{F}_i \approx \sum_i (\mathbf{F}_{t,i} + \mathbf{F}_i^{\text{flap.}} + \mathbf{F}_i^{i.d.}) \approx -T\mathbf{e}_3 - T\mathbf{A}_{\text{drag}} \mathbf{R}^\top \dot{\boldsymbol{\xi}}$$

with $\mathbf{A}_{\text{drag}} = \mathbf{A}_{\text{flap.}} + \mathbf{A}_{i.d.}$.

Equivalently to the translational dynamics, the effect of the drag forces on the rotational dynamics can be expressed as

$$\sum_i \mathbf{Q}_i + \mathbf{d}_i \times \mathbf{F}_i = \boldsymbol{\Gamma} - \sum_i T_i \mathbf{d}_i \times \mathbf{A}_{\text{drag}} \mathbf{v}_{ri}$$

with

$$\boldsymbol{\Gamma} = \sum_i \lambda_i c_Q \varpi_i^2 \mathbf{e}_3 - c_T \varpi_i^2 \mathbf{d}_i^\perp \times \mathbf{e}_3. \tag{4}$$

C. Model for the Control Design and Rate Control

The UAV model (1) can be rewritten as

$$\begin{cases} \Sigma_1 : \begin{bmatrix} m\ddot{\boldsymbol{\xi}} \\ \dot{\mathbf{R}} \end{bmatrix} = \begin{bmatrix} -T\mathbf{R}\mathbf{e}_3 + \mathbf{F}_e \\ \mathbf{R}\boldsymbol{\omega}_\times \end{bmatrix} & \text{(5a)} \\ \Sigma_2 : \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\Gamma} + \boldsymbol{\Gamma}_e & \text{(5b)} \end{cases}$$

where $\mathbf{F}_e$ and $\boldsymbol{\Gamma}_e$ are the sum of all the acting forces and moments on the vehicle except the thrust force $T\mathbf{R}\mathbf{e}_3$ and the torque $\boldsymbol{\Gamma}$ by the rotors defined by (3) and (4), respectively.

From (3) and (4), one can view $T \in \mathbb{R}^+$ and $\boldsymbol{\Gamma} \in \mathbb{R}^3$ as control inputs of the system (5). For N mounted rotors, we can rewrite (3) as a linear mapping from the square of the propellers' angular velocity to the total thrust T and torque $\mathbf{\Gamma}$ as follows:

$$\begin{bmatrix} T \\ \mathbf{\Gamma} \end{bmatrix} = \begin{bmatrix} c_T & c_T & \dots & c_T \\ c_T\, d_{2,1} & c_T\, d_{2,2} & \dots & c_T\, d_{2,N} \\ -c_T\, d_{1,1} & -c_T\, d_{1,2} & \dots & -c_T\, d_{1,N} \\ \lambda_1\, c_Q & \lambda_2\, c_Q & \dots & \lambda_N\, c_Q \end{bmatrix} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \vdots \\ \varpi_N^2 \end{bmatrix}. \tag{6}$$

If $N = 4$, one can determine the desired angular rates of the rotors by inverting (6). When the UAV is actuated by more than four rotors, the set of equations (6) is overdetermined and the Moore–Penrose pseudoinverse method can be used to determine the desired angular velocities of the propellers. Additionally, if the UAV is actuated by more than four rotors, robustness to motor or rotor failure is increased. By reformulating the control allocation equation (6) in case of a failure, attitude control may still be achieved [35].

The system (5) shows full actuation of the rotational dynamics and underactuation of the translational dynamics. For the rotational motion, exponential convergence of the angular velocity $\boldsymbol{\omega}$ to any bounded desired value $\boldsymbol{\omega}_d$ is easy to obtain, since the subsystem Σ_2 is fully actuated and the angular velocity vector $\boldsymbol{\omega}$ can be measured at high frequency from embedded gyrometers. A possible control solution [33], [40]

$$\mathbf{\Gamma} = -\mathbf{J}\mathbf{K}_\omega(\boldsymbol{\omega} - \boldsymbol{\omega}_d) + \boldsymbol{\omega}_d \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}\dot{\boldsymbol{\omega}}_d - \mathbf{\Gamma}_e \tag{7}$$

with a diagonal positive gain matrix $\mathbf{K}_\omega$, yields the closed-loop equation

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}(\boldsymbol{\omega} - \boldsymbol{\omega}_d) - \mathbf{J}\mathbf{K}_\omega(\boldsymbol{\omega} - \boldsymbol{\omega}_d)$$

and thus, the exponential stability of $\boldsymbol{\omega} = \boldsymbol{\omega}_d$ with a rate of convergence given by $\mathbf{K}_\omega$. In practice, one can neglect the term $\mathbf{\Gamma}_e$ in (7) and choose a sufficiently high gain matrix $\mathbf{K}_\omega$ to dominate the disturbance torque $\mathbf{\Gamma}_e$. From here on, all attention of the control design can be given to the control of the subsystem Σ_1 using T and $\boldsymbol{\omega} \equiv \boldsymbol{\omega}_d$ as control inputs.

III. Hierarchical Nonlinear Control Architecture

Due to the underactuated nature of the translational motion, controllability is related to the nonlinear coupling term $T\mathbf{R}\mathrm{e}_3$ between the thrust magnitude T and the thrust direction characterized by the unit vector $\boldsymbol{\eta} := \mathbf{R}\mathrm{e}_3 \in \mathbb{S}^2$. The translational motion subsystem can be controlled by using the magnitude of the thrust vector T and the thrust direction $\boldsymbol{\eta}$ as intermediary control variables. Fig. 4 shows the proposed cascade control structure for several control modes associated with different levels of motion autonomy. We consider the following three modes: 1) thrust orientation control; 2) velocity control; and (c) position control. The hierarchical control approach adopted here is inspired by the two control approaches proposed in [14] and [28]. A "low-level" (LL) fast inner loop stabilizes the vehicle's thrust direction (i.e., thrust magnitude + thrust orientation) and a "high-level" (HL) slow outer-loop controls the translational dynamics. This control strategy is based on the principles of: 1) using the thrust magnitude and the vehicle's thrust orientation as intermediary control variables to control the translational dynamics, and 2) applying a high-gain controller to stabilize the desired thrust direction using the angular velocity as control input.

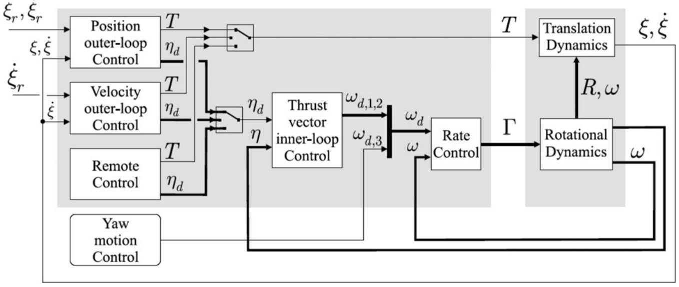


Fig. 4. Hierarchical nonlinear control architecture.

A. Inner-Loop Thrust-direction Control

The objective is to stabilize the thrust orientation $\boldsymbol{\eta}$ to a given desired unit vector $\boldsymbol{\eta}_d \in \mathbb{R}^3$ which can be specified either by a remote control (RC) or by the intermediary control of the outer loop (see Fig. 4). This objective is achieved by computing the first and second components of the desired angular velocity $\boldsymbol{\omega}_{d,1,2} := (\omega_{d,1}, \omega_{d,2})^\top$ as follows [14]:

$$\boldsymbol{\omega}_{d,1,2} = \left(\mathbf{R}^\top \left(k_\eta \frac{\boldsymbol{\eta} \times \boldsymbol{\eta}_d}{(1+\boldsymbol{\eta}^\top\boldsymbol{\eta}_d)^2} - (\boldsymbol{\eta}_\times)^2(\boldsymbol{\eta}_d \times \dot{\boldsymbol{\eta}}_d)\right)\right)_{1,2} \tag{8}$$

with a positive gain k_η. This controller is derived by considering the positive function $\mathcal{V} := 1 - \boldsymbol{\eta}^\top\boldsymbol{\eta}_d$ whose time derivative along any solution to the closed-loop system is $\dot{\mathcal{V}} = -k_\eta \frac{|\boldsymbol{\eta}\times\boldsymbol{\eta}_d|^2}{(1+\boldsymbol{\eta}^\top\boldsymbol{\eta}_d)^2}$. Exponential stability of the equilibrium $\boldsymbol{\eta} = \boldsymbol{\eta}_d$ is thus ensured, provided that $\boldsymbol{\eta}(0) \neq -\boldsymbol{\eta}_d(0)$ (see [14] for the proof). The control law (8) indicates that only the first two components of $\boldsymbol{\omega}_d$, i.e., $\boldsymbol{\omega}_{d,1,2}$, are involved in the realization of the thrust orientation control objective. Thus, the remaining component of $\boldsymbol{\omega}_d$, i.e., $\boldsymbol{\omega}_{d,3}$, can be used for a complementary objective related to the yaw motion.

B. Outer-Loop Velocity Control

The objective consists in stabilizing the vehicle's translational velocity $\dot{\boldsymbol{\xi}}$ to a reference vector $\dot{\boldsymbol{\xi}}_r \in \mathbb{R}^3$ using the thrust intensity T and the thrust orientation $\boldsymbol{\eta} \equiv \boldsymbol{\eta}_d$ as control inputs. Denote $\ddot{\boldsymbol{\xi}}_r$ as the time derivative of $\dot{\boldsymbol{\xi}}_r$ and $\dot{\tilde{\boldsymbol{\xi}}} := \dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_r$ as the velocity error. Let us introduce also either the integral of the velocity error $\tilde{\boldsymbol{\xi}} := \int_0^t \dot{\tilde{\boldsymbol{\xi}}}(s)ds$ or the position tracking error $\tilde{\boldsymbol{\xi}} := \boldsymbol{\xi} - \boldsymbol{\xi}_r$ when a reference trajectory $\boldsymbol{\xi}_r$ is specified. Using (5a), one obtains the following translational error dynamics:

$$\ddot{\tilde{\boldsymbol{\xi}}} = -\frac{1}{m}T\boldsymbol{\eta} + \boldsymbol{\gamma}_e - \ddot{\boldsymbol{\xi}}_r \tag{9}$$

with the external acceleration $\boldsymbol{\gamma}_e := \mathbf{F}_e/m$. Equation (9) can be rewritten as

$$\begin{cases} \ddot{\tilde{\boldsymbol{\xi}}} = -h(\dot{\tilde{\boldsymbol{\xi}}}) - \dfrac{1}{m}T\boldsymbol{\eta} + \boldsymbol{\gamma} \\ \boldsymbol{\gamma} := h(\dot{\tilde{\boldsymbol{\xi}}}) + \boldsymbol{\gamma}_e - \ddot{\boldsymbol{\xi}}_r \end{cases} \tag{10}$$

with $h : \mathbb{R}^3 \to \mathbb{R}^3$ a bounded function chosen in order to make $\dot{\tilde{\boldsymbol{\xi}}} = 0$ a globally asymptotically stable equilibrium of the equation $\ddot{\tilde{\boldsymbol{\xi}}} = -h(\dot{\tilde{\boldsymbol{\xi}}})$. We now have to ensure the asymptotic stability of $m\boldsymbol{\gamma} - T\boldsymbol{\eta} = 0$. To this end, assuming that $\boldsymbol{\gamma}$ does not vanish, we propose the following outer-loop control:

$$(T, \boldsymbol{\eta}_d) = (m|\boldsymbol{\gamma}|, \boldsymbol{\gamma}/|\boldsymbol{\gamma}|). \tag{11}$$

Applying the inner-loop control $\boldsymbol{\omega}_{d,1,2}$ defined by (8) ensures the asymptotic stability of $\boldsymbol{\eta} = \boldsymbol{\eta}_d$, and thus, of $m\boldsymbol{\gamma} - T\boldsymbol{\eta} = 0$. Finally, the asymptotic stability of $\dot{\tilde{\boldsymbol{\xi}}} = 0$ directly follows (see [14] and [15] for more details of the proof).

In the traditional literature on control of small-scale UAVs, gravity is considered to be the only external force acting on the system. The computation of $\boldsymbol{\gamma}$ is thus straightforward. If the flapping and induced drag forces are included, the computation of $\boldsymbol{\gamma}$ is more involved. This is due to the fact that the drag forces are a function of the orientation $\mathbf{R}$ of the UAV. With the inclusion of the drag forces in the controller, the desired acceleration, which determines the desired orientation of the UAV, is now a function of the orientation itself. As a consequence, a closed-form solution for the desired acceleration may not exist (see [14] and [15] for more discussions on this issue). In Section III-D, we discuss a novel solution to this issue. For now, we assume that we approximately know the external acceleration $\boldsymbol{\gamma}_e$ and that it is independent of both the thrust magnitude T and the rotation matrix $\mathbf{R}$.

Note that the desired thrust orientation $\boldsymbol{\eta}_d$ given in (11) is not well defined if the term $\boldsymbol{\gamma}$ vanishes (see [14] and [15] for more discussions on this issue). In order to limit the risk of a vanishing $\boldsymbol{\gamma}$, a function $h(\dot{\tilde{\boldsymbol{\xi}}})$ is chosen to be uniformly smaller in norm than the gravity constant g (corresponding to the norm of $\boldsymbol{\gamma}_e$ when hovering in the absence of wind). We propose to consider

$$h(\dot{\tilde{\boldsymbol{\xi}}}) = \begin{bmatrix} \mathrm{sat}_{\Delta_1^v}(k_1^v \dot{\tilde{\boldsymbol{\xi}}}_{1,2}) \\ \mathrm{sat}_{\Delta_3^v}(k_3^v \dot{\tilde{\xi}}_3) \end{bmatrix} \tag{12}$$

with the classical saturation function $\mathrm{sat}_\Delta(\mathbf{x}) := \mathbf{x}\min(1, \Delta/|\mathbf{x}|)\ \forall \mathbf{x} \in \mathbb{R}^n$, a pair of positive gains (k_1^v, k_3^v), and a pair of positive parameters (Δ_1^v, Δ_3^v) here chosen to be smaller than $g/2$. Note that the decoupling of the first two components of $\dot{\tilde{\boldsymbol{\xi}}}$, i.e., $\dot{\tilde{\boldsymbol{\xi}}}_{1,2}$, and its last component $\dot{\tilde{\xi}}_3$ in the definition (12) of $h(\dot{\tilde{\boldsymbol{\xi}}})$ allows for the local decoupling of the vehicle's horizontal and vertical dynamics.

In practice, whatever the method, the estimation of the external acceleration vector $\boldsymbol{\gamma}_e$ (see, e.g., [16]) is never totally accurate. To further add robustness with respect to the imprecise knowledge of $\boldsymbol{\gamma}_e$ and other unmodeled dynamics, an integral term should be incorporated into the control law. To this end and also to limit the integral wind-up effects largely discussed in the literature (e.g., [14], [17], and [37]), we propose the following bounded nonlinear integrator of $\dot{\tilde{\boldsymbol{\xi}}}$:

$$\begin{cases} \dot{\mathbf{z}}_{1,2} = -k_1^z \mathbf{z}_{1,2} + \mathrm{sat}_{0.5\dot{z}_1^{\max}}(k_1^z(\mathbf{z}_{1,2} + \dot{\tilde{\boldsymbol{\xi}}}_{1,2})) \\ \dot{z}_3 = -k_3^z z_3 + \mathrm{sat}_{0.5\dot{z}_3^{\max}}(k_3^z(z_3 + \dot{\tilde{\xi}}_3)),\ \mathbf{z}(0) = 0 \end{cases} \tag{13}$$

with $k_1^z, k_3^z, \dot{z}_1^{\max}, \dot{z}_3^{\max}$ some positive numbers. This relation yields the following upper-bounds: $|\mathbf{z}_{1,2}| \leq \dot{z}_1^{\max}/(2k_1^z)$, $|\dot{\mathbf{z}}_{1,2}| \leq \dot{z}_1^{\max}$, $|z_3| \leq \dot{z}_3^{\max}/(2k_3^z)$, and $|\dot{z}_3| \leq \dot{z}_3^{\max}$. Defining a new tracking error variable $\dot{\tilde{\boldsymbol{\xi}}}_z := \dot{\tilde{\boldsymbol{\xi}}} + \mathbf{z}$ and using (9), one obtains

$$\begin{cases} \ddot{\tilde{\boldsymbol{\xi}}}_z = -h(\dot{\tilde{\boldsymbol{\xi}}}_z) - \dfrac{1}{m}T\boldsymbol{\eta} + \boldsymbol{\gamma}_z^v \\ \boldsymbol{\gamma}_z^v := h(\dot{\tilde{\boldsymbol{\xi}}}_z) + \boldsymbol{\gamma}_e - \ddot{\boldsymbol{\xi}}_r + \dot{\mathbf{z}} \end{cases} \tag{14}$$

with $h(\dot{\tilde{\boldsymbol{\xi}}}_z)$ defined by (12) with $\dot{\tilde{\boldsymbol{\xi}}}$ replaced by $\dot{\tilde{\boldsymbol{\xi}}}_z$. Modulo the difference between the expressions of $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}_z^v$, the equation of systems (10) and (14) are identical. From here, the outer-loop control $(T, \boldsymbol{\eta}_d)$ can be defined as in (11) with $\boldsymbol{\gamma}$ replaced by $\boldsymbol{\gamma}_z^v$. Then, applying the inner-loop control (8) ensures the asymptotic stability of $(\dot{\tilde{\boldsymbol{\xi}}}_z, \boldsymbol{\eta}) = (0, \boldsymbol{\eta}_d)$. Finally, it is easy to show that the convergence of $\dot{\tilde{\boldsymbol{\xi}}}_z$ to zero also ensures the one of $\dot{\tilde{\boldsymbol{\xi}}}$ to zero (see, e.g., [17] for a detailed proof).

C. Outer-Loop Position Control

The tracking of a reference trajectory $\boldsymbol{\xi}_r \in \mathbb{R}^3$ can be achieved via an adequate modification of the function $h(\cdot)$ in the system (10). For instance, the following new expression of $h(\cdot)$ [compare to (12)]:

$$h(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}}) = \begin{bmatrix} \mathrm{sat}_{\Delta_1^p}(k_1^p \tilde{\boldsymbol{\xi}}_{1,2}) + \mathrm{sat}_{\Delta_1^v}(k_1^v \dot{\tilde{\boldsymbol{\xi}}}_{1,2}) \\ \mathrm{sat}_{\Delta_3^p}(k_3^p \tilde{\xi}_3) + \mathrm{sat}_{\Delta_3^v}(k_3^v \dot{\tilde{\xi}}_3) \end{bmatrix} \tag{15}$$

with $k_1^p, k_1^v, k_3^p, k_3^v, \Delta_1^p, \Delta_1^v, \Delta_3^p, \Delta_3^v$ some positive numbers, ensures the almost global asymptotic stability of the equilibrium $(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}}) = (0, 0)$ of the nominal system $\ddot{\tilde{\boldsymbol{\xi}}} = -h(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}})$. Thus, the use of $h(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}})$ instead of $h(\dot{\tilde{\boldsymbol{\xi}}})$ in the definition of $\boldsymbol{\gamma}$ in (10), the outer-loop control (11) of $(T, \boldsymbol{\eta}_d)$ with the new $\boldsymbol{\gamma}$, and the inner-loop control (8) of $\boldsymbol{\omega}_{d,1,2}$ ensure the almost global stability of the equilibrium $(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}}, \boldsymbol{\eta}) = (0, 0, \boldsymbol{\eta}_d)$, provided that $\boldsymbol{\gamma}$ never crosses zero. Note that the saturation functions in the definition (15) of $h(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}})$ are introduced in order to reduce the risk of a vanishing $\boldsymbol{\gamma}$.

In a similar way as in the case of velocity control, we propose to add an integral term into the control law in order to enhance its robustness with respect to unmodeled dynamics. The following bounded integrator of $\tilde{\boldsymbol{\xi}}$ is applied [14]:

$$\begin{aligned} \ddot{\mathbf{z}}_{1,2} &= -2k_1^z \dot{\mathbf{z}}_{1,2} - (k_1^z)^2(\mathbf{z}_{1,2} - \mathrm{sat}_{\Delta_1^z}(\mathbf{z}_{1,2})) + \mathrm{sat}_{\Delta_1^{pz}}(k_1^{pz} \tilde{\boldsymbol{\xi}}_{1,2}) \\ \ddot{z}_3 &= -2k_3^z \dot{z}_3 - (k_3^z)^2(z_3 - \mathrm{sat}_{\Delta_3^z}(z_3)) + \mathrm{sat}_{\Delta_3^{pz}}(k_3^{pz} \tilde{\xi}_3) \\ \mathbf{z}(0) &= 0, \quad \dot{\mathbf{z}}(0) = 0 \end{aligned} \tag{16}$$

with $k_1^z, k_1^{pz}, k_3^z, k_3^{pz}, \Delta_1^z, \Delta_1^{pz}, \Delta_3^z, \Delta_3^{pz}$ some positive numbers. Defining new tracking error variables $\tilde{\boldsymbol{\xi}}_z := \tilde{\boldsymbol{\xi}} + \mathbf{z}$ and $\dot{\tilde{\boldsymbol{\xi}}}_z := \dot{\tilde{\boldsymbol{\xi}}} + \dot{\mathbf{z}}$ and using (9), one deduces

$$\begin{cases} \ddot{\tilde{\boldsymbol{\xi}}}_z = -h(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z) - \dfrac{1}{m}T\boldsymbol{\eta} + \boldsymbol{\gamma}_z^p \\ \boldsymbol{\gamma}_z^p := h(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z) + \boldsymbol{\gamma}_e - \ddot{\boldsymbol{\xi}}_r + \ddot{\mathbf{z}} \end{cases} \tag{17}$$

with the function $h(\cdot)$ defined by (15) with $(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}})$ replaced by $(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z)$. Now, similarly to the position controller described previously, one can define the outer-loop control (11) of $(T, \boldsymbol{\eta}_d)$ with $\boldsymbol{\gamma}$ replaced by $\boldsymbol{\gamma}_z^p$ and apply the inner-loop control (8) of $\boldsymbol{\omega}_{d,1,2}$ in order to ensure the convergence of $\tilde{\boldsymbol{\xi}}_z$ to zero. From here, using the definition of $\tilde{\boldsymbol{\xi}}_z$ and the dynamics of $\mathbf{z}$ (16), it is not difficult to show the convergence of $\tilde{\boldsymbol{\xi}}$ to zero (see [14] for more details). The following remark concerns the integrator (16). As pointed out in [14], the second-order time-derivative of $\mathbf{z}$ can be bounded by an arbitrary positive number specified by the user. For instance, $|\ddot{\mathbf{z}}_{1,2}(t)| \le z_1^{\max} := 6((k_1^z)^2\Delta_1^z + \Delta_1^{pz})$ and $|\ddot{z}_3(t)| \le z_3^{\max} := 6((k_3^z)^2\Delta_3^z + \Delta_3^{pz})$ $\forall t$. Therefore, a small bound of $\ddot{\mathbf{z}}$ can be designed to limit the risk of a vanishing $\boldsymbol{\gamma}_z^p$, a necessary condition for well defining the outer-loop control.

D. Including Blade Flapping and Induced Drag

When drag forces due to blade flapping and induced drag are incorporated in the control design, one may include them in the external force as $\mathbf{F}_e = mg\mathbf{e}_3 - T\mathbf{R}\mathbf{A}_{\text{drag}}\mathbf{R}^\top\dot{\boldsymbol{\xi}}$. However, as explained in Section III-B, the desired acceleration that determines the desired thrust orientation $\boldsymbol{\eta}_d = (\mathbf{R}\mathbf{e}_3)_d$ of the UAV is now a function of the orientation. This implies that the control expression of $\boldsymbol{\omega}_{1,2}$ given in (8) is implicit since it depends on the time derivative of $\boldsymbol{\eta}_d$, i.e., $\dot{\boldsymbol{\eta}}_d$, which is a function of $\boldsymbol{\omega}$. Therefore, a closed-form control solution may not exist. However, we can rewrite the drag forces as

$$T\mathbf{R}\mathbf{A}_{\text{drag}}\mathbf{R}^\top\dot{\boldsymbol{\xi}} \approx T\mathbf{R}\begin{bmatrix} c_a + c_d & 0 & 0 \\ 0 & c_a + c_d & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{R}^\top\dot{\boldsymbol{\xi}}$$

$$\approx T(c_a + c_d)\left(\mathbf{R}\mathbf{R}^\top - \mathbf{R}\mathbf{e}_3\mathbf{e}_3^\top\mathbf{R}^\top\right)\dot{\boldsymbol{\xi}} \approx T(c_a + c_d)\left(\dot{\boldsymbol{\xi}} - \mathbf{R}\mathbf{e}_3 v_3\right)$$

where v_3 is the vehicle's velocity along the $\overrightarrow{k}$ direction fixed to the frame $\mathcal{B}$ and assuming that the vehicle is symmetrical $c_{dx} = c_{dy} = c_d$ and that the coefficient c_b is negligible compared to $c_a + c_d$.

Revisiting the translational dynamics model (5a), one can now include the drag forces as

$$\begin{aligned} m\ddot{\boldsymbol{\xi}} &= -T\mathbf{R}\mathbf{e}_3 + \mathbf{F}_e \\ &= -T\mathbf{R}\mathbf{e}_3 + mg\mathbf{e}_3 - T(c_a + c_d)\left(\dot{\boldsymbol{\xi}} - \mathbf{R}\mathbf{e}_3 v_3\right) \\ &= -T(1 - (c_a + c_d)v_3)\mathbf{R}\mathbf{e}_3 - T(c_a + c_d)\dot{\boldsymbol{\xi}} + mg\mathbf{e}_3. \end{aligned} \tag{18}$$

Furthermore, the additive drag term can be approximately rewritten as

$$T(c_a + c_d)\dot{\boldsymbol{\xi}} = (mg + \delta)(c_a + c_d)\dot{\boldsymbol{\xi}} \approx mg(c_a + c_d)\dot{\boldsymbol{\xi}} \tag{19}$$

assuming that the offset δ between the magnitude of the actual thrust and the hovering thrust is small and is compensated by the integral term in the controller when the UAV is moving at constant velocity. In case of a varying velocity, the tracking error is compensated by a sufficiently large gain of the D-part of the controller.

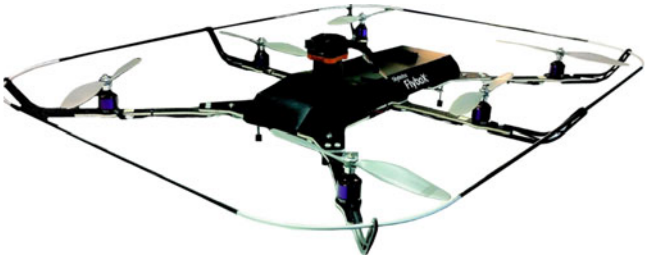


Fig. 5. Flybox hexacopter by Skybotix.

Using (18) and (19) and by defining the augmented control thrust T_a that compensates for the drag by

$$T_a := T\left(1 - (c_a + c_d)v_3\right) \tag{20}$$

one can rewrite the integrator-augmented translational error dynamics as [compared with (17)]

$$\begin{aligned} \ddot{\tilde{\boldsymbol{\xi}}}_z &\approx -\frac{1}{m}T_a\boldsymbol{\eta} + g\mathbf{e}_3 - g(c_a + c_d)\dot{\boldsymbol{\xi}} - \ddot{\boldsymbol{\xi}}_r + \ddot{\mathbf{z}} \\ &\approx -h(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z) - \frac{1}{m}T_a\boldsymbol{\eta} + \boldsymbol{\gamma}_z^p - g(c_a + c_d)\dot{\tilde{\boldsymbol{\xi}}}_z \end{aligned} \tag{21}$$

with the apparent acceleration $\boldsymbol{\gamma}$ now defined by

$$\boldsymbol{\gamma}_z^p := g\mathbf{e}_3 - g(c_a + c_d)(\dot{\boldsymbol{\xi}}_r - \dot{\mathbf{z}}) - \ddot{\boldsymbol{\xi}}_r + \ddot{\mathbf{z}}. \tag{22}$$

From here, one can apply the outer-loop position controller exactly as described in Section III-B using $(T_a, \boldsymbol{\eta})$ as control inputs instead of $(T, \boldsymbol{\eta})$. The term $-g(c_a + c_d)\dot{\tilde{\boldsymbol{\xi}}}_z$ involved in the right-hand side (RHS) of (21) does not affect the stability property of the controller since it is a dissipative term.

IV. Experimental Framework

A. Flybox Platform

1) Platform Description: The flybox platform shown in Fig. 5 is a small-scale hexacopter which has been designed mainly for the industrial inspection market. Reliability, safety, endurance and ease-of-use are the most important design aspects. The flybox configuration described here is a hexacopter configuration, consisting of three counter-rotating propellers pairs (see Fig. 1). While a four-rotor configuration already allows for full actuation of the vehicle's attitude, a six-rotor configuration was chosen for more payload and fault tolerance. In the failure case of one or, in some cases, two motors, attitude control can still be achieved [35].

Keeping the total weight of the system low was of utmost importance since extra weight directly reduces the maximal achievable flight time. As a rule of thumb, every $3 - 4$ g of extra weight requires 1 W of motor power to lift it [41]. Therefore, the structure of the platform is built using a lightweight, yet rigid carbon–balsa–wood sandwich structure. The main body consists of a plate where the LL autopilot and the HL computer are mounted. The low center of mass below the rotor plane as well as the flat and plate-like structure of the UAV reduces the agility of the UAV but renders it more stable during hovering. The overall size of the platform is 740 mm × 900 mm × 100 mm with a weight of 950 g when equipped with minimal payload.

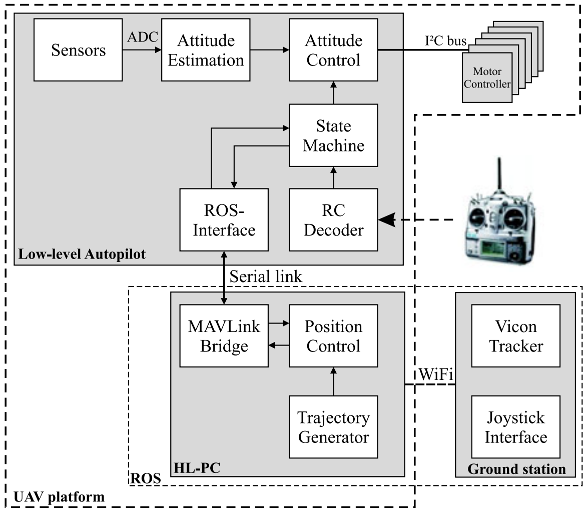


Fig. 6. System architecture of the flybox UAV.

The propulsion group consists of Hacker brushless motors with 10-in propellers and Mikrokopter motor controllers. The motor controllers allow for closed-loop rotor-speed control, thus ensuring good rotor-speed tracking irrespective of the battery voltage. The propulsion group was optimized so that the UAV is hovering at approximately 50% of the maximal thrust. It has been shown that this configuration allows for the highest torque-control possible without running into motor saturation [6].

Using an off-the-shelf three-cell Lithium Polymer (LiPo) with a capacity of 8000 mAh, the flight time reaches 30 min when flying with minimal payload. When a payload of 450 g is added, the achievable flight time reduces to 20 min.

2) Vibration Reduction: The vibrations, mainly induced by the rotor blades, can cause severe problems for the attitude estimation. In practice, even with balanced rotor blades, the noise on the inertial sensors can overwhelm the useful signals. A Fourier analysis of the accelerometer and gyrometer output showed that there was a large noise level on the whole frequency spectrum with peaks at the multiples of the rotation rate of the rotors. Since the signal-to-noise ratio was poor, it was decided to decouple the sensors from the main platform, instead of only using digital filters. Initially, only the LL autopilot, containing the inertial sensors, was decoupled from the main platform using industrial silicone dampers. However, since the weight of the decoupled mass was very small compared to the rest of the platform, the results were rather poor—even when using very soft dampers with low Shore numbers. The decoupled mass has been increased by adding the battery to it. This resulted in acceptable noise levels on the inertial sensors, even in the presence of strong vibrations caused by unbalanced rotors. Therefore, the inertial measurement unit (IMU) is now rigidly fixed to the battery housing which is then decoupled from the main frame using industrial foam dampers.

B. Hardware Architecture

The overall hardware architecture as depicted in Fig. 6 is presented next. The most time- and safety-critical tasks, such as estimation and control of the vehicle's attitude are performed on the LL autopilot. The LL autopilot tracks a reference thrust orientation coming from either a HL on board computer or directly from a human operator over the RC. The advantage of this cascade system is that, once the attitude controller is tuned, the LL platform can be considered as a black box which directly accepts acceleration commands. Therefore, when designing a HL position or velocity controller, the UAV can be considered as a point-mass moving in 3-D space since all the inherent nonlinearities of the attitude controller are hidden in the LL autopilot.

The HL computer is used to perform less time critical and more computationally demanding tasks, such as position- and velocity control or obstacle avoidance. The HL computer is aboard the flybox and can be interfaced from a ground station (GS) computer over WiFi.

1) LL Autopilot: The custom-made autopilot is built around the *STM32F103VE* Cortex M3 32-bit microcontroller. The microcontroller runs at 72 MHz and has 64 kbytes of RAM and 512 kbytes of flash memory. The autopilot is equipped with a custom-made IMU consisting of a three-axis gyrometer, a three-axis accelerometer, and a three-axis magnetometer as well as a GPS and a pressure sensor.

The autopilot is performing the following LL tasks:

1) interfacing, filtering and calibrating the inertial sensors at 10 kHz;
2) estimating and controlling the attitude at 1 kHz;
3) interfacing the motor controllers at 1 kHz;
4) sending and receiving messages from the HL computer at up to 1 kHz;
5) decoding RC pulse-position modulation messages at 50 Hz;
6) detecting and handling events such as a motor or rotor failure or loss of RC signal at 50 Hz.

A light-weight task scheduler is implemented to run periodic tasks up to 10 kHz on the microcontroller. Since the Cortex M3 does not provide a floating point unit, all code is written in fixed-point ANSI-C code to enable high filtering, estimation, and control update rates.

2) Sensors: While selecting the sensors, a lot of effort was devoted to choosing sensors which provide enough bandwidth to accurately capture the dynamics of the vehicle while still providing maximal resolution. Another important criterion was vibration rejection of the sensors, since the autopilot may be subject to substantial vibrations during operation of the UAV.

The autopilot contains a three-axis MEMS gyrometer *ADXRS610* by analog devices. This gyrometer has been selected for its good vibration rejection capabilities as well as very stable output with only very slow temporal zero-offset drift. Additionally, the autopilot is equipped with a three-axis MEMS accelerometer *MXR9500* by Memsic.

Both sensors are sampled by the analog to digital converter (ADC) of the microcontroller at 10 kHz using direct memory access. The sensor output is digitally processed using a FIR filter. To optimally determine the cutoff frequency of the filter, the angular and translational dynamics of the UAV were modeled, given the inertia of the flybox as well as the time constant of the motors. Then, the cutoff frequency was set to exceed the dynamics of the UAV by a factor of 2.

The autopilot is also equipped with a magnetometer, a GPS sensor and a high-resolution pressure sensor. There are numerous other papers discussing estimation and control strategies using those sensors on UAVs in detail [19], [42]. Therefore, we omit discussion of this sensor setup.

3) HL Computer: The HL computer is a strapped down, off-the-shelf Atom 1.6 GHz with 1 GB DDR2 RAM. The computer is both lightweight and low in power consumption (< 6 W). To save weight and increase reliability, a flash drive is used as a hard drive. The operating system is an Ubuntu 12.04 Server version. The choice of a standard x86 processor and a regular Linux OS makes the development of HL software particularly easy, because the developer can rely on a large array of highly optimized and widely used software libraries.

The computer provides its own onboard wireless access point and can be accessed using the SSH protocol over this link.

To run applications on the HL computer, the robotic operating system (ROS) [31] is used as middleware. It is based on an architecture where processing takes place in nodes that may communicate with each other using messages transmitted using the UDP or TCP network protocol. As a consequence, ROS can be run transparently over several machines. This enables the developer to outsource computationally expensive, yet non-time-critical tasks to a more powerful ground station with no power- and weight limitations. However, in the setup presented in this paper, the ROS ground station is only used to display the current state of the UAV and to interface the joystick that is used as input to the position controller. For safety reasons, all HL tasks such as position- and velocity control and obstacle avoidance are performed onboard the UAV. Therefore, even in the case where the data link to the ground station is perturbed, safe operation of the UAV is still ensured.

4) Communication Link of the LL Autopilot to the HL Computer: The HL computer interfaces the LL autopilot over a bidirectional high-speed serial interface at 1 MBaud. Using this link, the LL autopilot sends a variety of messages to the HL computer such as current attitude, sensor data, RC data, state machine state, health of LL autopilot, etc. In the opposite direction, the HL computer can have full control over the LL autopilot by controlling its internal state machine. Additionally, the HL computer can send reference thrust vectors to the autopilot. However, for safety reasons the user can always supersede the HL commands using the RC.

For the communication link, the Mavlink protocol is used. This protocol enables to send and receive custom-defined messages reliably with only little overhead. On the HL computer, the messages are received by the *Mavlink-bridge* ROS node and published as custom ROS messages. The refresh rate of the messages varies from 1 Hz (state machine state) to 1 kHz (IMU sensor information) and can be set during runtime using the dynamic reconfiguration feature of the ROS.

For control and estimation, it is highly important to have an accurate time synchronization of the two platforms. This is done using an NTP-like approach: A message is sent from the HL PC to the LL autopilot at HL time t_1, it is received at LL time s_1 and sent back at s_2 and received on the PC at t_2. Assuming that the transport time α is symmetrical, one has

$$\begin{cases} \alpha = 0.5[(t_2 - t_1) - (s_2 - s_1)] \\ \beta = s_1 - (t_1 + \alpha) \end{cases}$$

with β the time offset defined by $t_2 = s_2 + \beta$. In practice, since there is some jitter on the serial link, we use a low-pass filter to smooth the transport delay β.

C. Control Implementation

This subsection discusses the implementation of the control scheme presented in Section III on the flybox platform. The most time-critical parts of the control scheme, the torque and thrust vector control are implemented in fixed-point ANSI-C on the LL autopilot. The higher-level position- and velocity controllers are implemented on the HL computer as a ROS node. Using this approach, all higher-level functionality such as path planning or obstacle avoidance can be implemented as ROS nodes on the HL computer. Currently, the controller node receives position and velocity data from the Vicon node as well as position, velocity and acceleration commands from the trajectory generator (see Section IV-C6). Due to the similarity of the position and velocity controllers and the length limitation of the paper, the discussion of the velocity controller is omitted.

Even though we focus on the design and implementation of the controller in this paper, we quickly outline the attitude observer first.

1) Attitude Observer: Since the flybox platform is inherently unstable, it is of utmost importance to have a high bandwidth, good quality estimate of the vehicle's attitude. On the flybox platform, a highly optimized fixed-point implementation of the nonlinear attitude observer by Hamel *et al.* [9] is used. The observer runs at 1 kHz, which corresponds to four times the frequency of the attitude controller. The original attitude observer is based on the assumption that the accelerometer can be directly used as an inclinometer. Clearly, this assumption does not hold under dynamic flight conditions when linear accelerations are apparent. Therefore, we introduce a variable gain for the accelerometer correction which is based on the norm of the accelerometer measurement vector. Only if the vector is close to the gravity for a certain amount of time, we use the vector for attitude correction. Therefore, during temporal high-acceleration flight phases, the attitude is propagated using only the gyrometer output while completely rejecting the accelerometer measurements. We are aware that the attitude could potentially diverge, i.e., when continuously flying in circles. However, in practice, the UAV is experiencing low-acceleration flight phases periodically, thus allowing the observer to correct the attitude and update the gyrometer biases.

2) Outer-Loop Position Control: The position controller is the outmost controller in the cascade control structure which generates a desired thrust vector given some trajectory in terms of position, velocity, and acceleration. The controller is running at 100 Hz as a ROS node and can be tuned in flight using the dynamic reconfigure feature of ROS from the ground station over WiFi.

In view of (17), we can consider the desired thrust orientation $\boldsymbol{\eta}_d$ as the desired acceleration $\mathbf{a}_d$. This has the advantage that we can use the same position controller on a large variety of platforms, irrespective of their mass. For instance, for illustration purposes let us consider a near hovering flight in the absence of wind and neglecting blade flapping such that $\boldsymbol{\gamma}_e \approx g\mathbf{e}_3$. Now, we can write the desired accelerations as

$$\begin{cases} \mathbf{a}_{1,2} = \mathrm{sat}_{\Delta_1^p}(k_1^p \tilde{\boldsymbol{\xi}}_{z,1,2}) + \mathrm{sat}_{\Delta_1^v}(k_1^v \dot{\tilde{\boldsymbol{\xi}}}_{z,1,2}) - \ddot{\boldsymbol{\xi}}_{r,1,2} + \ddot{\mathbf{z}}_{1,2} \\ a_3 = \mathrm{sat}_{\Delta_3^p}(k_3^p \tilde{\xi}_{z,3}) + \mathrm{sat}_{\Delta_3^v}(k_3^v \dot{\tilde{\xi}}_{z,3}) - \ddot{\xi}_{r,3} + \ddot{z}_3 + g. \end{cases}$$

In the traditional literature on control of small-scale UAVs, the torque and forces due to flapping and induced drag are neglected and only the gravity is incorporated as an external force in the model. While there exist techniques for estimating the external acceleration $\boldsymbol{\gamma}_e$ [16], [12, Ch.2] we assume that the aerodynamic drag forces caused by parasitic-, flapping-, or induced drag are relatively small compared to the gravity. We assume that a high-gain controller is able to dominate all these external forces. This assumption is backed by experiments in which the position control error of a small-scale UAV is in the range of a few millimeter even in highly dynamic manoeuver while using a high-gain controller that does not take those effects into account [26]. As a consequence, we can set the external acceleration to the gravity and compensate the remaining forces via the use of the position integrator (16).

The choice of the saturation value Δ_3^p and Δ_3^v involved in the function $h(\cdot)$ defined by (15) is based on the fact that the desired thrust vector should not cross zero. Therefore, the following equation has to be met

$$-\Delta_3^p - \Delta_3^v - \ddot{\xi}_3^{\max} - \ddot{z}_3^{\max} + g = a_3^{\min}$$

with some positive number $a_3^{\min} < g$. The choice of the saturation value Δ_1^p and Δ_1^v involved in the function $h(\cdot)$ defined by (15) can be based on the maximal expected tilting angle of the UAV. The maximal tilting angle $\Theta_{\max}$is directly linked to the maximal lateral acceleration by $\Theta_{\max} = \arctan(a_{1,2}^{\max}/a_3^{\min})$ with $a_{1,2}^{\max} = \Delta_1^p + \Delta_1^v + \ddot{\xi}_{r,1,2}^{\max} + \ddot{z}_1^{\max}$. The value of $\ddot{z}_i^{\max}$ $(i = 1, 2, 3)$ is set when tuning the integrator and $\ddot{\xi}_{r,i}^{\max}$ $(i = 1, 2, 3)$ is a design variable of the trajectory generator node (see Section IV-C6). Using the remaining degree of freedom, the control designer can then adjust the control authority of the P- and D- parts of the controller.

For completeness, let us provide a more explicit expression of $\dot{\boldsymbol{\eta}}_d$. Since $\boldsymbol{\eta}_d = \boldsymbol{\gamma}_z^p/|\boldsymbol{\gamma}_z^p|$ with $\boldsymbol{\gamma}_z^p$ defined by (17), one has (see [14]) $\dot{\boldsymbol{\eta}}_d = -\frac{1}{|\boldsymbol{\gamma}_z^p|}(\boldsymbol{\eta}_{d\times})^2\dot{\boldsymbol{\gamma}}_z^p$ and

$$\dot{\boldsymbol{\gamma}}_z^p = \frac{\partial h(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z)}{\partial \tilde{\boldsymbol{\xi}}_z}\dot{\tilde{\boldsymbol{\xi}}}_z + \frac{\partial h(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z)}{\partial \dot{\tilde{\boldsymbol{\xi}}}_z}\ddot{\tilde{\boldsymbol{\xi}}}_z + \dot{\boldsymbol{\gamma}}_e - \boldsymbol{\xi}_r^{(3)} + \mathbf{z}^{(3)}.$$

The term $\frac{\partial h(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z)}{\partial \tilde{\boldsymbol{\xi}}_z}$, $\frac{\partial h(\tilde{\boldsymbol{\xi}}_z, \dot{\tilde{\boldsymbol{\xi}}}_z)}{\partial \dot{\tilde{\boldsymbol{\xi}}}_z}$, and $\mathbf{z}^{(3)}$ involve the derivative of the classical saturation function $\mathrm{sat}_\Delta(\cdot)$ which is not differentiable at points with norm equal to Δ. One can easily replace this saturation function by a smooth approximation like some $\tanh(\cdot)$ function for the continuity of the controller (see, e.g., [14] for some solutions). However, the fact that $\mathrm{sat}_\Delta(\cdot)$ is right-differentiable along any smooth curve is sufficient for stability analysis (see [14]). Finally, once the unit desired thrust vector $\boldsymbol{\eta}_d$ and its time derivative $\dot{\boldsymbol{\eta}}_d$ are computed from the position outer-loop control, they are sent to the LL autopilot together with the desired heading vector.

3) Inner-Loop Thrust-Direction Control: The thrust-direction controller runs at 250 Hz on the LL autopilot. We consider that the UAV is used in an inspection setting so that we can assume that the reference trajectory $\boldsymbol{\xi}_r$ is not very aggressive, i.e., its high-order time-derivatives are negligible like in the case of waypoint navigation. As a consequence, in our implementation, we can neglect the term involving $\dot{\boldsymbol{\eta}}_d$ in the control expression (8). However, for the sake of completeness, we describe the implementation of the thrust direction controller including $\dot{\boldsymbol{\eta}}_d$.

We now simplify (8) for the efficient computation of $\boldsymbol{\omega}_d$. First, to avoid singularity occurred when $\boldsymbol{\eta} = -\boldsymbol{\eta}_d$, we replace the first term in the RHS of (8) by $\frac{k_\eta \mathbf{R}^\top(\boldsymbol{\eta}\times\boldsymbol{\eta}_d)}{\max(\epsilon,(1+\boldsymbol{\eta}^\top\boldsymbol{\eta}_d)^2)}$ with some security threshold $\epsilon > 0$. Keeping in mind that $\mathbf{R}^\top\boldsymbol{\eta} = \mathbf{e}_3$, this term can be rewritten as

$$\frac{k_\eta \mathbf{R}^\top(\boldsymbol{\eta} \times \boldsymbol{\eta}_d)}{\max(\epsilon, (1 + \boldsymbol{\eta}^\top\boldsymbol{\eta}_d)^2)} = \frac{k_\eta\,[-r_2,\, r_1,\, 0]^\top}{\max(\epsilon, (1 + r_3)^2)} \tag{23}$$

with $\mathbf{R}^\top\boldsymbol{\eta}_d = [r_1,\, r_2,\, r_3]^\top \in \mathbb{R}^3$ with unit norm and the second term in the RHS of (8) can be rewritten as

$$\mathbf{R}^\top(\boldsymbol{\eta}_\times)^2(\boldsymbol{\eta}_d \times \dot{\boldsymbol{\eta}}_d) = [-r_2 s_3 + r_3 s_2, -r_3 s_1 + r_1 s_3, 0]^\top \tag{24}$$

with $\mathbf{R}^\top\dot{\boldsymbol{\eta}}_d = [s_1,\, s_2,\, s_3]^\top \in \mathbb{R}^3$. Finally, the algorithm to compute $\boldsymbol{\omega}_{d,1,2}$ boils down to a few following simple steps:

1) rotate the vectors $\boldsymbol{\eta}_d$ and $\dot{\boldsymbol{\eta}}_d$ from the world frame into the body frame using the quaternion from the attitude estimator. This can be done efficiently in fixed-point [13];
2) compute the desired angular rates $\boldsymbol{\omega}_{d,1,2}$ using (23) and (24);
3) Saturate the desired body angular rates at 75% of the gyrometer range to prevent gyrometer saturation.

The computation of the desired angular rates $\boldsymbol{\omega}_{d,1,2}$ in fixed point is particularly easy and stable since we know the length of $\boldsymbol{\eta}_d$ already at compile time which is always equal to one.

4) Yaw Control: As in the case of the thrust vector controller, the yaw controller runs at 250 Hz. The yaw angle can be extracted from the quaternion using the $\arctan 2$ function. Since memory is not an issue on the Cortex M3 microcontroller, the atan2 can be implemented efficiently using a look-up-table (see [13] for details). Using a simple P-controller, we can compute the desired angular velocity as $\omega_{d,3} = k_\psi(\psi_d - \psi)$. Similar to the case of the thrust orientation controller, we saturate the $\omega_{d,3}$ to 75% of the gyro range. It has to be noted that for many applications, it is enough to control the yaw rate. In this case, the user can directly specify $\omega_{d,3}$, and thus, avoiding the extraction of the yaw angle from the quaternion.

5) Rate Control: The rate control allows the vehicle to track the desired angular velocities coming from the thrust vector and yaw controllers. It is at the core of the cascade control strategy. In order to ensure good tracking performance, it runs at 1 kHz. For simplicity, we neglect the feedforward term $\mathbf{J}\dot{\boldsymbol{\omega}}_d$ in the torque control expression (7). In the case of unknown inertia of the vehicle, the term $\boldsymbol{\omega}_d \times \mathbf{J}\boldsymbol{\omega}$ can also be neglected and the gain matrix $\mathbf{K}_\omega$ can be increased instead. In terms of implementation in fixed point, it has to be noted that $\boldsymbol{\omega}_d$ and $\boldsymbol{\omega}$ are bounded. Therefore, the maximal value for the torque is known and the fixed-point precision can be set at compile time to efficiently prevent buffer overruns.

6) Trajectory Generation: When the UAV is operated by a human, the trajectory may not be known *a priori*, therefore, the velocity and acceleration setpoints of the UAV have to be estimated. By assuming that the human operator provides a position input, we can design an observer

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathrm{sat}_{v_{\max}}(\mathbf{x}_2) \\ \dot{\mathbf{x}}_2 = \mathrm{sat}_{a_{\max}}(-k_d \mathbf{x}_2 - k_p(\mathbf{x}_1 - \mathbf{u})) \end{cases} \tag{25}$$

with $\mathbf{u}$ the input specified by the operator, and $\mathbf{x}_1, \mathbf{x}_2$, and $\dot{\mathbf{x}}_2$ the desired position, velocity, and acceleration setpoints, respectively, for the position controller. Using the saturation functions, it is easy to define maximal desired accelerations and velocities. Additionally, the aggressiveness of the trajectory generator can be tuned by adjusting the gains k_p and k_d.

V. Experimental Results

The performance of the proposed hierarchical position control scheme on the flybox is evaluated via several experiments performed indoors in a Vicon motion tracking system. The Vicon system provides millimeter-accuracy position and subdegree accuracy attitude ground truth at 100 Hz. While the UAV is designed to be used in an industrial setting where no Vicon system is available, we use the tracking system to validate the control algorithms in the majority of experiments. The reasoning is that the Vicon tracker provides repeatable data with well-known accuracy and delay for the experiments, so other researchers can reproduce the experiments. To show the performance of the UAV in an industrial environment, we also perform a wall-following experiment where only onboard sensors are used to control the pose of the UAV, while using the Vicon system only for ground truth. In this experiment, a 2-D laser range scanner is used to estimate the distance and heading of the UAV w.r.t. wall and a pressure sensor, fused with an accelerometer, is used to estimate the altitude.

The reported experiments were filmed and are available in [5].

In general, we assume that the UAV is used in an industrial inspection setting. Therefore, we focus rather on quick disturbance rejection and robust performance than on aggressive position control when tuning the gains. The control gains have been determined via a pole placement procedure performed on the linearized system of system (5a)–system (16) at hovering with all aerodynamic forces being neglected. Details on the gain-tuning process can be found in [12, Ch.2].

The original attitude observer uses magnetometer measurements to correct for a drift in yaw. When flying indoors or close to structure, the measurement of the earth magnetic field is often perturbed by strong external magnetic fields. Therefore, we use the Vicon attitude to emulate a virtual magnetic vector in all Vicon-based experiments. In the wall-following experiment, we use the laser-scanner to emulate a virtual magnetic vector. To decouple correction using the accelerometer from Vicon correction, we generate a vector which is perpendicular to gravity (for more details, see [13]).

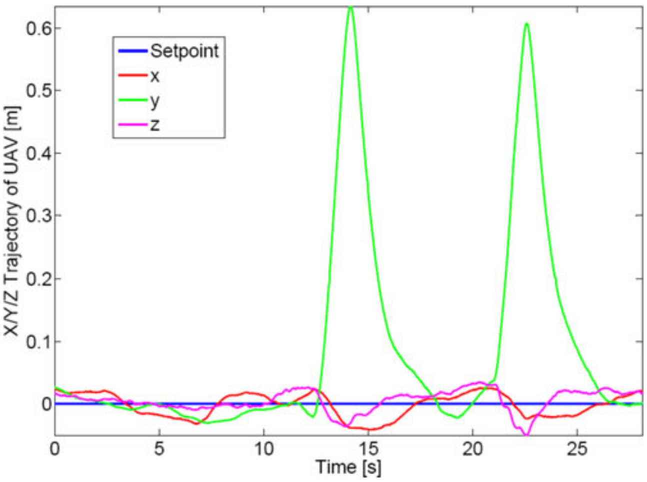


Fig. 7. Hovering performance of UAV. At $t = 10$ s and $t = 20$ s, the UAV is manually deviated from the setpoint by 60 cm.

A. Hovering Performance and Disturbance Rejection

In the first experiment, we show the performance of the position controller in the Vicon system while hovering, as depicted in Fig. 7. The root-mean-square (RMS) error of the position controller is 1.8 cm in *x*-, 1.6 cm in *y*-, and 0.6 cm in *z*-direction during hovering. At $t = 10$ s and $t = 20$ s, the flybox is manually deviated from the setpoint by about 60 cm. Evidently, the controller manages to bring the vehicle back to the setpoint with small overshoot in an acceptable time frame.

B. Wind Rejection

In a second experiment, the performance of the saturated integrators of the position controller is evaluated. We emulate an external force disturbance using a fan with a diameter of 0.75 m that generates a wind speed of 10 m/s. At $t = 10$ s, the fan is started, as depicted in Fig. 8. Immediately, the UAV starts drifting away until the controller manages to counteract the disturbance at $t = 15$ s. In total, the integrator manages to return the UAV to the setpoint within less than 10 s.

C. Square Tracking

In the third experiment, we evaluate the performance of the UAV when tracking a square trajectory with a side length of 1 m. In every corner, the UAV is commanded to come to a complete stop thus imposing large short-term accelerations on the UAV. Taking those requirements into account, the trajectory generator is outputting a position, velocity, and acceleration profile for the UAV to track. Maximal translational velocity and acceleration in this trajectory are 1 m/s and 1.7 m/s^2, respectively. The UAV manages to track the square trajectory twice within less than 20 s. The RMS trajectory tracking error is 7.4 cm in *x*-, 8.3 cm in

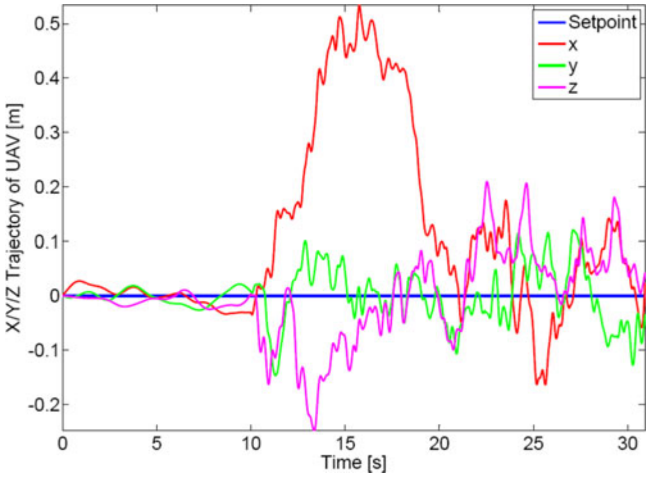


Fig. 8. Performance of the UAV when flying in wind at 10 m/s. The wind source is acting on the UAV after $t = 10$ s. The integrator manages to bring back the UAV to the setpoint within 10 s.

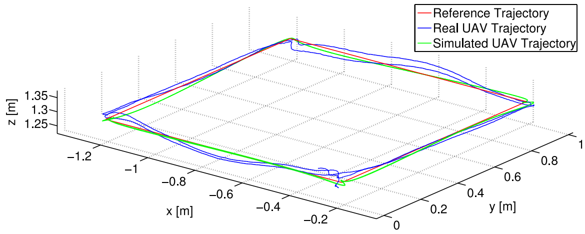


Fig. 9. UAV tracks the square trajectory with 1 m side length twice within less than 20 s. In every corner of the square, the UAV stops completely. Maximal reference translational velocity and acceleration are 1 m/s and 1.7 m/s^2, respectively. Please note the different scale for the vertical plot axis.

y- and 1.5 cm in z-direction.[1] As depicted in Fig. 9, the tracking error evolves similarly during each iteration of the square.

To investigate the causes for the tracking error, the same experiment using the same control gains is performed again in simulation. To this end, we assume that all parameters of the UAV (such as mass, inertia, center of gravity, thrust- and torque coefficient, etc.) are known, the noise level in position and velocity measurements are comparable to the Vicon system and that we do not have any aerodynamic disturbances. The effects preventing perfect trajectory tracking in simulation are 1) the lowpass behavior of the motor (rotor speed is assumed to follow the dynamics of a first-order lowpass with a time constant of 0.05 s), and 2) the position and velocity measurement noise. As a consequence, the difference in performance between the real and simulated experiments are due to the imperfect knowledge of the system parameters and the aerodynamic disturbances. The RMS trajectory tracking error in simulation is 4.1 cm in x-, 3.9 cm in y-, and 1.6 cm in z-direction. The vertical control error remains approximately the same in simulation compared to the real experiment. If the motor dynamics are ignored, this error reduces significantly. This can be explained by the fact that the thrust magnitude is not roll- and pitch compensated. Due to the attitude tracking error caused by the motors dynamics, there is a mismatch in the required thrust magnitude which results in vertical control error.

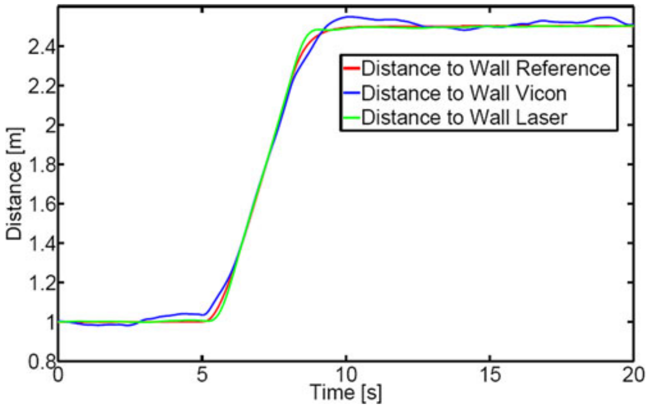


Fig. 10. Performance of the laser-based wall following algorithm. At $t = 5$ s, the distance of the UAV to the wall is changed from $d = 1$ m to $d = 2.5$ m at 0.5 m/s.

The horizontal control errors are reduced by approximately a factor 2. Monte Carlo simulations also showed that the horizontal translational tracking error is highly sensitive to a shift in CoM. Moving the CoM by 3 cm causes an error comparable to the one in the real experiment. Several error sources due to an unknown CoM are not slowly time varying and can, therefore, not be compensated by the integrator of the translational controller. For one, the unknown CoM affects the control allocation matrix. Secondly, an unknown CoM causes the rotational dynamics to affect the estimation of the translational dynamics since there is an unknown position offset of the position sensor (e.g., laser scanner or Vicon marker) with respect to the CoM.

Additionally, simulation also showed that performance of the position controller is limited by the delay of the position signal. With a delay of 25 ms, as in the case of the Vicon signal that is transmitted over the 2.4-GHz WiFi link, the control performance already deteriorates significantly.

D. Laser-Based Wall Following

In a fourth experiment, we evaluate the performance of the position controller when employed in a real-life inspection scenario with no Vicon motion tracking system available. In this experiment, a wall-following procedure is performed using a 2-D laser range scanner (Hokuyo UTM-30LX) for lateral control and a pressure sensor for altitude control. More precisely, the laser range scanner is used to estimate the heading and the orthogonal distance of the UAV to the wall. The pressure sensor, fused with the onboard accelerometer, is used to measure the altitude of the UAV. The remaining degree of freedom, the movement laterally along the wall, is controlled in open loop, where the human operator directly sets the desired tilting angle of the UAV in direction along the wall. In this experiment, the Vicon is only used to collect the ground truth. As depicted in Fig. 10, the distance estimate using the laser is close to the Vicon ground truth. Consequently, the distance error of the UAV

[1] The trajectory tracking error is defined as the distance of the current position of the UAV to the setpoint at the same time. Often in literature, only the error in normal distance to the trajectory is given. This is misleading as this does not take into account the dynamic properties of the trajectory tracking.

to the wall is comparable to the previous, Vicon-based experiments (RMS error 2.3 cm). This can be explained by the fact that the extracted distance (and its estimated derivative) has approximately the same noise and bandwidth characteristics than the Vicon system.

For space reasons, the plot of the altitude controller is omitted. However, the interested reader can see the performance of the controller in the video available at [5].

VI. Conclusion

In this paper, a generic dynamic model of multirotor vehicles is provided and tailored to the flybox hexacopter. The modeling takes into account the first order drag forces due to blade flapping and induced drag. The flight controller presented in this study can be applied to any type of VTOL vehicle. It is based on a hierarchical nonlinear control structure. It uses saturated integrators with fast desaturation properties. This renders the control system more robust to modeling uncertainties and external perturbations. The vehicle's thrust magnitude and orientation are used as intermediary control variables to control the translational and rotational dynamics. As a consequence, the UAV can be considered as a fully actuated point mass with force control inputs when designing the position or velocity controller. This paper explains how the controllers are implemented in practice, and what are the necessary hardware and software components to successfully perform autonomous flight. The experimental results validate the entire design: hardware, software, and control approach.

Acknowledgment

The authors would like to thank P. Strupler, CEO of Skybotix, for his support.

References

[1] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *Proc. IEEE Conf. Robot. Autom.*, 2004, pp. 4393–4398.

[2] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit, "The navigation and control technology inside the AR.Drone micro UAV," in *Proc. IFAC World Congr.*, 2011, pp. 1477–1484.

[3] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE Control Syst. Mag.*, vol. 25, no. 6, pp. 45–55, Dec. 2005.

[4] X. Ding and Y. Yu, "Motion planning and stabilization control of a multipropeller multifunction aerial robot," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 645–656, Apr. 2013.

[5] G. Ducard. (2012, Aug.). I3S UAV Video Page, [Online]. Available: goo.gl/btkth

[6] G. Ducard and M.-D. Hua, "Discussion and practical aspects on control allocation for a multi-rotor helicopter," in *Proc. Conf. Unmanned Aerial Veh. Geomat.*, 2011, pp. 1–6.

[7] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 331–340, Apr. 2008.

[8] D. Gurdan, J. Stumpf, M. Achtelik, K. M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1kHz," in *Proc. IEEE Conf. Robot. Autom.*, 2007, pp. 361–366.

[9] T. Hamel and R. Mahony, "Attitude estimation on SO(3) based on direct inertial measurements," in *Proc. IEEE Conf. Robot. Autom.*, 2006, pp. 2170–2175.

[10] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an X4-flyer," in *Proc. IFAC World Congr.*, 2002, pp. 200–212.

[11] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. AIAA Guid., Navigat. Control Conf. Exhibit.*, 2007, no. 2007–6461.

[12] M.-D. Hua, "Contributions to the automatic control of aerial vehicles," Ph.D. dissertation, Univ. Nice-Sophia Antipolis, Nice, France, 2009.

[13] M.-D. Hua, G. Ducard, T. Hamel, R. Mahony, and K. Rudin, "Implementation of a nonlinear attitude estimator for aerial robotic vehicles," *IEEE Trans. Control Syst. Technol.*, DOI: 10.1109/TCST.2013.2251635, 2013.

[14] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1837–1853, Aug. 2009.

[15] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles," *IEEE Control Syst. Mag.*, vol. 33, no. 1, pp. 61–75, Feb. 2012.

[16] M.-D. Hua, P. Morin, and C. Samson, "Balanced-force-control of underactuated thrust-propelled vehicles," in *Proc. IEEE Conf. Decis. Control*, 2007, pp. 6435–6441.

[17] M.-D. Hua and C. Samson, "Time sub-optimal nonlinear PI and PID controllers applied to longitudinal headway car control," *Int. J. Control*, vol. 84, no. 10, pp. 1717–1728, 2011.

[18] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in agressive maneuvering," in *Proc. IEEE Conf. Robot. Autom.*, 2009, pp. 3277–3282.

[19] J. S. Jang and C. Tomlin, "Longitudinal stability augmentation system design for the DragonFly UAV using a single GPS receiver," presented at the AIAA Guid., Navigat, Control Conf., Austin, TX, USA, 2003, Paper 2003-5592.

[20] S. Lupashin, A. Schoellig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Proc. IEEE Conf. Robot. Autom.*, 2010, pp. 1642–1648.

[21] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008.

[22] R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.

[23] P. Martin and E. Salaun, "Design and implementation of a low-cost observer-based attitude and heading reference system," *Control Eng. Pract.*, vol. 18, pp. 712–722, 2010.

[24] P. Martin and E. Salaun, "Hardware and software architecture for state estimation on an experimental low-cost small-scaled helicopter," *Control Eng. Pract.*, vol. 18, pp. 733–746, 2010.

[25] P. Martin and E. Salaun, "The true role of accelerometer feedback in quadrotor control," in *Proc. IEEE Conf. Robot. Autom.*, 2010, pp. 1623–1629.

[26] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 664–674, 2012.

[27] 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 Eng. Pract.*, vol. 18, pp. 747–760, 2010.

[28] J.-M. Pflimlin, T. Hamel, P. Souères, and R. Mahony, "A hierarchical control strategy for the autonomous navigation of a ducted fan flying robot," in *Proc. IEEE Conf. Robot. Autom.*, 2006, pp. 2491–2496.

[29] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Eng. Pract.*, vol. 18, no. 7, pp. 691–699, 2010.

[30] R. W. Prouty, *Helicopter Performance, Stability, and Control.* Melbourne, FL, USA: Krieger, 2002.

[31] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: An open-source robot operating system," presented at the ICRA Workshop Open Source Software, Kobe, Japan, 2009.

[32] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Path tracking of a UAV via an underactuated H_∞ control strategy," *Eur. J. Control*, vol. 17, no. 2, pp. 194–213, 2011.

[33] K. Rudin, M.-D. Hua, G. Ducard, and S. Bouabdallah, "A robust attitude controller and its application to quadrotor helicopters," in *Proc. IFAC World Congr.*, 2011, pp. 10379–10384.

[34] S. Saripalli, J. M. Roberts, P. I. Corke, G. Buskey, and G. S. Sukhatme, "A tale of two helicopters," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2003, pp. 805–810.

[35] T. Schneider and G. Ducard, "Fault-tolerant control allocation for multirotor helicopters using parametric programming," presented at the Int. Micro Air Vehicle Conf., Braunschweig, Germany, 2012.

[36] A. Schoellig, F. Augugliaro, S. Lupashin, and R. D'Andrea, "Synchronizing the motion of a quadrocopter to music," in *Proc. IEEE Conf. Robot. Autom.*, 2010, pp. 3355–3360.

[37] S. Seshagiri and H. Khalil, "Robust output feed back regulation of minimum-phase nonlinear systems using conditional integrators," *Automatica*, vol. 41, pp. 43–54, 2005.

[38] O. Shakernia, Y. Ma, T. Koo, and S. Sastry, "Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control," *Asian J. Control*, vol. 1, no. 3, pp. 128–145, 1999.

[39] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3D exploration with a micro-air vehicle," in *Proc. IEEE Conf. Robot. Autom.*, 2012, pp. 9–15.

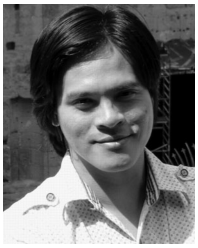
[40] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 3, pp. 562–571, May 2006.

[41] S. Weiss, M. Achtelik, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in *Proc. IEEE Conf. Robot. Autom.*, 2012, pp. 957–964.

[42] J. Wendel, O. Meister, C. Schlaile, and G. F. Trommer, "An integrated gps/mems-imu navigation system for an autonomous helicopter," *Aerosp. Sci. Technol.*, vol. 10, no. 6, pp. 527–533, 2006.

Sammy Omari received the master's degree in mechanical engineering from ETH Zurich, Zurich, Switzerland, in 2012, where he is currently working toward the Ph.D. degree with a thesis concerning intelligent control and teleoperation of unmanned aerial vehicles in the Autonomous System Laboratory. During his master's degree work he was involved in the development of the flybox platform.

Mr. Omari received the deVigier Young Entrepreneur Award for his involvement with the company Skybotix.



Minh-Duc Hua received the Eng. degree from the Ecole Polytechnique, Palaiseau, France, in 2006, and the Ph.D. degree from the University of Nice–Sophia Antipolis (UNS), Nice, France, in 2009.

He spent two years as a Postdoctoral Researcher at the I3S UNS-Centre National de la Recherche Scientifique (CNRS). He is currently a Researcher (Chargé de Recherche) of the CNRS at the laboratory ISIR UPMC-CNRS, Paris, France. His research interests include nonlinear control theory and estimation and teleoperation with applications to autonomous mobile robots such as UAVs.

Guillaume Ducard (M'07) received the master's degree in electrical engineering from ETH Zurich, Zurich, Switzerland, in 2004, where he also completed his Doctoral work in 2007 and his two-year Postdoctoral work in 2009 in the field of unmanned aerial vehicles.

He is currently conducting research at the I3S UNS-CNRS Sophia Antipolis, France, working on nonlinear control and estimation, fault-tolerant flight control and guidance systems for UAVs, and formation flight control.

Tarek Hamel (M'05) conducted his Ph.D. research at the University of Technologie of Compiègne (UTC), Compiègne, France, and received the Doctorate degree in robotics in 1996.

He has been a Professor at the University of Nice Sophia Antipolis, Nice, France, since 2003. After two years as a Research Assistant at the UTC, he joined the Centre d'Etudes de Mcanique d'Iles de France in 1997 as an Associate Professor. His research interests include nonlinear control theory and estimation and vision-based control with applications to unmanned aerial vehicles.

Dr. Hamel is currently an Associate Editor for the IEEE Transactions on Robotics and for *Control Engineering Practice*.