J Intell Robot Syst (2015) 77:95–111
DOI 10.1007/s10846-014-0096-8

Model-Free Control of a Hovering Flapping-Wing Microrobot

The Design Process of a Stabilizing Multiple-Input–Multiple-Output Controller

Néstor O. Pérez-Arancibia · Pierre-Emile J. Duhamel · Kevin Y. Ma · Robert J. Wood

Received: 27 January 2014 / Accepted: 8 August 2014 / Published online: 2 October 2014

Abstract We present a model-free experimental method to find a control strategy for achieving stable flight of a dual-actuator biologically inspired flapping wing flying microrobot during hovering. The main idea proposed in this work is the sequential tuning of parameters for an increasingly more complex strategy in order to sequentially accomplish more complex tasks: upright stable flight, straight vertical flight, and stable hovering with altitude and position control. Each term of the resulting *multiple-input–multiple-output* (MIMO) controller has a physical intuitive meaning and the control structure is relatively simple such that it could potentially be applied to other kinds of flapping-wing robots.

Keywords Microrobotics · Flapping-wing flight · Real-time control · Experimental robotics

1 Introduction

Experiments demonstrating the first controlled vertical unconstrained flight of a 83-mg flapping-wing flying microrobot were presented in [1]. There, the idea of using separate actuators exclusively for control was introduced and demonstrated, through static and flying experiments. The argument for designing, developing, and integrating separate actuators exclusively for control is biologically inspired, based on evidence suggesting that natural insects evolved separate specialized muscles for power and control [2]. There are important practical problems that arise in the fabrication process developed for materializing the design in [1]. Specifically, fabrication needs to be essentially perfect in order to avoid asymmetries

N. O. Pérez-Arancibia and P.-E. J. Duhamel contributed equally to this work.

This work was supported in part by the National Science Foundation (award number CCF-0926148) and the Wyss Institute for Biologically Inspired Engineering. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Electronic supplementary material The online version of this article (doi:10.1007/s10846-014-0096-8) contains supplementary material, which is available to authorized users.

N. O. Pérez-Arancibia (✉)
Department of Aerospace and Mechanical Engineering,
University of Southern California, Los Angeles,
CA 90089-1453, USA
e-mail: perezara@usc.edu

P.-E. J. Duhamel · K. Y. Ma · R. J. Wood
School of Engineering and Applied Sciences, Harvard
University, Cambridge, MA 02138, USA

P.-E. J. Duhamel
e-mail: pduhamel@fas.harvard.edu

K. Y. Ma
e-mail: kevinma@seas.harvard.edu

R. J. Wood
e-mail: rjwood@eecs.harvard.edu

P.-E. J. Duhamel · K. Y. Ma · R. J. Wood
Wyss Institute for Biologically Inspired Engineering,
Harvard University, Boston, MA 02115, USA

in the prototypes that would make them very difficult to stabilize and control, or uncontrollable. Despite these fabrication challenges, unconstrained flight control of the prototype in [1] was preliminarily, but convincingly, demonstrated using the ideas and findings on altitude control and pitch control in [3–5], and references therein.

A different design approach, first proposed in [6], was fully developed in the work presented in [7], which is the basic design of the robotic prototype considered in this paper. This design consists of dual completely independent power actuators that drive each of the wings independently through two separate transmissions, and departs significantly from the previous purely biologically inspired robotic models in [1]. A model-based control strategy for the prototype in [7] was proposed and tested in the work presented in [8]. That paper describes a controller structure consisting of three independent modules for controlling body attitude, lateral position, and altitude. Therein, it is also explained that the stability of the independent attitude closed-loop is ensured by deriving the control law from a Lyapunov function. However, stability robustness and performance robustness with respect to model uncertainty are not analyzed or discussed, and consequently, not explicitly enforced by well-defined theoretical or empirical conditions. Furthermore, it is not clear how the control signals from the different modules interfere with each other and the actuators are finally excited. Here, we propose a new control strategy, which is direct, entirely experimental and model-free, where stability, performance, stability robustness and performance robustness are enforced empirically by conducting a series of real-time flying experiments. Also, the proposed *multiple-input–multiple-output* (MIMO) controller, once tuned, is designed to deal with the undesired effects produced by the cross-coupling between the channels forming the open-loop MIMO plant. The controller design method and overall control strategy described in this paper follow from, and take advantage of, the knowledge on flapping-wing systems gathered through the static and flying experiments discussed in [3–5] and the associated analyses presented therein.

In this paper, we provide evidence that the control philosophy first proposed in [1, 3], based on asymmetrical flapping patterns, is applicable to the general flapping-wing flight control case. The experimental results presented here are significantly better than those presented in [1], mainly because the design considered in this work (in Figs. 1 and 2) is, from a practical perspective, more controllable and robust to fabrication errors. The main new idea explored in this paper is the feasibility of finding a MIMO controller through a sequence of three set of tuning experiments, in which in every step, the complexity of the controller structure is increased in order to accomplish more sophisticated control objectives, while the microrobot is in unconstrained flight. In the first set of experiments, we tune a MIMO scheme that directly filters the pitch and roll angles of the robot. Simultaneously, we add to the control scheme a term that filters the velocities of the robot during flight along the pitch and roll axes, which makes the resulting controller robustly stable. In the second set of experiments, we demonstrate that pitch and roll can be actuated in order to control the robot's position on the horizontal plane. Clearly, the same method can be used to correct for drift, which allows the robot to perform straight vertical flight. Finally, in the last set of experiments, we close the altitude control loop in order to make the robot hover, using the controller scheme first proposed in [4].

Researchers have identified four basic flight modes observed in nature: *gliding*, *fast forward flapping*

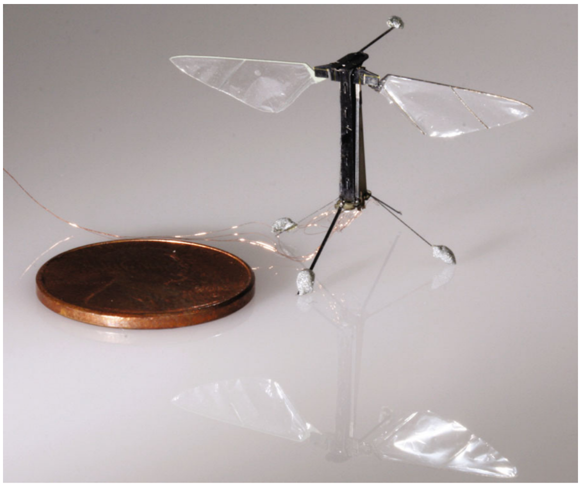

Fig. 1 Photograph of one of the flapping-wing flying microrobots used in the work presented in this paper. The prototypes considered in this work have two completely independent actuators used simultaneously for power and control. The ball-shaped markers glued to the robot are used by a Vicon motion capture system in the estimation of position and orientation of the robot. This robotic prototype was entirely developed and fabricated at the Harvard Microrobotics Laboratory

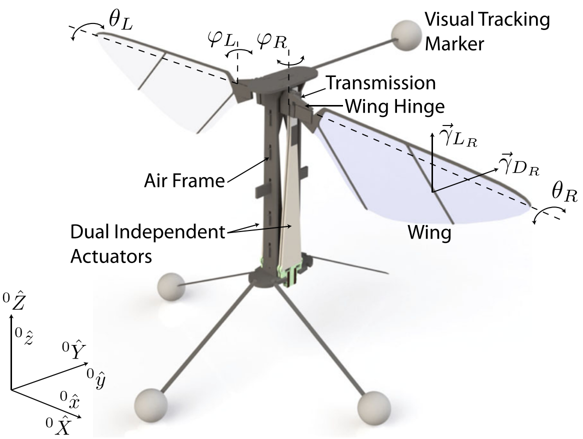

Fig. 2 Illustration of the robotic insect prototype employed in the flight experiments presented in this paper. The direction of the inertial frame axes are labeled as $\left\{{}^0\hat{X},{}^0\hat{Y},{}^0\hat{Z}\right\}$ and the direction of the body frame axes are labeled as $\left\{{}^0\hat{x},{}^0\hat{y},{}^0\hat{z}\right\}$. The origin of the body frame coincides with the robot's center of mass. In this illustration, the origin of the body frame is displaced for the sake of clarity

(FFF) flight, *slow forward flapping* (SFF) flight, and *hovering*. Different species perform variants of these basic modes. For example, *Chlorostilbon aureoventris* hummingbirds and most bee species hover using horizontal stroke planes and symmetrical flapping patterns (normal hovering), so that, the downstroke and upstroke are essentially identical, generating a positive average vertical thrust to compensate gravity, over a stroke cycle [9]. On the other extreme, *Pieris brassicae* butterflies are capable of hovering by flapping their wings in a vertical stroke plane. In between both extremes, bats such as the members of the species *Plecotus auritus* hover using an inclined stroke plane [10]. When the stroke plane is vertical or highly inclined, positive lift is generated during the downstroke and negative, zero or a small positive lift is generated during the upstroke. In all cases, hovering is achieved because the average lift force is positive even though if the instantaneous lift force oscillates from positive to negative values during a flapping cycle. For the kind of robot considered in this paper, from an aerodynamical perspective, the default flight mode is normal hovering, as defined in [9]. When this flight mode is used, the stroke plane is horizontal, so that, thrust is vertically generated in order to directly compensate the effect of gravity. Therefore, if robust stability for hovering is achieved, in theory, any point in space can be reached. This implies that, in essence, the general control problem concerning the robot has been solved.

The rest of the paper is organized as follows. Section 2 describes the flapping-wing microrobot used in the experiments and the main experimental setup. Section 3 describes the proposed model-free control strategy. Experimental results are presented in Section 4. Finally, some concluding remarks are given in Section 5.

Notation

- $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{R}^{++}$ denote the sets of reals, nonnegative reals and strictly positive reals, respectively.
- $\mathbb{Z}$, $\mathbb{Z}^+$ and $\mathbb{Z}^{++}$ denote the sets of integers, nonnegative integers and strictly positive integers, respectively.
- The variable t is used to index discrete time, i.e., $t = \{kT_s\}_{k=0}^{\infty}$, with $k \in \mathbb{Z}^+$ and $T_s \in \mathbb{R}^{++}$. T_s is referred to as the sample-and-hold time.
- The variable τ is used to index continuous time. Thus, for a generic continuous-time variable $x(\tau)$, $x(t)$ is the sampled version of $x(\tau)$, and vice versa.
- z^{-1} denotes the delay operator, i.e., for a signal x, $z^{-1}x(k) = x(k-1)$ and conversely $zx(k) = x(k+1)$. For convenience, z is also the complex variable associated to the z-transform.
- s^{-1} denotes the integrator operator and conversely s denotes the differential operator. For convenience, s is also the complex variable associated to the s-transform.
- A vector $\mathbf{v} \in \mathbb{R}^3$ is written with respect to the inertial orthogonal frame, $\left\{\hat{X}, \hat{Y}, \hat{Z}\right\}$, as ${}^0\mathbf{v}$, and with respect to the body orthogonal frame, $\left\{\hat{x}, \hat{y}, \hat{z}\right\}$, as ${}^b\mathbf{v}$. The unitary vector with the same direction as $\mathbf{v}$ is written as $\hat{v}$.

2 Description of the Microrobot and Main Experimental Setup

2.1 Force and Torque Generation

A photograph of one of the robots used in this work is shown in Fig. 1. An illustration of the same prototype, with the most relevant variables and components labeled, is shown in Fig. 2. This robotic design, first presented in [7], was entirely developed and fabricated at the Harvard Microrobotics Laboratory based on design principles and models that previously demonstrated the ability to liftoff [11] and fly under control in one degree of freedom [3, 4]. The default flight mode of the robot considered here (in Figs. 1 and 2) is the normal hovering mode, illustrated in Fig. 3, defined in [9] and extensively described in the literature [10, 12–14].

Natural flapping-wing flyers are very diverse in terms of size, shape and preferred flying strategies. Individual members of a given species might display a gamut of gaits, well adapted to different conditions, and therefore, it might be tempting to erroneously conclude that natural flying insects evolved to function in an optimal way, according to some figure of merit in order to increase their flying performance in terms of efficiency, maneuverability or other relevant system characteristic. In general, however, the evolved designs of flying insects are a compromise between several biological functions. For example, some elements in the complex flying behavior of honeybees might reflect an evolutionary adaptation to their complex social structure and not necessarily reflect advantages from an aerodynamics perspective [10]. Despite these biological complexities and subtleties, there exists evidence that suggests that there are strong reasons to use flapping-wing flight, normal hovering in particular, at small scale and intermediate Reynolds numbers ($10^2 \leqslant R_e \leqslant 10^4$) [15]. The most common reason is great maneuverability and agility, the most compelling is the integration of lift and thrust together with stability and control mechanisms. Thus, all forces on the surrounding fluid are derived from the motions of the same actuators, which has great implications in managing the final total vehicle weight. Furthermore, it has been argued that for low Reynolds numbers, flapping-wing flight and steady flight (fixed-wing flight) have similar aerodynamic power requirements [15], or even, for appropriate flight modes, flapping-wing flight can save aerodynamic power compared to steady flight [16].

The robot shown in Figs. 1 and 2 is composed of two independent identical piezoelectric bending bimorph cantilever actuators of the same kind in [11] and [17], which are used to provide mechanical power and for control simultaneously, two flexure-based transmissions (one for each side of the robot), a pair of airfoils, and an airframe, which serves as mechanical ground. Each transmission maps the approximately linear motions of its corresponding actuator into the flapping motion of the corresponding wing, labeled as $\varphi_R(\tau)$ and $\varphi_L(\tau)$ in Fig. 2 for the right and left wing, respectively. The transmissions consist of links and joints with geometries designed to maximize the product of stroke amplitude and first resonant frequency, given known actuator and airfoil properties [7].

The use of dual power actuators, as shown in Figs. 1 and 2, allows for multiple flapping combinations and strategies that are the subject of current and further research. In principle, it is not obvious what kinds of periodic flapping patterns are best in terms of energy efficiency, control effectiveness, and maneuverability. However, from what is observed in nature [2], it seems reasonable to use sinusoidal flapping patterns, though other options, such as split-cycle flapping, have been proposed in the literature [6]. Here, neglecting the

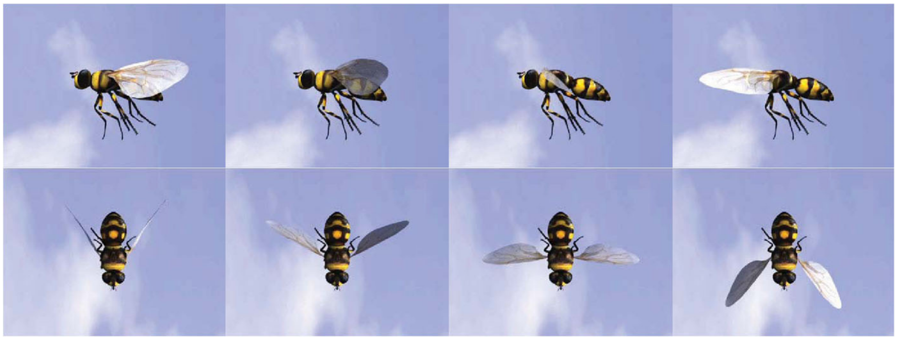

Fig. 3 Normal hovering is one of the flight modes observed in nature [9]. This flight mode is characterized because thrust is generated vertically in order to directly compensate gravity by flapping with a horizontal stroke plane, as depicted in the illustration. The default flight mode of the robot in Figs. 1 and 2 is normal hovering

nonlinearities observed in this kind of flapping system [18], the flapping patterns by design are chosen to be

$$\varphi_R(\tau) = \varphi_{AR}(\tau)\sin(2\pi f_R\tau) + \varphi_{BR}(\tau), \tag{1}$$

$$\varphi_L(\tau) = \varphi_{AL}(\tau)\sin(2\pi f_L\tau) + \varphi_{BL}(\tau), \tag{2}$$

for the right side and left side of the robot in Fig. 2, respectively. Notice that if

$$\varphi_{AR}(\tau) = \varphi_{AL}(\tau), \quad \text{with} \quad f_R = f_L \tag{3}$$

and

$$\varphi_{BR}(\tau) = \varphi_{BL}(\tau) = 0, \tag{4}$$

we obtain the symmetric flapping in Fig. 4a. For flapping systems actuated by piezoelectric actuators, the discrete-time identified mappings from the actuator excitation to the sampled flapping angle, are approximately *linear time-invariant* (LTI) [18], and therefore, Eqs. 1 and 2 are approximately achieved by exciting the right actuator and the left actuator respectively with

$$\nu_R(t) = \alpha_R(t)\sin(2\pi f_R t) + \beta_R(t), \tag{5}$$

$$\nu_L(t) = \alpha_L(t)\sin(2\pi f_L t) + \beta_L(t), \tag{6}$$

where $-1 \leqslant \nu_R(t) \leqslant 1$ and $-1 \leqslant \nu_L(t) \leqslant 1$. The excitation signals defined by Eqs. 5 and 6 are mapped to voltages with the same form over the range [0, 300] V, used to power the piezoelectric actuators of the robot, as previously described in [3–5, 18, 19], and references therein.

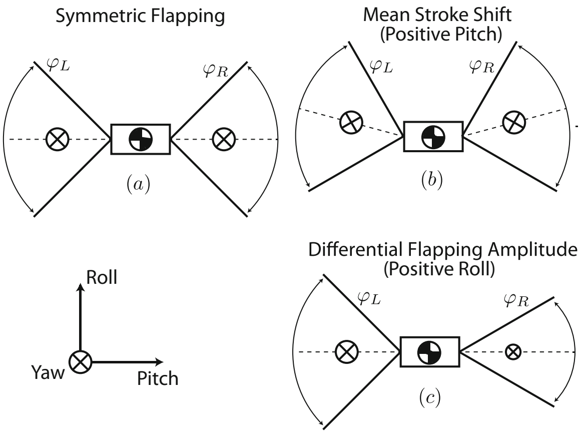

Fig. 4 (**a**) *Symmetric Flapping:* Assuming perfect fabrication, no body torques are generated and the angles of rotation in three dimensions about the robot's body axes, ${}^0\hat{x}$ (pitch), ${}^0\hat{y}$ (roll) and ${}^0\hat{z}$ (yaw), stay at 0°. (**b**) *Pitch Asymmetrical Flapping:* This flapping pattern makes the robot produce torques about the pitch axis. (**c**) *Roll Asymmetrical Flapping:* This flapping pattern makes the robot produce torques about the roll axis

Here, we employ two specific basic flapping modes that are combined in order to generate the body torques required for flight control. The first flapping mode, used to generate pitch torque, is shown in Fig. 4b, which is produced by exciting the actuators with

$$\alpha_R(t) = \alpha_L(t), \quad f_R = f_L, \quad \text{and} \quad \beta_R(t) = \beta_L(t) \neq 0. \tag{7}$$

In the particular case shown in Fig. 4b, $\beta_R(t) = \beta_L(t) > 0$, which produces a positive pitch torque by shifting the mean stroke, so that, $\varphi_{BR}(\tau) = \varphi_{BL}(\tau) > 0$. Similarly, $\beta_R(t) = \beta_L(t) < 0$ produces a negative pitch torque by shifting the mean stroke angle, so that, $\varphi_{BR}(\tau) = \varphi_{BL}(\tau) < 0$. The second flapping mode, used to generate roll torque, is shown in Fig. 4c, which is produced by exciting the actuators with

$$\alpha_R(t) \neq \alpha_L(t), \quad f_R = f_L, \quad \text{and} \quad \beta_R(t) = \beta_L(t) = 0. \tag{8}$$

In the particular case shown in Fig. 4c, $\alpha_R(t) < \alpha_L(t)$, which creates a positive roll torque by generating a larger average lift force on the left than the average lift force generated on the right. Similarly, $\alpha_R(t) > \alpha_L(t)$ produces a negative roll torque by generating a larger average lift force on the right than the average lift force on the left.

The production of forces by flapping wings is a very complex phenomenon to analyze in detail. However, there are some well established fundamental facts that allow us to understand how forces and body torques are generated and can be used to attain unconstrained controlled flight. An illustration of the robot's right wing depicting the generation of lift and drag forces during a flapping cycle is shown in Fig. 5. Here, a wing hinge connects the transmission to the wing, so that, when the wing is flapped according to a sinusoidal angle $\varphi_R(\tau)$, aerodynamic and inertial forces produce a passive rotation $\theta_R(\tau)$. The generation of $\theta_R(\tau)$ implies that the wing faces the air with an angle of attack necessary for the creation of time-varying lift and drag forces, the vectors $\boldsymbol{\gamma}_{L_R}(\tau)$ and $\boldsymbol{\gamma}_{D_R}(\tau)$. As is customary in the field of aerodynamics, the forces $\boldsymbol{\gamma}_{L_R}$ and $\boldsymbol{\gamma}_{D_R}$ are conceptual tools used to model a complex phenomenon in which the wing flapping produces a time-varying distribution of forces over the surfaces of the wing. In this case, the wing in Fig. 5 can be thought of as completely 2-D, which implies that the generation of forces can be treated as a vector field acting on a surface in space (the wing), which include aerodynamic and inertial contributions.

In principle, using basic vector calculus, it is possible to compute the time-varying total force exerted on the wing and the corresponding center of force exactly, from the force distribution. However, it is experimentally very challenging to design a sensor or devise an indirect method to measure the distribution of forces over the wing. A simplified approach is to measure the total forces along predetermined directions, as discussed in [4, 19, 20], and make some assumptions about the geometric location of the center of force. For a flapping system of the kind considered here, there exist accurate records of the forces generated along the $\hat{z}$-axis (lift forces) [4, 20] and preliminary measurements of the drag forces along the $\hat{\varphi}_R$ direction, as defined in Fig. 5 [21]. Also, from a simple analysis of the centrifugal forces acting on the wing, it is possible to determine that the forces along the $\hat{\rho}_R$ are negligible compared to the lift and drag forces acting on the wing. Thus, from the data presented in [4, 20] and [21], for the unbiased flapping $\varphi_R(\tau) = \varphi_{AR} \sin(2\pi f_R \tau)$, it is reasonable to model the sampled lift force as $\boldsymbol{\gamma}_{L_R}(t) = \gamma_{LR}(t)\hat{z}$, where $\gamma_{LR}(t)$ is a periodic signal with frequency $2f_R$ Hz with two positive maxima per cycle at $2\pi f_R t = \omega_R t = n\pi, n \in \mathbb{Z}^+$ and two negative minima per cycle at $2\pi f_R t = \omega_R t = \left(n + \frac{1}{2}\right)\pi$, $n \in \mathbb{Z}^+$. In steady state, for the case discussed here, the DC component of $\gamma_{L_R}(t)$, γ_{LR0}, is positive and can be estimated as

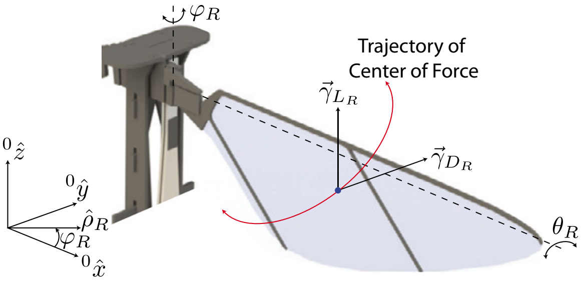

Fig. 5 Idealized illustration depicting the generation of lift and drag forces by a wing (the right wing in this case) during a single flapping cycle. A wing hinge connects the transmission to the wing, so that, when the wing is flapped according to a sinusoidal angle $\varphi_R(\tau)$, aerodynamic and inertial forces produce a passive rotation $\theta_R(\tau)$. The generation of $\theta_R(\tau)$ implies that the wing faces the air with an angle of attack necessary for creating time-varying lift and drag forces, the vectors $\boldsymbol{\gamma}_{L_R}(\tau)$ and $\boldsymbol{\gamma}_{D_R}(\tau)$. A simplified and idealized version of the center of force trajectory is depicted in red

$$\gamma_{LR0} \approx A_{\gamma_{LR}}(t) = \frac{1}{N_L}\sum_{i=0}^{N_L-1} \gamma_{LR}(t - T_s i) > 0, \quad (9)$$

where $t = kT_s$, with $k \in \mathbb{Z}$, $T_s \in \mathbb{R}$, and $N_L \in \mathbb{Z}^{++}$.

Similarly, the drag force can be modeled as $\boldsymbol{\gamma}_{D_R}(t) = \gamma_{DR}(t)\hat{\varphi}_R$, where $\gamma_{DR}(t)$ is a periodic signal with frequency f_R Hz with one positive maximum per cycle at $2\pi f_R t = \omega_R t = (2n+1)\pi$, $n \in \mathbb{Z}^+$. Here, the DC component of $\gamma_{DR}(t)$, γ_{DR0}, is theoretically 0 and can be estimated as

$$\gamma_{DR0} \approx A_{\gamma_{DR}}(t) = \frac{1}{N_L}\sum_{i=0}^{N_L-1} \gamma_{DR}(t - T_s i) \approx 0, \quad (10)$$

where $t = kT_s$, with $k \in \mathbb{Z}$, $T_s \in \mathbb{R}$, and $N_L \in \mathbb{Z}^{++}$.

Two fundamental signals in the analysis of autonomous flight and in the development of a suitable control strategy are the average lift force γ_{LR0} in Eq. 9 and the analogous average lift force γ_{LL0}, corresponding to the right wing and left wing, respectively. Similarly, other fundamental signals are the average pitch torque t_{P0} and average roll torque t_{R0}, to be discussed later in this paper, associated with the sampled instantaneous pitch torque $\mathbf{t}_P(t)$ and sampled instantaneous roll torque $\mathbf{t}_R(t)$, respectively. The reason why we care about these average signals is that the system dynamics filter out the high frequencies of the instantaneous signals acting on the system. To see this, consider the constrained one-dimensional case in [4], where the equation of motion is simply given by

$$\gamma_L(\tau) - mg - c\dot{p}_z(\tau) = m\ddot{p}_z(\tau), \quad (11)$$

where $\gamma_L(\tau) = \gamma_{LR}(\tau) + \gamma_{LL}(\tau)$, m is the robot's mass, g is the standard gravity acceleration constant, c is a damping coefficient, and $p_z(\tau)$ is the measured robot's altitude. Thus, defining $x(\tau) = \gamma_L(\tau) - mg$, it follows that the mapping from $x(\tau)$ to $p_z(\tau)$ is the low-pass filter

$$L(s) = \frac{1}{cs + ms^2}. \quad (12)$$

Thus, for the parameters of the system considered here, as can be easily demonstrated through simulations, the digital inputs $x(t)$ and $A_x(t) = \frac{1}{N+1}\sum_{n=0}^{N} x(t-n)$ produce the same outputs when the discrete-time version of Eq. 12 is excited with them. This also means that when Eq. 12 is excited with the continuous-time versions of $x(t)$ and $A_x(t)$, $x(\tau)$ and $A_x(\tau)$, the outputs are essentially the same.

By definition, in steady state (constant amplitude and constant frequency of the flapping angle), γ_{LR0} is constant. However, if the DC component of $\gamma_{LR}(t)$ is changed at a rate significantly slower than the rate of change of $\gamma_{LR}(t)$ itself, we can think of γ_{LR0} as a function that stays approximately constant during a stroke cycle, but that might change from one cycle to another. Thus, it is possible to assume that for a given stroke cycle, the DC component γ_{LR0} and the mean total force, Φ_R, generated by a single flapping wing (the right wing in this case) throughout a stroke, take the same value. This hypothesis agrees with the data obtained through the static flapping-wing experiments published in [3, 4]. In biological literature on flapping-wing flight [14, 22, 23], Φ_R is often estimated as

$$\Phi_R = \int_0^{\Xi} \rho \overline{C}_{\Phi_R} \overline{v_r^2}(\xi) c_R(\xi) d\xi, \qquad (13)$$

which is a standard quasi-steady blade-element formulation of flight force (see [20] and references therein), where ρ is the density of the air, $\overline{C}_{\Phi_R}$ is the mean force coefficient of the wing throughout the stroke, $\overline{v_r^2}(\xi)$ is the mean square relative velocity of each wing section, $c_R(\xi)$ is the chord length of the wing at a distance ξ from the base, and Ξ is the total wing length. In the case of a sinusoidal stroke of frequency f_R Hz, $\varphi_R(\tau) = \varphi_{AR}\sin(2\pi f_R \tau)$, with a horizontal stroke plane, the mean square relative velocity of each wing section can be roughly estimated as

$$\begin{aligned}\overline{v_r^2}(\xi) &= 4\pi^2 f_R^2 \xi^2 \varphi_{A_R}^2 \frac{1}{T_R}\int_0^{T_R} \cos^2(2\pi f_R \tau) d\tau \\ &= 2\pi^2 \xi^2 \varphi_{A_R}^2 f_R^2, \end{aligned} \qquad (14)$$

with $T_R = f_R^{-1}$. This implies that, regardless of the size and shape of the wing, the estimated mean total flight force directly depends on f_R^2 and $\varphi_{A_R}^2$, which indicates that in order for flying insects to accelerate against gravity or hover at a desired altitude, they can modulate the average lift force by changing the stroke amplitude, φ_{A_R}, or by changing the stroke frequency, f_R. Clearly, the same analysis is valid for the left wing.

At this point, we have all the basic information for proposing an actuation strategy that would allow us to directly modify the pitch and roll torques, and the lift force acting on the robot. First, let us consider the lift force and assume that both wings are flapped at the same frequency, i.e., $f_R = f_L$, then from Eqs. 13 and 14 it immediately follows that the magnitude of the total average lift force acting on the robot, $\gamma_{L0} = \gamma_{LL0} + \gamma_{LR0}$, can be modulated by simultaneously varying the sampled amplitudes $\varphi_{A_L}(t)$ and $\varphi_{A_R}(t)$, which are proportional to $\alpha_L(t)$ and $\alpha_R(t)$, because here the mapping from $[\nu_R(t)\ \nu_L(t)]^T$ to $[\varphi_R(t)\ \varphi_L(t)]^T$ is approximately LTI. Now, considering Eqs. 13 and 14, it immediately follows that the magnitude of the sampled total average roll torque acting on the robot, t_{R0}, associated with the instantaneous magnitude $t_R(t)$, can be modulated by flapping the robot's right and left wings with different amplitudes, i.e., $\varphi_R(t) \neq \varphi_L(t)$, which are proportional to $\alpha_L(t)$ and $\alpha_R(t)$, because as explained before, the mapping from $[\nu_R(t)\ \nu_L(t)]^T$ to $[\varphi_R(t)\ \varphi_L(t)]^T$ is approximately LTI. Notice that from Eq. 14, it follows that it is also possible to modulate forces by varying the flapping frequency. This notion was experimentally demonstrated correct in the work published in [19].

Finally in this subsection, we discuss the actuation method used to generate and modulate pitch torques. As discussed in [3], a biased flapping with respect to the $\hat{x}$ axis, i.e., $\varphi_{B_R}(\tau) = \varphi_{B_L}(\tau) > 0$, produces a sampled total average pitch torque t_{P0}, associated with the instantaneous magnitude $t_P(t)$, about the pitch axis that allows us to control the pitch angle. The reason for the generation of this effect is that the trajectory of the center of force, in red in Fig. 5, is rotated about the $\hat{z}$ axis, so that, the center of motion of the center of force, marked with a symbol $\otimes$ in Fig. 4, is shifted off the $\hat{x}$ axis. Once again, since the mapping from $[\nu_R(t)\ \nu_L(t)]^T$ to $[\varphi_R(t)\ \varphi_L(t)]^T$ is approximately LTI, it follows that the sampled biases $\varphi_{B_R}(t)$ and $\varphi_{B_L}(t)$ are proportional to $\beta_R(t)$ and $\beta_L(t)$, as defined by Eqs. 5 and 6, respectively.

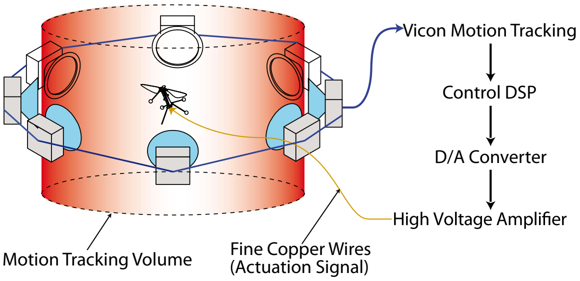

Fig. 6 Schematic of the Vicon motion capture system used as a sensor. In this case, six high-speed cameras capture the position of four reflective markers (small white balls) at 500 *frames per second* (FPS). The positions of the four reflective markers are used to determine the center of the frame of coordinates fixed to the robot's body, $\{^0p_x, ^0p_y, ^0p_z\}$, and the attitude of the robot defined by the Euler angles $\{\theta_1, \theta_2, \theta_3\}$, which correspond to three consecutive rotations about the $\hat{x}$ (pitch), $\hat{y}$ (roll) and $\hat{z}$ (yaw) axes fixed to the robot's body. The center of the body frame coincides with the body center of mass

2.2 Sensing

The feedback control strategy proposed in this paper is implemented using the robot's position and orientation, measured using the Vicon motion capture system described in [3] and illustrated in Fig. 6. The orientation of the robot with respect to a fixed inertial frame is computed and then represented by the Euler angles $\{\theta_1, \theta_2, \theta_3\}$, which correspond to three consecutive rotations about the $\hat{x}$ (pitch), $\hat{y}$ (roll) and $\hat{z}$ (yaw) axes fixed to the robot's body. Thus, using the standard notation $S_k = \sin\theta_k$ and $C_k = \cos\theta_k$, for $k = 1, 2, 3$, it follows that

$$^0\mathbf{p} = R_{0\leftarrow b}{}^b\mathbf{p}, \tag{15}$$

where $^0\mathbf{p}$ is the vector $\mathbf{p}$ written with respect to the inertial frame $\left\{\hat{X}, \hat{Y}, \hat{Z}\right\}$, $^b\mathbf{p}$ is the same vector $\mathbf{p}$ written with respect to the frame $\left\{\hat{x}, \hat{y}, \hat{z}\right\}$, fixed to the robot's body, and

$$R_{0\leftarrow b} = \begin{bmatrix} C_2C_3 & -S_3C_2 & S_2 \\ S_3C_1 + S_1S_2C_3 & C_1C_3 - S_1S_2S_3 & -S_1C_2 \\ S_1S_3 - S_2C_1C_3 & S_1C_3 + S_2S_3C_1 & C_1C_2 \end{bmatrix}. \tag{16}$$

From this point onwards, the $\{i, j\}$-entry of $R_{0\leftarrow b}$ will be denoted by $R^{(ij)}_{0\leftarrow b}$. Notice that from definition (16) it follows that $R^{-1}_{0\leftarrow b} = R^T_{0\leftarrow b}$.

The feedback controller described in Section 3, depicted in Fig. 8, in the most general case, requires the computation of the instantaneous pitch, roll, and yaw angles. The algorithm for computing the pitch angle is as follows.

Algorithm 1

```
(i)   Find the plane that contains the pitch axis and is perpendicular to the fixed inertial horizontal plane.
(ii)  Find the plane that contains the pitch axis and the yaw axis.
(iii) Find the dihedral angle between the planes in (ii) and (iii).
```

To translate Algorithm 1 into a formula, first note that the axes of the frame fixed to the robot's body expressed in the inertial frame are simply

$$^0\hat{x} = R_{0\leftarrow b}{}^b\hat{x}, \quad ^0\hat{y} = R_{0\leftarrow b}{}^b\hat{y}, \quad ^0\hat{z} = R_{0\leftarrow b}{}^b\hat{z}, \tag{17}$$

where $^b\hat{x} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$, $^b\hat{y} = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ and $^b\hat{z} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$. Recalling that the axes of the inertial frame written with respect to the inertial frame are $^0\hat{X} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$, $^0\hat{Y} = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ and $^0\hat{Z} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$, it follows that a vector perpendicular to the plane defined in (i), written with respect to the inertial frame, can be computed as $^0\mathbf{n}_1 = {}^0\hat{x} \times {}^0\hat{Z}$. Similarly, it follows that a vector perpendicular to the plane defined in (ii), written with respect to the inertial frame, is $^0\mathbf{n}_2 = {}^0\hat{x} \times {}^0\hat{z}$. Then, it follows that the dihedral angle between the two planes defined in items (i) and (ii) of Algorithm 1, the pitch angle, can be computed as

$$\theta_{Pitch} = \left[\mathrm{sign}\left(R^{(32)}_{0\leftarrow b}\right)\right]\left[\arccos\left(\frac{^0\mathbf{n}_1 \cdot {}^0\mathbf{n}_2}{\|^0\mathbf{n}_1\|\|^0\mathbf{n}_2\|}\right)\right], \tag{18}$$

where $-\pi \leqslant \theta_{Pitch} \leqslant \pi$.

A similar method to the one used to compute θ_{Pitch} can be employed to find the roll angle θ_{Roll}. Thus, after some algebraic work it follows that

$$\theta_{Roll} = -\left[\mathrm{sign}\left(R^{(31)}_{0\leftarrow b}\right)\right]\left[\arccos\left(\frac{^0\boldsymbol{\eta}_1 \cdot {}^0\boldsymbol{\eta}_2}{\|^0\boldsymbol{\eta}_1\|\|^0\boldsymbol{\eta}_2\|}\right)\right], \tag{19}$$

where $-\pi \leqslant \theta_{Roll} \leqslant \pi$, with $^0\boldsymbol{\eta}_1 = {}^0\hat{y} \times {}^0\hat{Z}$ and $^0\boldsymbol{\eta}_2 = {}^0\hat{y} \times {}^0\hat{z}$.

Finally, using the same logic, the yaw angle θ_{Yaw} can be computed as

$$\theta_{Yaw} = \left[\text{sign}\left(R_{0\leftarrow b}^{(21)}\right)\right]\left[\arccos\left(\frac{{}^0\varsigma_1 \cdot {}^0\varsigma_2}{\|{}^0\varsigma_1\|\|{}^0\varsigma_2\|}\right)\right], \tag{20}$$

where $-\pi \leqslant \theta_{Yaw} \leqslant \pi$, ${}^0\varsigma_1 = {}^0\hat{Z} \times {}^0\hat{x}$ and ${}^0\varsigma_2 = {}^0 \hat{Y}$.

2.3 Filtering

As it is the case of most sensing systems, the position and orientation information gathered using the Vicon motion capture system is degraded by noise. In this case, a particular kind of noise appears as spikes added to the measured $\left\{{}^0p_x, {}^0p_y, {}^0p_z, \theta_1, \theta_2, \theta_3\right\}$. It seems that spikes occur at the boundary of the tracking volume, where only two of the eight cameras can see the robot, and in general, when one or more markers are not seen by any of the cameras. If spikes appear sporadically, an extrapolation filter can be used to filter them out, when detected. For a generic variable $\psi(t)$, we propose the estimation scheme

$$\tilde{\psi}(t-\zeta) = \tilde{a}\,(t-\zeta)^2 + \tilde{b}\,(t-\zeta) + \tilde{c}, \tag{21}$$

where ζ is a positive integer. In order to estimate $\left\{\tilde{a}, \tilde{b}, \tilde{c}\right\}$, without loss of generality, we can assume that $t = 0$. Once $\left\{\tilde{a}, \tilde{b}, \tilde{c}\right\}$ are computed, we can estimate

$$\tilde{\psi}(1) = \tilde{a} + \tilde{b} + \tilde{c}. \tag{22}$$

In this case, we consider $\zeta = kN$, where $0 < N \in \mathbb{Z}$ and $k = \{0, 1, 2, 3, \cdots, K\}$, where $0 < K \in \mathbb{Z}$. For example, for $K = 5$, we can formulate the *least-squares* (LS) [24] problem

$$\min_{\left\{\tilde{a},\tilde{b},\tilde{c}\right\}} \left\| \Psi - A \begin{bmatrix} \tilde{a} \\ \tilde{b} \\ \tilde{c} \end{bmatrix} \right\|_2, \tag{23}$$

where

$$\Psi = \begin{bmatrix} \psi(-5N) \\ \psi(-4N) \\ \psi(-3N) \\ \psi(-2N) \\ \psi(-N) \\ \psi(0) \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} 25N^2 & -5N & 1 \\ 16N^2 & -4N & 1 \\ 9N^2 & -3N & 1 \\ 4N^2 & -2N & 1 \\ N^2 & -N & 1 \\ 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

Thus, the searched parameters can be estimated by simply finding the optimal solution to Eq. 23, using the well-known *singular value decomposition* (SVD)-based method, which implies that

$$\begin{bmatrix} \tilde{a}^\star \\ \tilde{b}^\star \\ \tilde{c}^\star \end{bmatrix} = A^\dagger \Psi, \tag{25}$$

where $A^\dagger$ is the pseudo inverse of A, which implies that

$$\begin{aligned} \tilde{a}^\star + \tilde{b}^\star + \tilde{c}^\star = {} & \alpha_6\psi(0) + \alpha_5\psi(-N) + \alpha_4\psi(-2N) \\ & + \alpha_3\psi(-3N) + \alpha_2\psi(-4N) \\ & + \alpha_1\psi(-5N), \end{aligned} \tag{26}$$

with $\alpha_j = \sum_{i=1}^{3} A_{ij}^\dagger$, where $j = \{1, \cdots, 6\}$ and $A_{ij}^\dagger$ is the $\{i, j\}$-entry of the matrix $A^\dagger$. Thus, we can write the estimator

$$\begin{aligned} \tilde{\psi}(t+1) = {} & \alpha_6\psi(t) + \alpha_5\psi(t-N) + \alpha_4\psi(t-2N) \\ & + \alpha_3\psi(t-3N) + \alpha_2\psi(t-4N) \\ & + \alpha_1\psi(t-5N), \end{aligned} \tag{27}$$

which in filter notation simply becomes

$$\tilde{\psi}(t) = \frac{\alpha_6 z^{5N} + \alpha_5 z^{4N} + \alpha_4 z^{3N} + \alpha_3 z^{2N} + \alpha_2 z^{N} + \alpha_1}{z^{5N+1}} \psi(t). \tag{28}$$

An example demonstrating the effectiveness of the proposed extrapolation filter, with $N = 100$, is shown in Fig. 7. In this figure, the upper plot compares a raw with a filtered measured Euler angle θ_1 signal, and the lower plot shows a closeup of the upper plot. The Vicon system runs at 500 Hz, but the control algorithms are run at a sample-and-hold rate of 10 KHz, using an xPC-target system. This speed difference between the sensing system and the controller implies that each measurement is recorded by the *digital signal processor* (DSP) 20 times approximately, and therefore, with $N = 100$, the filter is looking back approximately 5 unique Vicon data points.

3 Control Strategy

The main research issue addressed in this paper is the development of a model-free tunable strategy for simultaneously controlling the variables $\theta_{Pitch}(t)$, $\theta_{Roll}(t)$ and ${}^0p_z(t)$, associated with the prototype described in Section 2, during unconstrained flight. In

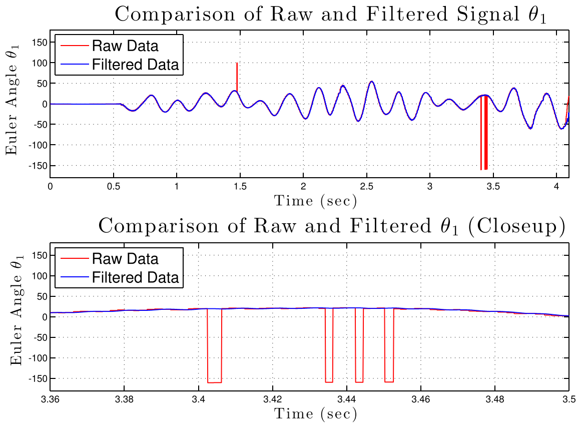

Fig. 7 Example demonstrating the use of the extrapolation filter in Eqs. 27 and 28. The *upper plot* shows a comparison of a raw with a filtered measured Euler angle θ_1 signal. The *lower plot* shows a closeup of the upper plot

principle, there are many options in order to create a control strategy for achieving stable and autonomous flight of the robot. As explained in Section 2, here we directly modulate the controlled variables by varying signals $\nu_R(t)$ and $\nu_L(t)$ as defined by Eqs. 5 and 6. With this purpose in mind, we define

$$\beta(t) = \beta_R(t) = \beta_L(t), \tag{29}$$

$$\alpha_R(t) = \alpha_{R0}(t) + \alpha_{R1}(t) + \alpha_{R2}(t), \tag{30}$$

$$\alpha_L(t) = \alpha_{L0}(t) + \alpha_{L1}(t) + \alpha_{L2}(t), \tag{31}$$

and then we choose

$$\alpha_0(t) = \alpha_{R0}(t) = \alpha_{L0}(t), \tag{32}$$

$$\alpha_1(t) = \alpha_{R1}(t) = -\alpha_{L1}(t), \tag{33}$$

$$\alpha_2(t) = \alpha_{R2}(t) = \alpha_{L2}(t). \tag{34}$$

Here, $\alpha_0(t)$ is chosen to be a constant that generates a baseline lift force. The variable $\beta(t)$ is used to modulate θ_{Pitch}, the variable $\alpha_1(t)$ is used to modulate $\theta_{Roll}(t)$ and the variable $\alpha_2(t)$ is used to modulate the altitude ${}^0p_z(t)$. Thus, as described in Fig. 8, the two relevant vectorial signals in the control problem considered here are chosen to be

$$u_B(t) = \begin{bmatrix} \beta(t) \\ \alpha_1(t) \\ \alpha_2(t) \end{bmatrix}, \qquad y_B(t) = \begin{bmatrix} \tilde{\theta}_{Pitch}(t) \\ \tilde{\theta}_{Roll}(t) \\ {}^0\tilde{p}_z(t) \end{bmatrix}, \tag{35}$$

where the tilde symbols employed in the definition of $y_B(t)$ are used to emphasize the fact that $y_B(t)$ emerges from an estimation process (using the Vicon motion capture system described in the previous subsection), and consequently, each entry of $y_B(t)$ is the sum of the true value of a variable ($\theta_{Pitch}(t)$, $\theta_{Roll}(t)$ or ${}^0p_z(t)$) and sensor noise. Recalling that due to physical constraints on the manner in which the actuators should be excited, the exciting signals must satisfy that $-1 \leqslant \nu_R(t) \leqslant 1$ and $-1 \leqslant \nu_L(t) \leqslant 1$, and therefore, it follows that

$$0 \leqslant \alpha_R(t) + \beta_R(t) \leqslant 1, \tag{36}$$

$$0 \leqslant \alpha_L(t) + \beta_L(t) \leqslant 1, \tag{37}$$

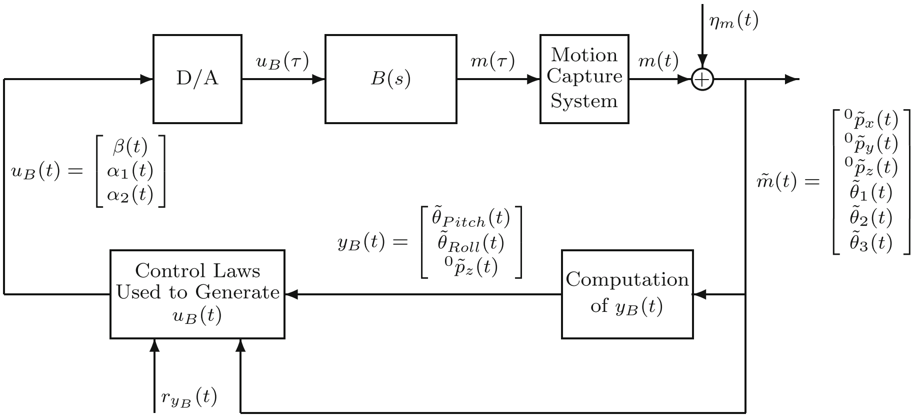
$\eta_m(t)$
D/A
$u_B(\tau)$
$B(s)$
$m(\tau)$
Motion Capture System
$m(t)$
$u_B(t) = \begin{bmatrix} \beta(t) \\ \alpha_1(t) \\ \alpha_2(t) \end{bmatrix}$
$\tilde{m}(t) = \begin{bmatrix} {}^0\tilde{p}_x(t) \\ {}^0\tilde{p}_y(t) \\ {}^0\tilde{p}_z(t) \\ \tilde{\theta}_1(t) \\ \tilde{\theta}_2(t) \\ \tilde{\theta}_3(t) \end{bmatrix}$
$y_B(t) = \begin{bmatrix} \tilde{\theta}_{Pitch}(t) \\ \tilde{\theta}_{Roll}(t) \\ {}^0\tilde{p}_z(t) \end{bmatrix}$
Control Laws Used to Generate $u_B(t)$
Computation of $y_B(t)$
$r_{y_B}(t)$

Fig. 8 Upper level flight control diagram. The motion capture system is the sensor used to estimate the robot position $\left\{{}^0p_x(t), {}^0p_y(t), {}^0p_z(t)\right\}$ and orientation $\{\theta_1(t), \theta_2(t), \theta_3(t)\}$. The controlled variables are $\left\{\theta_{Pitch}(t), \theta_{Roll}(t), {}^0p_z(t)\right\}$. The measurement error $\eta_m(t)$ is transformed to sensor noise $\eta_B(t)$. The control signal is $u_B(t) = [\beta(t)\ \alpha_1(t)\ \alpha_2(t)]^T$. The reference is $r_{y_B}(t) = \left[r_{\theta_{Pitch}}(t)\ r_{\theta_{Roll}}(t)\ r_{{}^0p_z}(t)\right]^T$

which implies that $\beta(t)$, $\alpha_1(t)$ and $\alpha_2(t)$ must meet strict saturation limits.

The approach proposed in this work for finding an experimentally feasible and stable controller is very intuitive and based on the idea illustrated in Fig. 9. In this figure, it is easy to see that if we assume a perfectly symmetric aerodynamic design and the absence of fabrication errors, a simple experimentally-tuned MIMO LTI controller with the form

$$u_B(t) = [C(z)e_B](t) = \left[C(z)\left(r_{y_B} - y_B\right)\right](t) \qquad (38)$$

and an appropriate reference r_{y_B} can be used to compensate for deviations from a desired vertical path. In this paper, $C(z)$ is a diagonal transfer matrix with the diagonal labeled as $\left\{C_{Pitch}(z), C_{Roll}(z), C_{^0p_z}(z)\right\}$. In general, the control strategy in Eq. 38 is insufficient, from a stability perspective, in order to deal with asymmetries produced by small fabrication errors and with strong external disturbances, such as wind gusts, for example. In this paper, we propose a combined robustly stable strategy that includes the operator defined by Eq. 38 and another component that processes the information associated with the robot's velocities along the ${}^0\hat{x}$ and ${}^0\hat{y}$ axes. This information is relevant because it relates to the restoring forces acting on the robot while flying horizontally. Furthermore, we experimentally demonstrate that the same proposed control method can be used for hovering and for implementing basic flying maneuvers. Note that, as shown in [4], the entry $C_{^0p_z}(z)$ is enough to control altitude.

The main control idea here, that follows from Fig. 9, consists of adding four decoupled filters to the general control scheme. The first two filters take the velocity and position of the robot along the roll axis, ${}^0\hat{y}$, as inputs and deliver outputs that are added to the total actuation $\beta(t)$, which allows us to robustly stabilize the pitch-angle degree of freedom. The additional two filters take the velocity and position of the robot along the pitch axis, ${}^0\hat{x}$, as inputs and deliver outputs that are added to the total actuation $\alpha_1(t)$, which allows us to robustly stabilize the roll-angle degree of freedom. It is important to note that with the appropriate tuning of these filters, the position of the robot on the $\hat{X}\hat{Y}$ plane can be controlled. The robot's position $\left\{{}^0p_x(t), {}^0p_y(t), {}^0p_z(t)\right\}$ is estimated using the previously described Vicon motion capture system. The velocity of the robot with respect to the inertial frame $\left\{\hat{X}, \hat{Y}, \hat{Z}\right\}$ is simply

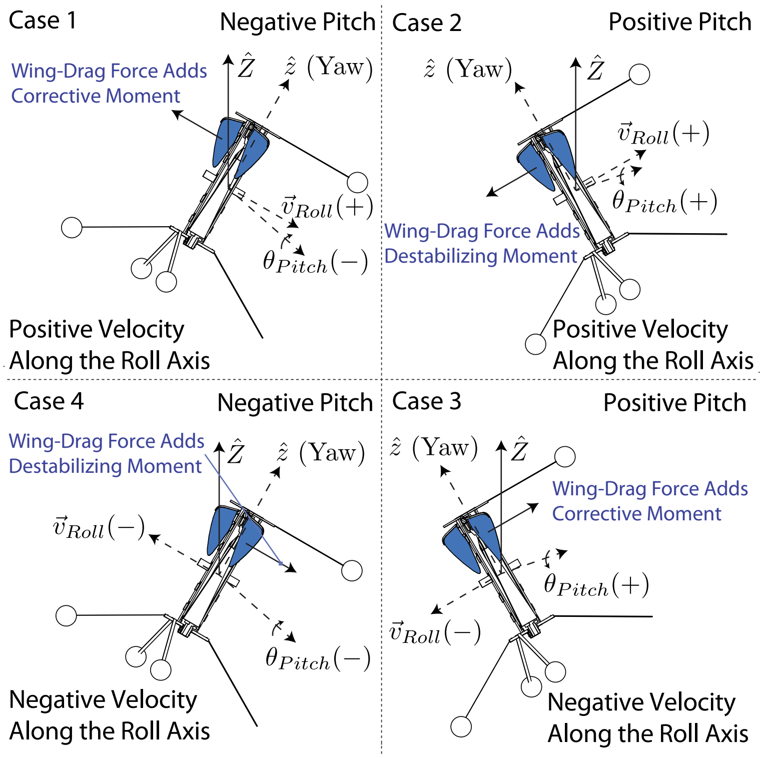

Fig. 9 Four flying states (cases), defined with respect to the pitch angle θ_{Pitch} and the robot velocity along the roll axis $\hat{y}$, $\mathbf{v}_{Roll}$

calculated as

$$
{}^0\mathbf{v}_p = \begin{bmatrix} {}^0\tilde{v}_{p_x}(t) \\ {}^0\tilde{v}_{p_y}(t) \\ {}^0\tilde{v}_{p_z}(t) \end{bmatrix} = \begin{bmatrix} \frac{{}^0\tilde{p}_x(t) - {}^0\tilde{p}_x(t-1)}{T_s} \\ \frac{{}^0\tilde{p}_y(t) - {}^0\tilde{p}_y(t-1)}{T_s} \\ \frac{{}^0\tilde{p}_z(t) - {}^0\tilde{p}_z(t-1)}{T_s} \end{bmatrix}, \qquad (39)
$$

where T_s is the sampling rate already defined.

First, we describe the method for finding the filter that maps the velocity of the robot along the roll axis to the output that is added to the total actuation $\beta(t)$. The robot's velocity and position along the roll axis are given by

$$
{}^0\mathbf{v}_{roll}(t) = \left[{}^0\mathbf{v}_p(t) \cdot {}^0\hat{y}(t) \right] {}^0\hat{y}(t), \qquad (40)
$$

$$
{}^0\mathbf{p}_{roll}(t) = \left[{}^0\mathbf{p}(t) \cdot {}^0\hat{y}(t) \right] {}^0\hat{y}(t), \qquad (41)
$$

where, as defined before, ${}^0\hat{y}(t) = R_{0\leftarrow b}{}^b\hat{y}(t)$.

The idea of using ${}^0\mathbf{v}_{roll}(t)$ in the computation of the control signal $\beta(t)$ follows from the cartoon in Fig. 9. This cartoon shows the four relevant cases (flying states) that should be considered in order to robustly stabilize the pitch-angle of the robot during flight. The main notion behind the definition of different cases is that restoring forces play a relevant role when the stability of the flying robot is considered. To see this, assume without loss of generality that the pitch angles depicted in Case 1 and Case 4 in Fig. 9 are identical and that the magnitudes of the robot's velocities along the roll axis are also identical in both cases. The main difference between Case 1 and Case 4 is that in Case 1 the interaction of the robot with the air produces a restoring force that adds a *corrective* pitch moment and that in Case 4 the interaction of the robot with the air produces a force that adds a *destabilizing* pitch moment. This tells us that a successful and robust control strategy must add a term to the left side of Eq. 38 in order to account for the different amount of actuation required in Case 4 with respect to Case 1.

Now, let us consider Cases 2 and 3. Analogous to Cases 4 and 1 (with negative pitch angles), in Cases 2 and 3 the pitch angle, θ_{Pitch}, is positive and without loss of generality, it is assumed that the magnitude of the pitch angle in Case 2 is identical to the pitch angle in Case 3. In Case 2, the robot's velocity along the roll axis is positive, which implies that wing drag adds a *destabilizing* pitch moment. In Case 3, the robot's velocity along the roll axis is negative, which implies that wing drag adds a *corrective* pitch moment. Thus, as argued in the previous paragraph, a new term must be added to the left side of Eq. 38 in order to account for the different amount of actuation required in Case 3 with respect to Case 2.

Thus, as a design choice, for the pitch angle degree of freedom, we propose a control structure with the form

$$
\begin{aligned}
\beta(t) = {} & C_{Pitch}(z) \left[r_{\theta_{Pitch}}(t) - \tilde{\theta}_{Pitch}(t) \right] \\
& - sign\left[{}^0\mathbf{v}_p(t) \cdot {}^0\hat{y}(t) \right] \\
& \cdot V_{Pitch}(z) \left[\left| {}^0\mathbf{v}_{Roll}(t) \right| \right],
\end{aligned} \qquad (42)
$$

which allows us to tune a robustly stable controller for the first entry of the MIMO strategy depicted in Fig. 8.

The analysis done for the pitch case can be replicated for the roll case. Thus, for the roll angle degree of freedom, we propose a control structure with the form

$$
\begin{aligned}
\alpha_1(t) = {} & C_{Roll}(z) \left[r_{\theta_{Roll}}(t) - \tilde{\theta}_{Roll}(t) \right] \\
& - sign\left[{}^0\mathbf{v}_p(t) \cdot {}^0\hat{x}(t) \right] \\
& \cdot V_{Roll}(z) \left[\left| {}^0\mathbf{v}_{Pitch}(t) \right| \right],
\end{aligned} \qquad (43)
$$

where

$$
{}^0\mathbf{v}_{Pitch}(t) = \left[{}^0\mathbf{v}_p(t) \cdot {}^0\hat{x}(t) \right] {}^0\hat{x}(t). \qquad (44)
$$

As we demonstrate in Section 4, the control strategy in Eqs. 42 and 43 allows us to stabilize the pitch angle and roll angle degrees of freedom, so that, the robot achieves unconstrained flight and autonomy from a control point of view. Thus, in theory, the robot can fly indefinitely. The main limitations of this approach is that there is not a direct way to set a reference for the position of the robot over the $\hat{X}\hat{Y}$ plane and that the robot displays a tendency to drift, because the method as proposed in Eqs. 42 and 43 allows for the robot to move laterally in order to reject random disturbances that might affect the pitch angle or roll angle degrees of freedom. Another limitation, contingent to the experimental setup employed in this work, is that when the robot drifts, it can fly outside the control volume observed by the Vicon motion capture system, described in Section 2 and illustrated in Fig. 6. Also, in this setup, the mobility of the robot is limited by the power wire (recall that the robot is not autonomous from a power point of view).

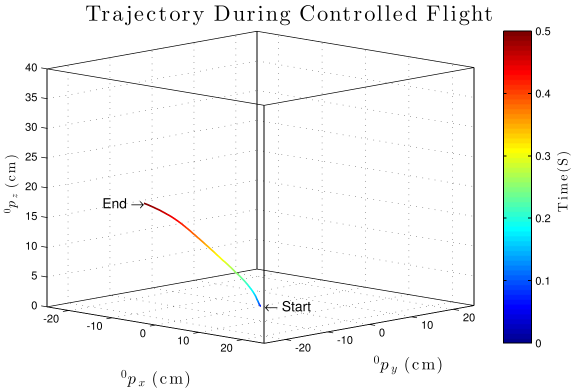

Fig. 10 3-D trajectory of the robot during controlled flight corresponding to Experiment 1

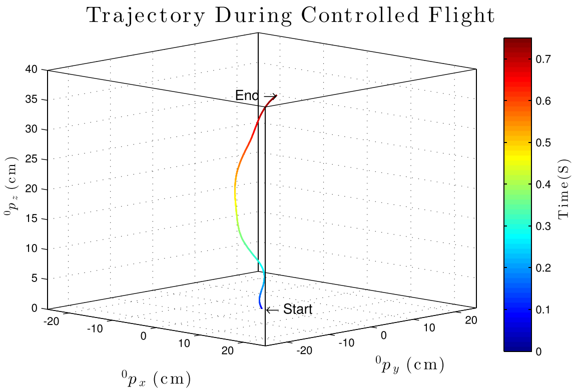

Fig. 12 3-D trajectory of the robot during controlled flight corresponding to Experiment 2

In order to deal with the limitations described in the previous paragraph, we modify the control strategy defined by Eqs. 42 and 43 as follows

$$\begin{aligned}\beta(t) = {} & C_{Pitch}(z)\left[r_{\theta_{Pitch}}(t) - \tilde{\theta}_{Pitch}(t)\right] \\ & - sign\left[{}^0\mathbf{v}_p(t)\cdot{}^0\hat{y}(t)\right] \\ & \cdot V_{Pitch}(z)\left[\left|{}^0\mathbf{v}_{Roll}(t)\right| - r_{|{}^0\mathbf{v}_{Roll}(t)|}\right] \\ & - sign\left[{}^0\mathbf{p}(t)\cdot{}^0\hat{y}(t)\right] \\ & \cdot P_{Pitch}(z)\left[\left|{}^0\mathbf{p}_{Roll}(t)\right| - r_{|{}^0\mathbf{p}_{Roll}(t)|}\right]\end{aligned} \tag{45}$$

and

$$\begin{aligned}\alpha_1(t) = {} & C_{Roll}(z)\left[r_{\theta_{Roll}}(t) - \tilde{\theta}_{Roll}(t)\right] \\ & - sign\left[{}^0\mathbf{v}_p(t)\cdot{}^0\hat{x}(t)\right] \\ & \cdot V_{Roll}(z)\left[\left|{}^0\mathbf{v}_{Pitch}(t)\right| - r_{|{}^0\mathbf{v}_{Pitch}(t)|}\right] \\ & - sign\left[{}^0\mathbf{p}(t)\cdot{}^0\hat{x}(t)\right] \\ & \cdot P_{Roll}(z)\left[\left|{}^0\mathbf{p}_{Pitch}(t)\right| - r_{|{}^0\mathbf{p}_{Pitch}(t)|}\right],\end{aligned} \tag{46}$$

where,

$${}^0\mathbf{p}_{Pitch}(t) = \left[{}^0\mathbf{p}(t)\cdot{}^0\hat{x}(t)\right]{}^0\hat{x}(t). \tag{47}$$

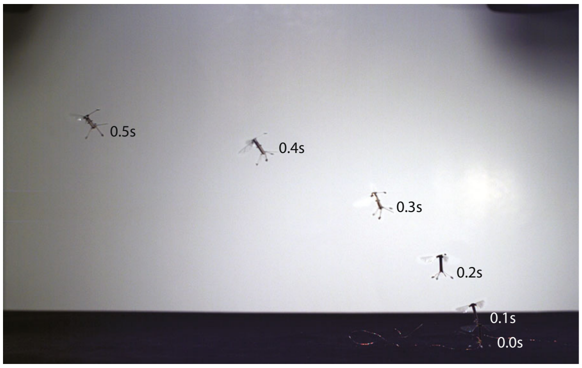

Fig. 11 Frames from high-speed video of Experiment 1 are overlaid and composited together to show the frontal view of the robot trajectory during controlled flight. The complete experiment is shown in the supplementary movie S1.mp4, also available online through the website pointed by [25]

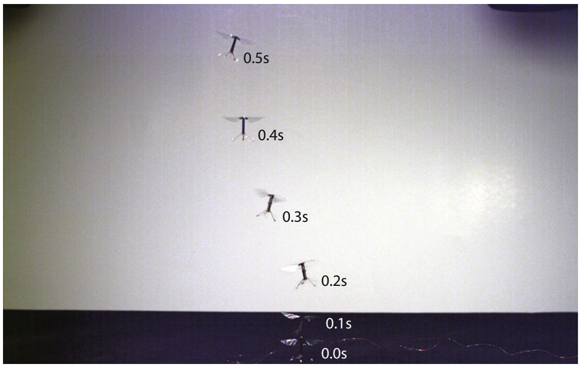

Fig. 13 Frames from high-speed video of Experiment 2 are overlaid and composited together to show the frontal view of the robot trajectory during controlled flight. The complete experiment is shown in the supplementary movie S1.mp4, also available online through the website pointed by [25]

The logic behind this modified structure is that filters $V_{Pitch}(z)$ and $V_{Roll}(z)$, which are LTI, can be chosen to have a *proportional-integral-derivative* (PID) form, which implies that another interpretation of the laws in Eqs. 42 and 43 is that the second term in Eq. 42 and the second term in Eq. 43 minimize the magnitudes of ${}^{0}\mathbf{v}_{Roll}(t)$ and ${}^{0}\mathbf{v}_{Pitch}(t)$, so that in steady state, velocities equal to 0 are achieved along the roll and pitch axes, respectively. Thus, the same filters $V_{Pitch}(z)$ and $V_{Roll}(z)$ can be used to minimize an error and follow a reference, as is done in the second terms of Eq. 45 and 46. Similarly, using the third terms of Eqs. 45 and 46, the position of the robot on the $\hat{X}\hat{Y}$ can be controlled. Here, the structures of filters $P_{Pitch}(z)$ and $P_{Roll}(z)$ are also PID.

4 Experimental Results and Discussion

In this section, we present three representative experimental cases that demonstrate the most relevant aspects of the model-free control strategy proposed in this paper. In the course of the research discussed here, the first set of experiments conducted were attempts of open-loop unconstrained flight, which as expected, did not achieve hovering because the MIMO open-loop plant of the system is unstable. A representative experiment of open-loop flight, labeled as Experiment 0, is shown in the accompanying video S1.mp4, also available at [25], where it can be seen that the robot lifts off but topples over almost immediately.

Experiment 1 shows a stable flight employing the method in Eqs. 42–43, which uses $V_{Pitch}(z)$ and $V_{Roll}(z)$ to filter the robot's velocities along the roll and pitch axes, respectively in order to achieve robust stability. As previously explained, $V_{Pitch}(z)$ and $V_{Roll}(z)$ were developed based on the idea described in Fig. 9. The results for Experiment 1 are summarized by a plot of the 3-D trajectory in Fig. 10 and a composite image showing frames of a frontal view taken with a high speed camera is shown in Fig. 11. The controller used in Experiment 1 was repeatedly tested and consistently demonstrated robust stability in the sense that the robot was capable of flying oriented upright and not toppling over, even in the presence of strong disturbances such as wind gusts generated by the room air conditioning system. In Figs. 10 and 11, it can be seen that the robot drifts from its starting position on the horizontal plane, so that, once the robot leaves the tracking volume, the

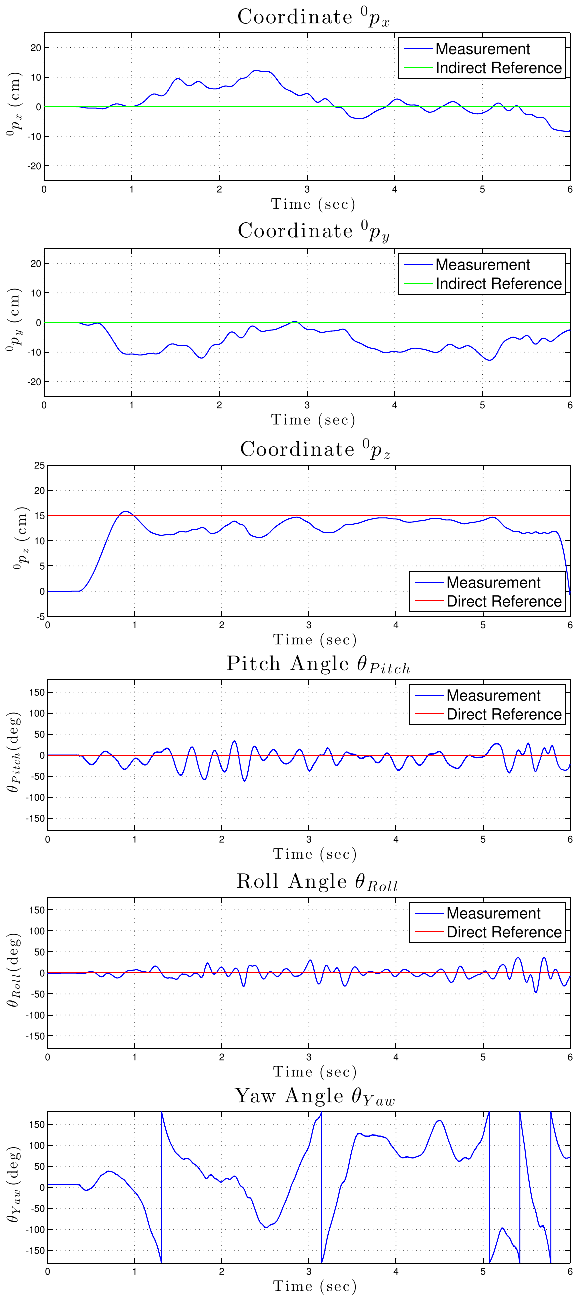

Fig. 14 Experiment 3 time-series of the robot's degrees of freedom during unconstrained controlled flight. The variables ${}^{0}p_z(t)$, $\theta_{Pitch}(t)$ and $\theta_{Roll}(t)$ are actively controlled, using direct references. The variables ${}^{0}p_x(t)$ and ${}^{0}p_y(t)$ are indirectly controlled by defining references for $\theta_{Pitch}(t)$ and $\theta_{Roll}(t)$. The variable $\theta_{Yaw}(t)$ is not controlled and it is allowed to drift freely

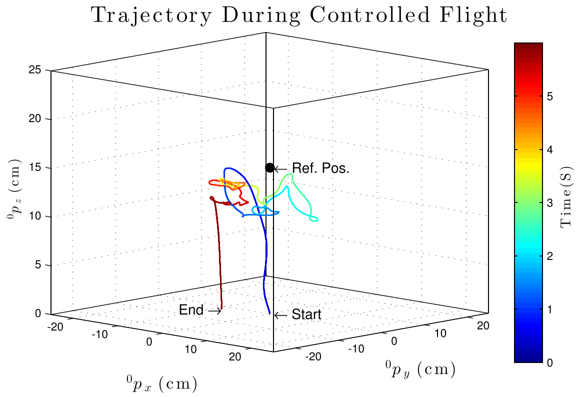

Fig. 15 3-D trajectory of the robot during controlled flight corresponding to Experiment 3

experiment ends. We believe that this drifting behavior is partially due to the effect of the velocity filters, $V_{Pitch}(z)$ and $V_{Roll}(z)$, which during the transient time help to stabilize orientation, but potentially add lateral velocity resulting in drift.

In Experiment 2, the robot flies employing the control laws in Eqs. 45 and 46, which address the drift issue observed in Experiment 1 by adding a correction for position on the horizontal plane, using the position filters $P_{Pitch}(z)$ and $P_{Roll}(z)$. The results for Experiment 2 are summarized by a plot of the 3-D trajectory in Fig. 12 and a composite image showing frames of a frontal view taken with a high speed camera shown in Fig. 13. These two figures demonstrate that the control laws in Eqs. 45 and 46 make the system robustly stable in the sense that the robot flies oriented upright, even in the presence of strong disturbances. Simultaneously, the control laws in Eqs. 45

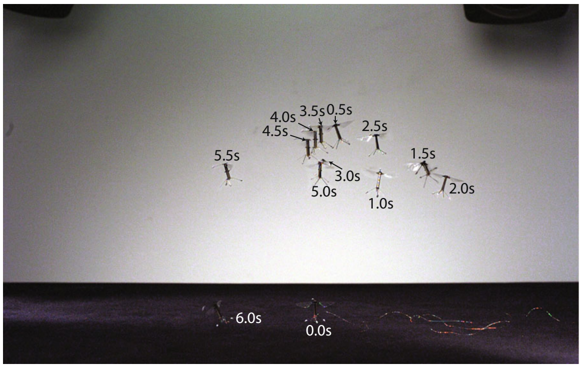

Fig. 16 Frames from high-speed video of Experiment 3 are overlaid and stitched together to show the frontal view of the robot trajectory during controlled flight. The complete experiment is shown in the supplementary movie S1.mp4, also available online through the website pointed by [25]

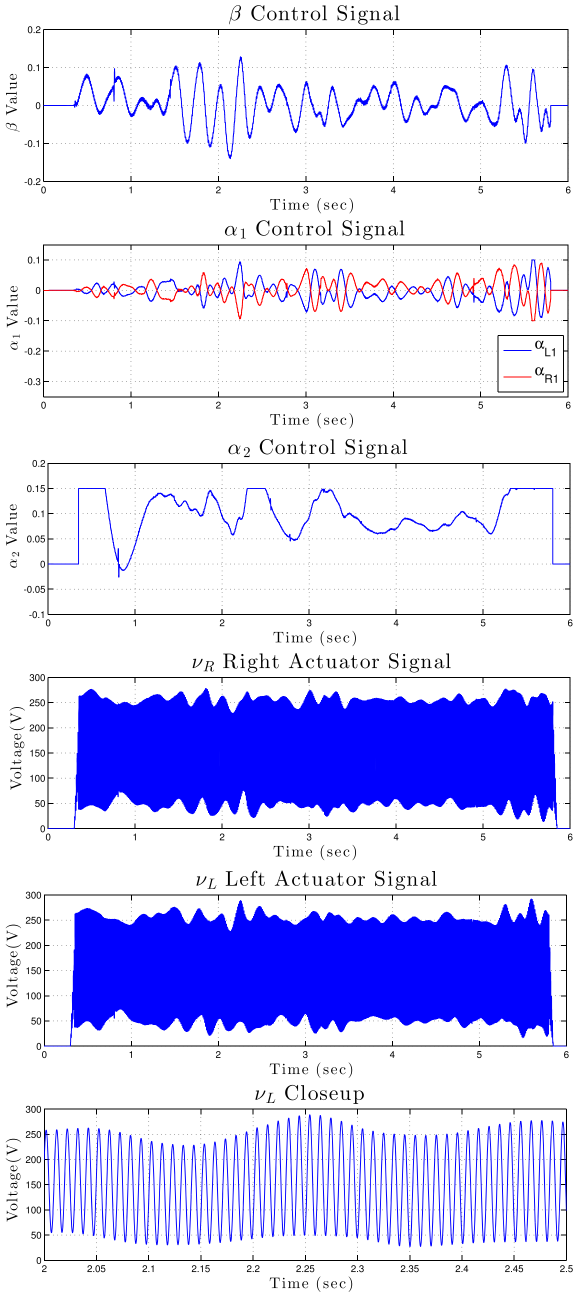

Fig. 17 Experiment 3 control signals β (top plot), α_1 (second plot), and α_2 (third plot) modulate the pitch torque, roll torque, and lift generated by the robot, respectively. These control signals are mapped to actuator input signals for the right actuator (fourth plot) and left actuator (fifth plot). A closeup of the left actuator input is shown in the bottom plot. Notice that the inputs to each actuator are different, because the two actuators are independent

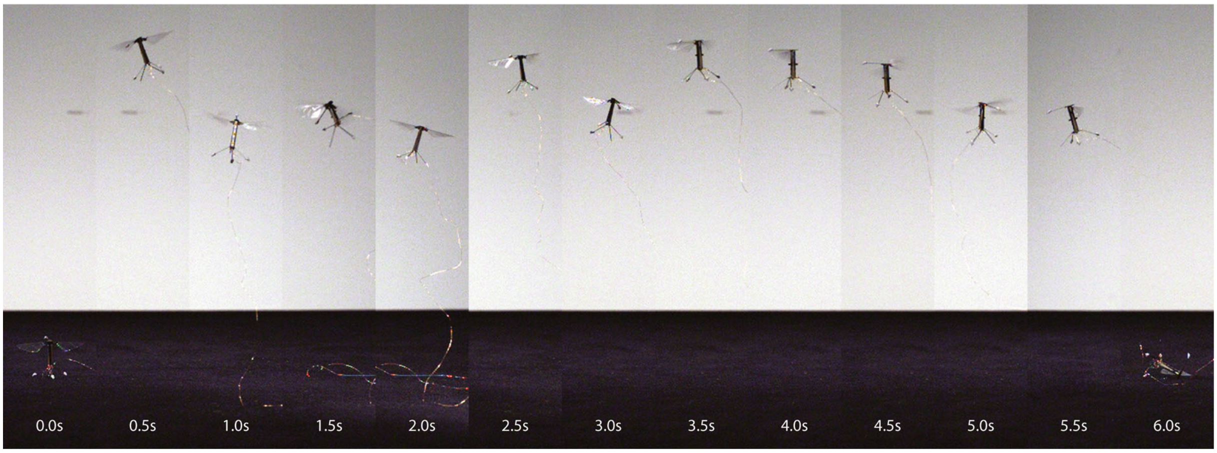

Fig. 18 Photographic sequence showing a controlled flight of the robot, corresponding to Experiment 3 described in Figs. 14, 15, 16 and 17. The complete experiment is shown in the supplementary movie S1.mp4, also available online through the website pointed by [25]

and 46 correct for and eliminate the drifting phenomenon observed in Experiment 1, such that, the robot flies straight up. The controller used in Experiment 2 was repeatedly tested and consistently proved capable of achieving stable straight vertical flight for the robot. In this kind of experiment, once the robot leaves from the top the tracking volume, the experiment ends.

Finally, Experiment 3 shows hovering and trajectory following. In this case, we use the same strategy in Eqs. 45 and 46 for stabilizing the system and controlling position on the $\hat{X}\hat{Y}$ plane. But, simultaneously we control altitude by choosing a finite reference for this degree of freedom (15 cm in this case). In Fig. 14, we show all the relevant degrees of freedom of the robot while flying. As explained in Section 3, ${}^{0}p_z(t)$, $\theta_{Pitch}(t)$ and $\theta_{Roll}(t)$ are directly controlled, in this case with references 15 cm, 0 rad and 0 rad. The variables ${}^{0}p_x(t)$ and ${}^{0}p_y(t)$ are indirectly controlled using the second and third terms of the control laws in Eqs. 45 and 46, in this case with references 0 cm and 0 cm. The variable $\theta_{Yaw}(t)$ is not controlled and it is allowed to drift freely. Figure 15 shows the 3-D trajectory of the robot while flying, and a composite image, formed with the frames of a frontal view taken with a high speed camera, is shown in Fig. 16. The relevant control outputs are shown in Fig. 17. Lastly, a photographic sequence showing the robot's position and orientation through time is shown in Fig. 18. Figures 14, 15, 16, 17 and 18 present compelling and comprehensive evidence that the robot remains stable and hovers around the reference position for six seconds, until the controller is turned off. The supplementary movie S1.mp4, also available online at [25], presents all the cases discussed in this section.

5 Conclusion

We have described the development of a model-free control strategy and its associated controller design method for achieving stable autonomous flight and hovering of an insect-scale flapping-wing flying microrobot. By experimentally identifying and tuning key parameters affecting flight stability, a relatively simple MIMO controller resulted, where each term of the controller has a known physically intuitive meaning. To accomplish stable straight vertical flight, controlling pitch and roll directly was not sufficient, and therefore, additional terms were necessary to account for the asymmetric nature of the wing drag force affect on pitch and roll moments. Stable hover was then achieved by adding direct altitude control and indirect position control. While the final experiments demonstrated a long and stable flight, a future model-based controller, based on system identification techniques, could offer better flight performance. One advantage of the proposed control strategy and its associated model-free controller design method is that the resulting control structures could potentially be easily applied to a wide gamut of flying robots. We believe that the results presented in this paper represent a significant step towards the development of a future robustly-stable model-based control strategy.

References

1. Finio, B.M.: Roll, Pitch and Yaw Torque Control for a Robotic Bee. Ph.D, Dissertation, Harvard University, Cambridge, MA (2012)
2. Vigoreaux, J.O.: Nature's Versatile Engine: Insect Flight Muscle Inside and Out. Landes Bioscience & Springer, Georgetown, TX and New York, NY (2006)
3. Pérez-Arancibia, N.O., Chirarattananon, P., Finio, B.M., Wood, R.J.: Pitch-angle feedback control of a biologically inspired flapping-wing microrobot. In: Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics, pp. 1495–1502. Phuket Island, Thailand (2011)
4. Pérez-Arancibia, N.O., Ma, K.Y., Galloway, K.C., Greenberg, J.D., Wood, R.J.: First controlled vertical flight of a biologically inspired microrobot. Bioinspir. Biomim. **6**(3), 036009–1–11 (2011)
5. Finio, B.M., Pérez-Arancibia, N.O., Wood, R.J.: System identification and linear time-invariant modeling of an insect-sized flapping-wing micro air vehicle. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1107–1114. San Francisco, CA (2011)
6. Doman, D.B., Oppenheimer, M.W., Sigthorsson, D.O.: Wingbeat shape modulation for flapping-wing micro-air-vehicle control during hover. J. Guid. Control. Dyn. **33**(3), 724–739 (2010)
7. Ma, K.Y., Felton, S.M., Wood, R.J.: Design, fabrication, and modeling of the split actuator microrobotic bee. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1133–1140. Vilamoura, Algarve, Portugal (2012)
8. Ma, K.Y., Chirarattananon, P., Fuller, S.B., Wood, R.J.: Controlled flight of a biologically inspired, insect-scale robot. Science **340**(6132), 603–607 (2013)
9. Ellington, C.P.: The aerodynamics of hovering insect flight. I. The quasi-steady analysis. Phil. Trans. R. Soc. Lond. B **305**(1122), 1–15 (1984)
10. Alexander, D.E.: Nature's Flyers. The Johns Hopkins University Press, Baltimore, MD (2002)
11. Wood, R.J.: The first takeoff of a biologically inspired at-scale robotic insect. IEEE Trans. Robot **24**(2), 341–347 (2008)
12. G. R. Spedding: The aerodynamics of flight. In: Alexander, R. McN. (eds.) Mechanics of Animal Locomotion, pp. 51–111. Springer-Verlag, Berlin & Heidelberg, Germany (1992)
13. Ellington, C.P.: The aerodynamics of hovering insect flight. II. morphological parameters. Phil. Trans. R. Soc. Lond. B **305**(1122), 17–40 (1984)
14. Ellington, C.P.: The aerodynamics of hovering insect flight. VI. lift and power requirements. Phil. Trans. R. Soc. Lond. B **305**(1122), 145–185 (1984)
15. Spedding, G.R., Lissaman, P.B.S.: Technical aspects of microscale flight systems. J. Avian Biol. **29**(4), 458–468 (1998)
16. Pesavento, U., Wang, Z.J.: Flapping wing flight can save aerodynamic power compared to steady flight. Phys. Rev. Lett. **103**(11), 118102 (2009)
17. Wood, R.J., Steltz, E., Fearing, R.S.: Nonlinear performance limits for high energy density piezoelectric bending actuators. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 3633–3640. Barcelona, Spain (2005)
18. Pérez-Arancibia, N.O., Whitney, J.P., Wood, R.J.: Lift force control of a flapping-wing microrobot. In: Proceedings of the 2011 American Control Conference, pp. 4761–4768. San Francisco, CA (2011)
19. Pérez-Arancibia, N.O., Whitney, J.P., Wood, R.J.: Lift force control of flapping-wing microrobots using adaptive feedforward schemes. IEEE/ASME Trans. Mechatron **18**(1), 155–168 (2013)
20. Whitney, J.P., Wood, R.J.: Aeromechanics of passive rotation in flapping flight. J. Fluid Mech. **660**, 197–220 (2010)
21. Whitney, J.P.: Design and Performance of Insect-Scale Flapping-Wing Vehicles. Ph.D. Dissertation, Harvard University, Cambridge, MA (2012)
22. Lehmann, F.-O., Dickinson, M.H.: The control of wing kinematics and flight forces in fruit flies (Drosophila spp.) J. Exp. Biol. **201**(3), 385–401 (1998)
23. Weis-Fogh, T.: Quick estimates of flight fitness in hovering animals, including novel mechanisms for lift production. J. Exp. Biol. **59**(1), 169–230 (1973)
24. Laub, A.J.: Matrix Analysis for Scientists and Engineers. SIAM, Philadelphia, PA (2004)
25. Pérez-Arancibia, N.O., Duhamel, P.-E.J., Ma, K.Y., Wood, R.J.: Model-Free Control of a Hovering Flapping-Wing Flying Microrobot. [Online]. Available: http://micro.seas.harvard.edu/JINT2014/S1.mp4 (2014)