Journal of Intelligent & Robotic Systems (2022) 105: 19
https://doi.org/10.1007/s10846-021-01556-2

A Multiplatform Position Control Scheme for Flying Robotic Insects

Ryan M. Bena[1] · Xuan-Truc Nguyen[1] · Xiufeng Yang[2] · Ariel A. Calderón[3] · Ying Chen[3] · Néstor O. Pérez-Arancibia[1,4]

Received: 28 May 2020 / Accepted: 25 November 2020 / Published online: 11 May 2022

Abstract

We present a multiplatform control architecture to enable stable hovering flight and trajectory tracking for insect-scale *flapping-wing micro air vehicles* (FWMAVs). In the proposed approach, flight controllers are synthesized by using modern design techniques and applying the quaternion representation to describe the attitude dynamics of the controlled FWMAVs. During controller synthesis, the nominal stability of the closed-loop system is analyzed and ensured employing nonlinear Lyapunov methods. Given stringent fabrication, payload, size, and actuation limits at the insect scale, all subgram FWMAVs developed to date have been underactuated and, therefore, operate under multiple constraints. This observation has motivated us to develop an approach in which attitude is directly controlled by generating roll, pitch, and yaw torques through asymmetrical flapping; however, positional control forces are generated by manipulating the attitude of the robot to align its body-fixed aerodynamic-thrust vector along a desired direction. By leveraging this concept, we endow the proposed control architecture with the desired multiplatform characteristic and flexibility required to achieve multiple different objectives. The functionality, performance, and suitability of the proposed control approach were demonstrated experimentally using two different FWMAVs: a new version of the RoboBee, a 75-mg two-wing robot originally developed at Harvard University; and the Bee^{++}, a 95-mg four-wing robot that we developed in the *Autonomous Microrobotic Systems Laboratory* (AMSL) at the *University of Southern California* (USC). The use of a single control architecture to fly both robots allows an objective comparison of their flight capabilities. The performance achieved by the Bee^{++} highlights the significant potential of its new design.

Keywords Artificial insects · Flapping-wing flyers · Microrobotic design and fabrication · Autonomous control systems · Nonlinear stability · Aerodynamic modeling

This work was partially supported by the National Science Foundation (NSF) through NRI Award 1528110; the University of Southern California Viterbi School of Engineering through fellowships to X.-T. Nguyen, A. A. Calderón, and Y. Chen; and a start-up fund to N. O. Pérez-Arancibia. Part of the material in this paper was presented at the 19th International Conference on Advanced Robotics (ICAR), December 2–6, 2019 in Belo Horizonte, Brazil.

✉ Ryan M. Bena
bena@usc.edu

✉ Xuan-Truc Nguyen
xuantruc.nguyen@usc.edu

✉ Xiufeng Yang
xiufeng@usc.edu

✉ Ariel A. Calderón
aacalder@usc.edu

✉ Ying Chen
chen061@usc.edu

✉ Néstor O. Pérez-Arancibia
n.perezarancibia@wsu.edu

1 Introduction

Since the early 2000s, the term *flapping-wing micro air vehicle* (FWMAV) has been used to describe the notion of insect-scale aerial robots that can use biologically-inspired techniques to generate thrust and lift for flight as eventually demonstrated in [1–8] and references therein. In this context, as examples of pioneering non-flying flapping-wing prototypes, we can include the robots in [9] and [10], which introduced transmission mechanisms that replicated some of the mechanical functions of the common blowfly (genus *Calliphora*). An early example of a flying robotic

[1] Department of Aerospace and Mechanical Engineering, University of Southern California (USC), Los Angeles, CA 90089-1453, USA

[2] Applied Materials, Santa Clara, CA 95054-3299, USA

[3] Intuitive Surgical Inc., Sunnyvale, CA 94086-5301, USA

[4] School of Mechanical and Materials Engineering, Washington State University (WSU), Pullman, WA 99164-2920, USA

insect is the ornithopter presented in [11], which has a butterfly-inspired design and is *passively* powered by the elastic potential energy stored in a rubber band. The first fully-functional flapping-wing flyers are the single-actuator two-wing fly-inspired robots in [12] and [13], which were respectively demonstrated to counteract gravity and hover under control but remained constrained to the vertical degree of freedom, and the two-actuator two-wing bee-inspired robot in [2–5] (the Harvard RoboBee), which can fly unconstrained using feedback control. Presently, the quest to develop a fully-autonomous subgram FWMAV is still underway. This line of research is of great interest because FWMAVs could significantly expand capabilities in the areas of terrestrial and extraterrestrial exploration, search and rescue, artificial pollination, reconnaissance, and countless others. Additionally, these biologically-inspired robots can serve as invaluable research tools to improve our scientific understanding of real insects.

The microscopic dimensions of FWMAVs generate many mechanical and electrical design challenges because most traditional techniques employed to develop larger robots do not translate well to the centimeter and millimeter scales. At these small sizes, designs must be optimized for minimal complexity and maximum efficiency without overly complicating the fabrication process. A very effective solution to the combined problem of microrobotic design and fabrication is the *smart composite microstructure* (SCM) method [14] and its later iterations [15], which enable the integration of *carbon fiber* (CF) structural components with flexible joints and piezoelectric actuators. This approach eliminates the need for intricate mechanisms and leverages high-precision laser micro-machining, thus enabling the physical realization of designs that can produce the necessary power and articulation for microrobotic flight. In fact, the SCM method was the key element in the development of the robots presented in [12] and [13], which eventually led to the creation of the 75-mg Harvard RoboBee [2–5] that employs two bimorph piezoelectric actuators to independently flap its two wings through mechanical transmissions.

Theoretically, despite being underactuated, two-wing flyers of the RoboBee class are fully controllable during unconstrained flight through the modulation of body-pitch and body-roll via asymmetrical flapping, and of body-yaw by employing split-cycle flapping [16]. In flight experiments, robots of this type are capable of adjusting the flapping patterns of each wing to control body-roll and body-pitch; however, they are fundamentally limited in their ability to produce body-yaw torques for control due to the narrow bandwidths of the piezoelectric-based mechanisms that map input voltages to output flapping motions. To address this issue, at the USC *Autonomous Microrobotic Systems Laboratory* (AMSL), we recently introduced a new 95-mg four-wing FWMAV, the Bee^{+} [8]. This robot uses four unimorph piezoelectric actuators to independently drive its four wings, thus substantially increasing the control authority of the system compared to that of its two-wing counterparts. Additionally, each wing of the Bee^{+} is installed with a preset inclination with respect to the plane defined by the leading edges of the four composing wings while at rest, which is the key design innovation that makes possible the experimental control of the robot's yaw *degree of freedom* (DOF). Although still biologically inspired, this design also draws ideas from the configuration and functionality of modern quadrotor *unmanned aerial vehicles* (UAVs), which similarly use four propellers to generate control forces and torques [17–21]. In the research presented in this article, we used a new advanced version of the four-wing Bee^{+}, recently developed at the AMSL, that we refer to as the Bee^{++} to avoid confusion with the original model presented in [8].

As the science and technology required to create increasingly better FWMAVs advance, a general methodology to synthesize and implement robust multiplatform flight controllers becomes necessary for two main reasons: (a) to objectively compare the flight capabilities of different FWMAV prototypes; and (b) to guide robotic design from a control perspective. The latter reason is relevant because the flight capabilities of an FWMAV depend on the position and velocity references that can be accurately tracked using feedback control. In the past few years, several control methods for FWMAVs, with varying degrees of complexity and success, have been proposed. For example, [3] discusses the development of a simple model-based *linear time-invariant* (LTI) controller that was employed to stabilize the three positional degrees of freedom, as well as the pitch and roll angles, of the first two-wing Harvard RoboBee prototype. Furthermore, [6] presents an adaptive controller for this same robot, which was developed to compensate for model-parameter uncertainty, internal noise, and environmental disturbances. Following a different approach, an intuitive model-free control architecture for two-wing robots of the RoboBee class is presented in [4, 5]. That controller was experimentally demonstrated to stabilize the tested robots during hovering flight; however, its local attitude formulation theoretically restricts nominal stability to a relatively small region of attraction. Recently, an LTI controller with a structure very similar to that in [3] was implemented to fly the cross-shaped four-wing FWMAV presented in [7]. This result indicates that generic multiplatform controllers for FWMAVs can be developed.

A common characteristic of all the controllers mentioned in the previous paragraph is that the positional control-force signal is generated by manipulating the attitude of the robot to point the body-fixed thrust vector, generated by the flapping wings, along a required direction to reach a desired position [3–7]. By leveraging this concept, we developed a control architecture that demonstrates the utility of a multiplatform

approach with the flexibility to accomplish numerous control objectives. A key advantage of this proposed scheme is its versatility, which is achieved by capitalizing on the similarities between many different FWMAVs to ensure broad applicability without eliminating valuable tuning and customization. Additionally, drawing from [8], we introduce a new approach that is more effective for controlling the yaw DOF of the Bee^{++} and similar four-wing flyers, thus facilitating future advances in unexplored domains of attitude control. For example, the design, implementation, and execution of aerobatic maneuvers.

The main contributions presented in this article are: (a) the introduction of a new four-wing microrobotic design, the Bee^{++}, which is capable of producing a maximum thrust force 2.3 times larger than that of the original Bee^{+} robot in [8]; (b) a detailed general mathematical description of the dynamics of the FWMAV class to which the RoboBee and Bee^{++} belong; (c) a comprehensive description of the functionality and detailed stability analysis of the proposed multiplatform control architecture; and (d) the presentation and analysis of new flight experimental results that were obtained using the RoboBee and new Bee^{++} platforms. The rest of the paper is organized as follows. To provide background, Section 2 briefly describes the robotic designs and fabrication processes of the two FWMAV platforms used in the research presented here. In particular, we highlight how robotic design determines the set of feasible control structures for the considered FWMAVs. Section 3 discusses the structure and stability of the proposed multiplatform position-control architecture. Section 4 presents results that demonstrate the functionality, stability, and performance of the multiplatform control method through hovering and tracking flight experiments. Lastly, Section 5 states conclusions about the broader impacts of the proposed controller approach, areas for improvement, and future research objectives.

Notation:

1) Scalars are represented by italic lowercase symbols, e.g., q; vectors are represented by bold lowercase symbols, e.g., $\boldsymbol{q}$; matrices are represented by bold uppercase symbols, e.g., $\boldsymbol{Q}$; and quaternions are represented by bold crossed lowercase symbols, e.g., $\boldsymbol{q}$.
2) The dot operator is used to denote time differentiation, e.g., $\dot{q} = \frac{dq}{dt}$. Consistently, additional dots represent higher order derivatives.
3) Throughout the paper, the variable t denotes time.
4) The symbol s represents the complex variable associated with the Laplace transform. Also, as customary, $\frac{1}{s}$ denotes the integrator operator.
5) The symbol $\times$ represents the vector cross-product operation and the symbol $\otimes$ represents the quaternion multiplication operation.
6) The symbol $\|\cdot\|_2$ denotes the Euclidean norm of a vector.
7) The symbol $[\,\cdot\,]^T$ denotes the transpose of a matrix.
8) The operator tr$\{\cdot\}$ computes the trace of a matrix.
9) The operator sgn$\{\cdot\}$ computes the sign of a scalar.
10) The symbols $>$, $<$, $\geq$, and $\leq$ denote ordering when used with scalars, and denote definiteness relationships when used with matrices.
11) The summation and product operations of multiple elements are represented by the $\sum$ and $\prod$ symbols, respectively.

2 Robotic Design and Fabrication

The class of FWMAVs considered in this paper is composed of flyers that flap their wings to generate yaw-axis-aligned thrust forces in order to fly according to the normal hovering mode as defined in [22]. Specifically, to analyze and experimentally test the proposed control architecture, we used a modified version of the Harvard RoboBee [3] and an upgraded USC Bee^{+} prototype [8] that we denominate as the Bee^{++}. These two FWMAVs are shown in Figs. 1(a) and (b), respectively. While these two robots share many common elements, they also have several unique design features.

2.1 Robotic Design of the RoboBee

The basic robotic design and components, the flapping mechanism, and one of the two bimorph actuators of a RoboBee prototype are depicted in Figs. 2(a), (b), and (c), respectively. This flying microrobot has a wingspan of 35 mm and a mass of 75 mg. Fig. 2 also defines the yaw, pitch, and roll rotations of the robot's body during flight. As customary, the term roll-pitch-yaw corresponds to the X-Y-Z rotation sequence with respect to an inertial frame and, consistently, the term yaw-pitch-roll denotes the z-y-x rotation sequence about the body-fixed axes defined in Section 3. Also, throughout the paper, we refer to these body-fixed axes as the roll, pitch, and yaw axes of the robot.

During flight, the rotational motion of each wing of a RoboBee prototype is characterized by three simultaneous modes: *flapping*, *pitching*, and *stroke-plane deviation*. Flapping is the rotation of a wing about the yaw axis of the robot and centered at the wing root, as depicted in Fig. 2(b). In this illustration, the instantaneous flapping-angle signal is denoted by φ. This flapping motion is actively and independently generated through a four-bar transmission mechanism that maps the corresponding actuator output δ to φ. As depicted in Fig. 2(c), for the RoboBee, the approximately-linear bidirectional output displacement δ is produced with a bimorph piezoelectric actuator. Depending on the instantaneous direction of rotation, the flapping

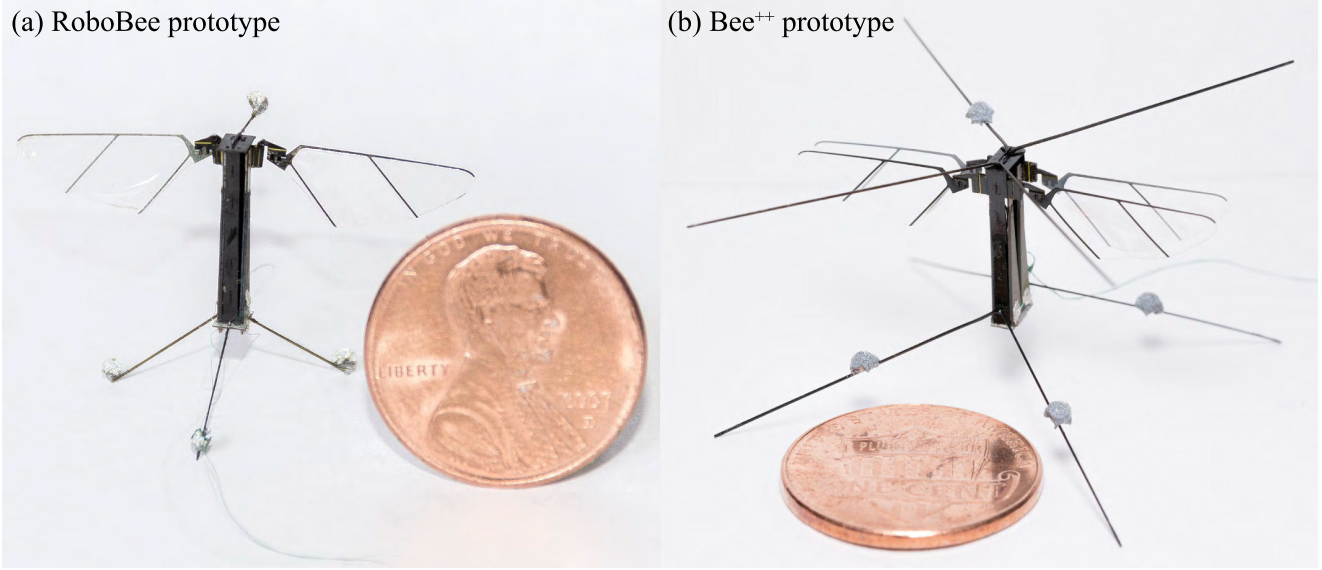

Fig. 1 Photographs of the RoboBee and Bee^{++} models used in the experiments. **(a)** A RoboBee prototype next to a US penny. This robot weighs 75 mg and has a wingspan of 35 mm. **(b)** A Bee^{++} prototype next to a US penny. This robot weighs 95 mg and has a wingspan of 33 mm. Before performing flight experiments, thin CF spars and reflective markers, which in total weigh about 5 mg, are attached to the robots. The spars provide collision protection and support the reflective markers that are used by a Vicon motion capture system to track the positions and attitudes of the tested FWMAVs

motion of a wing can be split into upstroke and downstroke phases. For the purpose of notational accuracy, in Section 3, the flapping angle of the wing on the right is labeled as φ_1 and the corresponding actuator output as δ_1. Consistently, the flapping angle of the wing on the left is labeled as φ_2 and the corresponding actuator output as δ_2

Pitching is the rotation of a wing about its leading edge, which is parallel to the pitch axis of the robot during rest. This rotation is produced passively by the interaction of the wing with the surrounding air and is enabled by a flexible hinge that connects the wing with the transmission mechanism, as shown in Figs. 2(a) and (b). Each hinge can be modeled as a rotational spring as described in [24, 25]. Lastly, stroke-plane deviation is defined as the rotation of a wing about the roll axis of the robot. A nonzero stroke-plane deviation results in a flapping stroke that is not in a plane parallel to that defined by the roll and pitch axes of the robot. In the RoboBee, stroke-plane deviations are caused by undesired deformations of the robotic structure and, therefore, correspond to disturbances that affect the closed-loop dynamics of the system.

In the RoboBee design, each of the two bimorph piezoelectric actuators, with a total weight of 25 mg, consists of two layers of PZT ceramic material (depicted in blue in Fig. 2(c)) and a center beam made of CF (depicted in black in Fig. 2(c)). This bilayer configuration enables the actuators to generate positive and negative deflections with respect to the neutral position. The conversion of electrical energy into mechanical work is produced by synchronously applying voltages across the faces of the two piezoelectric layers of the actuator according to the method described in [13] and shown in Fig. 2(c). Here, the voltage across the upper piezoelectric layer is positive and held constant; the voltage across the bottom layer, labeled as e, is positive and time-varying. As discussed in Section 3, e is varied to modulate δ. This excitation method allows for precise control of the induced bending of each actuator and, thus, the instantaneous flapping motion of each wing, φ, through a four-bar transmission mechanism, as already described above. For notational accuracy, in Section 3, the time-varying voltage excitation on the right is labeled as e_1 and the one on the left is labeled as e_2.

The passive wing-pitching response during flapping partly depends on the geometric design and material properties of the hinge that connects the wing to the transmission; further discussions of these topics are presented in [24, 25]. All the functional components of the robot are mechanically connected and structurally supported by a CF airframe, as depicted in Fig. 2(a). To ensure that the tested RoboBee prototype can generate sufficient thrust to fly under feedback control, we conducted thrust-force-generation experiments in an iterative design and fabrication process, using a sensor developed specifically to measure forces at the scale of this flyer [26]. The resulting microrobot can produce a maximum thrust force of approximately 1.34 mN, a value very similar to the 1.36 mN reported for the original Harvard prototype [3].

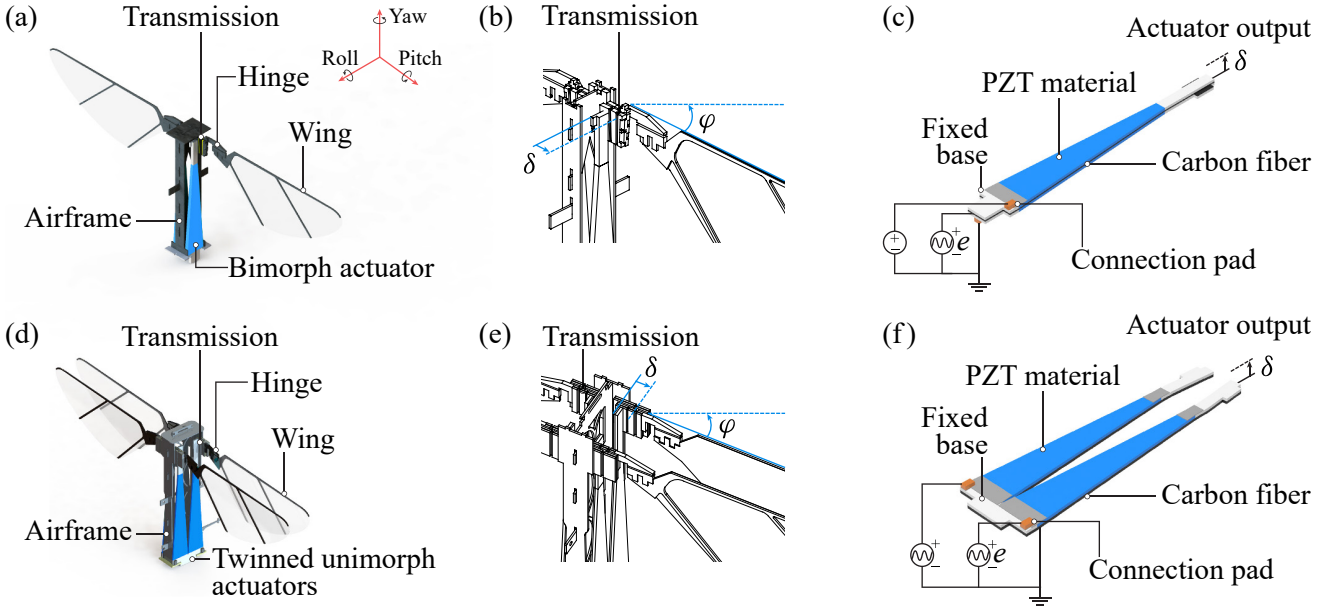

Fig. 2 Schematic diagrams of the RoboBee and Bee^{++} designs. **(a)** Diagram of the two-wing two-actuator RoboBee design showing the main components of the system. **(b)** Transmission mechanism used to map the instantaneous approximately-linear displacement generated by a bimorph piezoelectric actuator of the RoboBee, δ, into the corresponding instantaneous flapping angle, φ. **(c)** Illustration of a bimorph piezoelectric actuator used to drive one of the two wings of a RoboBee prototype. This actuator has an upper piezoelectric layer (shown in blue), a middle CF structural layer (shown in black), and a bottom piezoelectric layer (shown in blue). During assembly, two bimorph actuators are attached to the airframe of the microrobot by their bases. Enabled by the two-layer configuration of the bimorph design, the output δ of each actuator is bidirectional. That is, in terms of this illustration, the bimorph actuator can deflect both upward and downward. In this case, the two piezoelectric layers of each actuator are electrically excited according to the *bimorph with simultaneous drive* configuration described in [23], with e denoting the corresponding positive exciting voltage used for control as specified by Eq. 11. **(d)** Diagram of the four-wing four-actuator Bee^{++} design showing the main components of the system. **(e)** Transmission mechanism used to map the instantaneous approximately-linear displacement generated by a unimorph piezoelectric actuator of the Bee^{++}, δ, into the corresponding instantaneous flapping angle, φ. **(f)** Illustration of a pair of twinned unimorph actuators used to drive two of the four wings of a Bee^{++} prototype. Each unimorph actuator has an upper piezoelectric layer (shown in blue) and a bottom CF structural layer (shown in black). During assembly, the two pairs of twinned unimorph actuators are attached to the airframe of the robot by their bases. Given the one-layer configuration of the unimorph design, the output δ is unidirectional. Namely, in terms of this illustration, the unimorph actuators can only deform in the upward direction. In the unimorph case, the piezoelectric layer of each actuator is electrically excited according to the *alternating drive* configuration described in [8], with e denoting the corresponding positive exciting voltage used for control as specified by Eq. 11

2.2 Robotic Design of the Bee^{++}

The development of insect-scale four-wing flying robots has been a long-standing challenge in the field of microrobotics. The main difficulty that arises in the design of this type of flyer is that each additional independently-controlled wing requires the integration of another actuator into the robot. Since conventional microscale actuators are relatively heavy, adding more wings, and therefore more actuators, typically increases the design complexity without improving the thrust-to-weight ratio. The first successful subgram four-wing design was presented in [7], which is driven by four independent bimorph piezoelectric actuators. That robot weighs 143 mg, can achieve a maximum thrust-to-weight ratio of 2.8, and has been demonstrated to have good controllability characteristics; however, it is almost twice as heavy as a standard RoboBee prototype and its cross-shaped configuration creates a rather large convex hull.

Following a different approach, we developed the four-wing four-actuator Bee^{+}, whose design and fabrication process were first presented in [8] and further discussed in [27]. This robot is the same size as the RoboBee and only 27 % heavier, with a mass of 95 mg. A typical Bee^{+} prototype has a wingspan of 33 mm and exhibits an estimated maximum thrust-to-weight ratio of 1.5. The key innovation that made the physical realization of the Bee^{+} design possible is a new type of unimorph piezoelectric actuator that weighs only 14 mg. Actuators of this class are fabricated in pairs of twins in order to facilitate the manufacturing process. The Bee^{++} design used in the research discussed here has the same mass as the Bee^{+}; however, due to several key design improvements, this robot exhibits an estimated maximum thrust-to-weight ratio of 3.4. The corresponding CAD design, four-actuator flapping mechanism, and an illustration of a pair of twinned unimorph actuators used to drive the robot are depicted

in Figs. 2(d), (e), and (f), respectively. As in the RoboBee case, φ denotes the flapping angle of a wing and δ denotes the output displacement generated by the corresponding unimorph actuator. Also, e denotes the time-varying voltage applied across the single piezoelectric layer of the device. In Section 3, we use the symbols φ_i, δ_i, and e_i, for $i \in \{1, 2, 3, 4\}$, to denote the front-right, rear-right, front-left, and rear-left variables, respectively. The yaw, pitch, and roll rotations of the robot's body, and the corresponding axes, are defined exactly as in the case of the RoboBee (Section 2.1).

From both the flight performance and controllability perspectives, the Bee^{++} is significantly superior to any other flying robot at its scale (below 100 mg). Specifically, three key elements endow the Bee^{++} design with unique flight capabilities. The first element is that the four wings of this robot are driven independently by two pairs of twinned unimorph piezoelectric actuators. This innovation in actuation is what makes possible the integration of four independent actuators into the structure of the robot. Each unimorph actuator is composed of only one active layer of piezoelectric ceramic, which significantly increased both the measured thrust-to-weight ratio and actuation capabilities of the robot compared to those of the RoboBee, while the total weight of the actuators was increased by only 12 % (from 50 to 56 mg). As a direct consequence, the control authority and maneuverability of the Bee^{++} are unmatched by any other FWMAV. The second element is that the combination of the four-wing configuration of the robot and the flapping mode shown in Fig. 2(e) significantly damps the rotational disturbances that affect the yaw motion of the flyer, as supported by quasi-steady aerodynamic analyses [8]. The third element is that, since the Bee^{++} design has twice the number of wings as its RoboBee counterpart, the aerodynamic loading acting on each wing during hovering flight is significantly lessened, thus improving the overall life expectancy of Bee^{++} FWMAVs because the time required for the hinge mechanisms to fatigue is extended.

With respect to the original Bee^{+} in [8], the enhanced Bee^{++} design features three design modifications aimed to improve lift generation and flight performance. The first modification involves the wing profile, which was changed as depicted in Figs. 3(a) and (b). In this case, by adjusting the wing shape and location of the wing root, the moments of inertia about the flapping and pitching axes were reduced from $45.9\,\mathrm{mg \cdot mm^2}$ to $36.7\,\mathrm{mg \cdot mm^2}$ and from $2.3\,\mathrm{mg \cdot mm^2}$ to $2.1\,\mathrm{mg \cdot mm^2}$, respectively. The second modification involves the mapping ratio between δ and φ of the transmission-flapping mechanism, which was decreased from $3509\,\mathrm{rad \cdot m^{-1}}$ to $2694\,\mathrm{rad \cdot m^{-1}}$ by increasing the thickness of the intermediate CF layer, l, in the four-bar mechanism of each transmission, as depicted in Fig. 3(c).

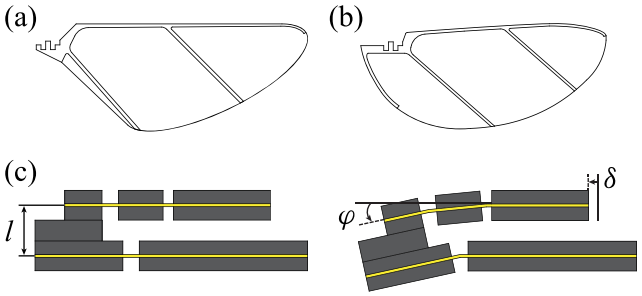

Fig. 3 Modifications of the Bee^{++} design with respect to the original Bee^{+} prototype. **(a)** Profile of the wing used by the original Bee^{+} prototype to fly. **(b)** Profile of the wing used by the new Bee^{++} prototype to fly. **(c)** Four-bar mechanism used to map the actuator output δ to the flapping angle φ in both FWMAVs considered in this paper. By increasing the parameter l from 285 µm to 371 µm, the transmission ratio from δ to φ was decreased from $3509\,\mathrm{rad \cdot m^{-1}}$ to $2694\,\mathrm{rad \cdot m^{-1}}$

The third modification involves the hinges that enable the passive pitching of the four wings of the robot, whose design was modified to have a stiffness of $2.3\,\mu\mathrm{N \cdot m \cdot rad^{-1}}$, which is about 1.64 times greater than the stiffness of the hinges used in the original Bee^{+} design in [8].

These three design modifications were motivated by the wing profile analysis in [28] and the results on dynamic modeling of piezoelectrically-driven flapping-wing robots presented in [29]. According to the logic and conclusions therein, these three modifications should significantly increase the overall stiffness of the actuation system, thus widening its bandwidth. We verified this prediction experimentally by measuring the bandwidths of the wing-driving subsystems that map the voltages that excite the actuators to the flapping amplitudes of the four wings of the robot. In those experiments, we used a Phantom Miro LAB310 high-speed camera and the experimental setup described in [26]. The experimental data indicate that, on average, the bandwidth of a typical driving subsystem used to flap a wing of the tested Bee^{++} prototype is about 165 Hz, which represents an improvement of more than 50 % over that of the original Bee^{+} design.

As a consequence of its wider actuation bandwidth, the Bee^{++} prototype used in the experiments presented in this paper can generate an estimated maximum cycle-averaged thrust force of 3.14 mN, which is about 2.3 times greater than the value of 1.37 mN produced by the original Bee^{+} prototype. Consistent with the time-averaged quasi-steady analyses in [8, 13, 24, 25] and references therein, for a sinusoidal flapping pattern, the improvement in cycle-averaged thrust generation is explained by the fact that the magnitude of the produced thrust force is directly proportional to both the square of the flapping amplitude and the square of the flapping frequency. Namely, the wide actuation bandwidth of the tested Bee^{++} robot enables larger flapping amplitudes at higher frequencies when compared

to those achieved by flyers of the RoboBee class and the original Bee^{+}, which results in larger thrust forces.

2.3 Fabrication

All the robotic components of the tested FWMAVs were fabricated using the SCM method according to the general procedure shown in Fig. 4. As seen in this illustration, the RoboBee and Bee^{++} prototypes are made in five steps. The airframes, transmissions, hinges, and wings of these two robots are manufactured using very similar processes, while the actuators of both robotic designs are sufficiently dissimilar to require different fabrication methods. As shown in Fig. 4(a), an airframe is micromachined from a 90-μm sheet of CF, laminated according to the 90-0-0-90° pattern (see Step 1 in Fig. 4(a)), using a *diode-pumped solid-state* ultraviolet laser (DPSS, Photonics Industries DCH-355-3) with a wavelength of 355 nm and a spot diameter of 10 μm. CF has an extremely-high strength-to-weight ratio, making it an ideal material to construct the basic structures of FWMAVs. As depicted in Fig. 4(b), the transmission and hinges of a prototype are laser cut from a premachined composite laminate made of Kapton polyimide film and CF sheets cured in the CF-Kapton-CF configuration (see Step 2 in Fig. 4(b)). The precursive CF sheets are manufactured by curing three layers of CF prepeg in the 90-0-90° lamination pattern (see Step 1 in Fig. 4(b)) and have a thickness of about 70 μm whereas the Kapton film has a thickness of 7.5 μm. The bonding between the CF laminated sheets and the Kapton film is achieved using sheets of adhesive (Dupont Pyralux FR), shown in light yellow in Fig. 4(b), during a cure cycle. As shown in Step 4 of Figs. 4(a) and (b), after the final release cut shown in Step 3 is completed, the resulting 2D parts are folded, glued, and reinforced to create the 3D airframes and transmission mechanisms of the robotic prototypes.

As shown in Fig. 4(c), wings are laser cut from premachined laminates made of CF and Mylar polyester film. The precursive CF sheet is manufactured by curing three layers of CF prepeg in the 0-135-0° lamination pattern (see Step 1 in Fig. 4(c)) and has a thickness of about 70 μm. Mylar film, with a thickness of 2.5 μm, is an ideal material to make the membranes for the wings of FWMAVs because it is extremely light yet strong enough to withstand high-frequency interaction with the surrounding fluid during flapping. CF spars provide structural support for the membranes and maintain a planar geometry. The bonding between the CF and Mylar layers is also achieved using a sheet of adhesive. During a cure cycle, pressure and heat are applied using an automatic hydraulic press. For curing transmissions, hinges, and wings, a pressure of 117 psi at 180 °C is applied for one hour. The precursory structural CF laminates used to make the parts shown in Figs. 4(a), (b), and (c) are fabricated by curing multiple layers of CF prepeg (see Step 1), according to the lamination patterns already described above, under a pressure of 45 psi at a temperature of 130 °C for two hours. Depending on the part being fabricated, the number and orientation of CF layers are chosen to maximize structural functionality. For airframes, transmissions, and hinges, CF provides structural integrity while the Kapton layers enable folding during fabrication and the creation of flexure joints that function as rotational springs during operation.

As shown in Fig. 4(d), a bimorph piezoelectric actuator for a RoboBee prototype is composed of two layers of excitable PZT material separated by a structural passive CF layer. In the fabrication process, first, two rectangular sheets with a thickness of 127 μm, each containing a piece of PZT ceramic material (shown in blue) and two pieces of alumina (shown in white), are bonded to a 100-μm layer of high-modulus CF prepreg through a first cure cycle. During this first curing procedure, a pressure of 15 psi at 180 °C is applied for two hours to the layers of material that are pin-aligned to form a stack (see Step 2 in Fig. 4(d)). Next, in a second cure cycle identical to the first (15 psi at 180 °C for two hours), the two faces of the cured stack are bonded to two additional premachined laminates of CF composite (with a thickness of 27 μm) and two outer premachined sheets of copper-clad FR4 (with a thickness of 137 μm) in order to strengthen the PZT-alumina interfaces at the tip and base of the actuator, and facilitate electrical connection, respectively. Last, the bimorph actuators are released using the laser cutter (see Step 3 in Fig. 4(d)).

As shown in Fig. 4(e), a pair of unimorph piezoelectric actuators for a Bee^{++} prototype is composed of one layer of excitable PZT material and one structural passive CF layer. In the fabrication process, first, a single rectangular sheet with a thickness of 127 μm, containing a flat piece of PZT ceramic material (shown in blue) and two pieces of alumina (shown in white), is bonded by the bottom to two 63-μm layers of high-modulus CF prepreg through a first cure cycle. As in the bimorph actuator case, a pressure of 15 psi at 180 °C is applied for two hours to the layers of material that are pin-aligned to form a stack (see Step 2 in Fig. 4(e)). An additional bottom layer of alumina serves as a substrate that maintains the flatness of the stack; between this piece of alumina and the double layer of CF, we place a sheet of release film in order to prevent undesired bonding. Next, in a second cure cycle defined by the same parameters of heat, pressure, and time as those used in the first procedure (15 psi at 180 °C for two hours), the cured stack is bonded by the upper face to an additional premachined laminate of CF composite (with a thickness of 27 μm) and an outer premachined sheet of copper-clad FR4 (with a thickness of 137 μm) in order to strengthen the tip and base of the actuator, and facilitate electrical connection, respectively.

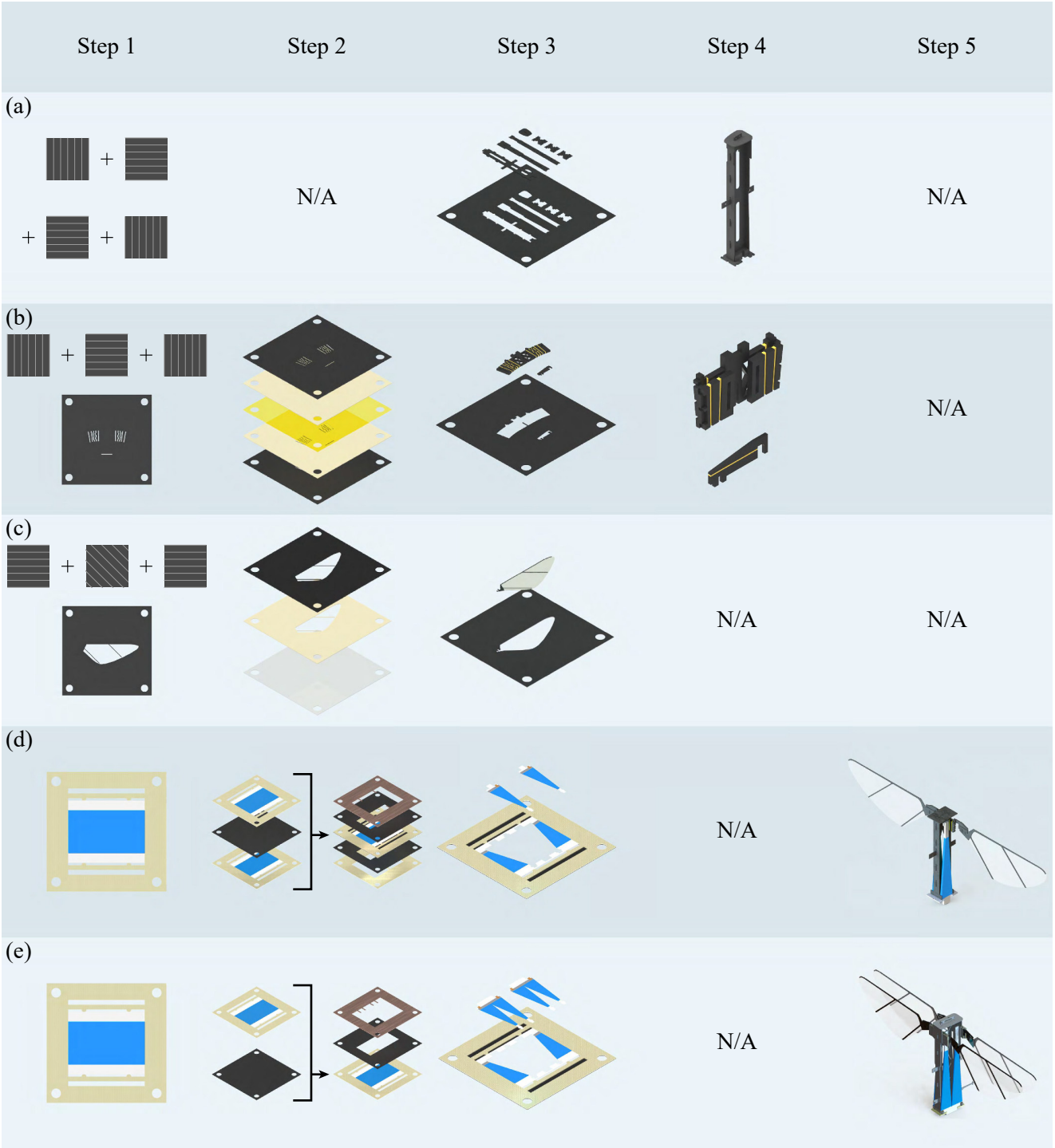

Fig. 4 Fabrication of the main robotic parts that compose the RoboBee and Bee^{++} prototypes. Excepting the actuators, the principal components of both robotic models are manufactured through very similar procedures. In general, a prototype is built in five steps. In **Step 1**, multilayer laminates are manufactured through the application of pressure and heat, or by assembling matching premachined pieces of material previously cut using a DPSS laser. In the fabrication of transmissions, hinges, and wings, additional permanent assembly features are laser cut in this step. In **Step 2**, the laminates fabricated in Step 1 are pin-aligned in stacks that are cured using heat and pressure to create 2D featured composites. In **Step 3**, 2D parts are released from the composites manufactured in Step 2, using a DPSS laser. In **Step 4**, the 2D parts manufactured in Step 3 are folded, glued, and reinforced to create the 3D components of the robotic prototypes. In **Step 5**, the 3D components manufactured in Step 4 are assembled to create the robotic prototypes. As examples, **(a)**, **(b),** and **(c)** depict the fabrication processes of an airframe, a transmission, and a wing for a RoboBee prototype. The fabrication processes of a bimorph piezoelectric actuator for a RoboBee prototype and a pair of unimorph piezoelectric actuators for a Bee^{++} prototype are depicted in **(d)** and **(e)**, respectively

Last, the pairs of twinned unimorph actuators are released using the laser cutter (see Step 3 in Fig. 4(e)). Further details on the design and fabrication of unimorph actuators are presented in [8].

In the case of both FWMAV designs, once all of the parts are fabricated, these are manually assembled under a microscope to create the robots (see Step 5 in Figs. 4(d) and (e)), using built-in aligning features to ensure proper placement. The pieces that compose an airframe have tabs and slots for precise assembly while the actuators, transmissions, hinges, and wings have mating protrusions and grooves to facilitate positioning. Last, long thin spars, cut from a single sheet of CF, are installed to provide ground stability on flat surfaces and protection in the event of a collision, as these spars can function as shock absorbers. Also, as seen in Fig. 1, the spars are used to support the reflective markers required by the Vicon motion capture system to track the motions of the tested FWMAVs during flight.

3 Controller Design

Feedback control is essential for FWMAVs to achieve sustained stable flight, as extensively discussed in [1–8] and references therein. Although liftoff can be accomplished using open-loop actuation, the nonlinear dynamics of all known FWMAVs are inherently unstable, which prevents sustained flight without the use of feedback control. Moreover, geometrical imperfections introduced during the fabrication process and disturbances generated by the local airflow surrounding the flyers cause unavoidable and unpredictable destabilizing forces and torques. Only through the use of precise sensors, quick-responding actuators, and an effective control architecture can FWMAVs overcome the effects of destabilizing forces and torques in order to accomplish complex control objectives. Overall, the control strategy presented in this paper uses a two-phase model-based approach to enforce rotational and positional stability. First, the instantaneous measured and reference position signals of the controlled robot are processed by an LTI control algorithm to determine the thrust-force vector required to reach a desired position in space. Next, taking into account that the class of flyers considered in this paper can only generate thrust forces along their yaw axes, the information about the thrust-force vector, along with that of the instantaneous desired yaw angle of the robot, is used to determine the attitude required to point the flyer along the direction of the previously-computed thrust-force reference. Then, this desired attitude and the measured attitude of the robot, in the form of unit quaternions, are processed by a second control algorithm that generates the torque signal used for attitude control. Last, this control signal and the magnitude of the thrust-force reference generated by the position controller are exerted on the system using the piezoelectrically-driven wings of the robot.

3.1 Dynamic Modeling of the Flying Microrobots

For the purposes of controller synthesis and simulation, a simple rigid-body model is sufficiently accurate to represent the dynamics of the robotic insects to be controlled. As discussed in Section 3.2, the intricacies of force and torque generation are generally dissimilar for different FWMAVs. However, the role of forces and torques as the inputs that excite the system dynamics can be described mathematically in the same manner for different flyers. Accordingly, we use identical models to represent the dynamics of both the RoboBee and Bee^{++}. First in the modeling process, we define the frames of reference and corresponding coordinate systems. The inertial frame, denoted by $\mathcal{N}$, is represented using the coordinate basis $\{\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3\}$ centered at a fixed point denoted by $\mathcal{O}_{\mathbf{0}}$. As shown in Fig. 5, the vector $\boldsymbol{n}_3$ is aligned with the vertical DOF. Also, a body-fixed reference frame, denoted by $\mathcal{B}$, is defined as shown in Fig. 5. The coordinate system associated with $\mathcal{B}$ is defined using the basis $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$, which is positioned at the *center of mass* (COM) of the modeled robot, denoted by $\mathcal{O}_{\mathcal{B}}$. Note that the vectors $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $\boldsymbol{b}_3$ correspond to the roll, pitch, and yaw axes of the modeled robotic flyer.

The translational motion of the modeled flying microrobot can be formulated, by invoking Newton's second law, as

$$m\ddot{\boldsymbol{r}} = f\boldsymbol{b}_3 - mg\boldsymbol{n}_3, \tag{1}$$

where m is the total mass of the robot; $\boldsymbol{r} = [r_1\ r_2\ r_3]^T$ is the position of the robot's COM relative to $\mathcal{O}_{\mathbf{0}}$; f is the magnitude of the cycle-averaged total thrust force generated by the flapping wings; and g is the acceleration due to gravity. Here, the thrust force generated by the modeled robot is assumed to be perfectly aligned with the direction of $\boldsymbol{b}_3$. That is, even though the direction of the instantaneous total aerodynamic force generated by each wing and the corresponding center of pressure are continuously shifting, the thrust model used to specify Eq. 1 assumes that the spatial center of force remains somewhere along the yaw axis of the robot and that the cycle-averaged aerodynamic total thrust remains aligned with this axis. Similarly, the rotational dynamics of the modeled robot is given by

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}, \tag{2}$$

where $\boldsymbol{J}$ is the inertia matrix of the robot, written with respect to $\mathcal{B}$; $\boldsymbol{\omega}$ is the angular velocity of $\mathcal{B}$ relative to $\mathcal{N}$; and $\boldsymbol{\tau} = [\tau_1\ \tau_2\ \tau_3]^T$ is the aerodynamic torque experienced by the robot due to flapping. Consistent with the definition

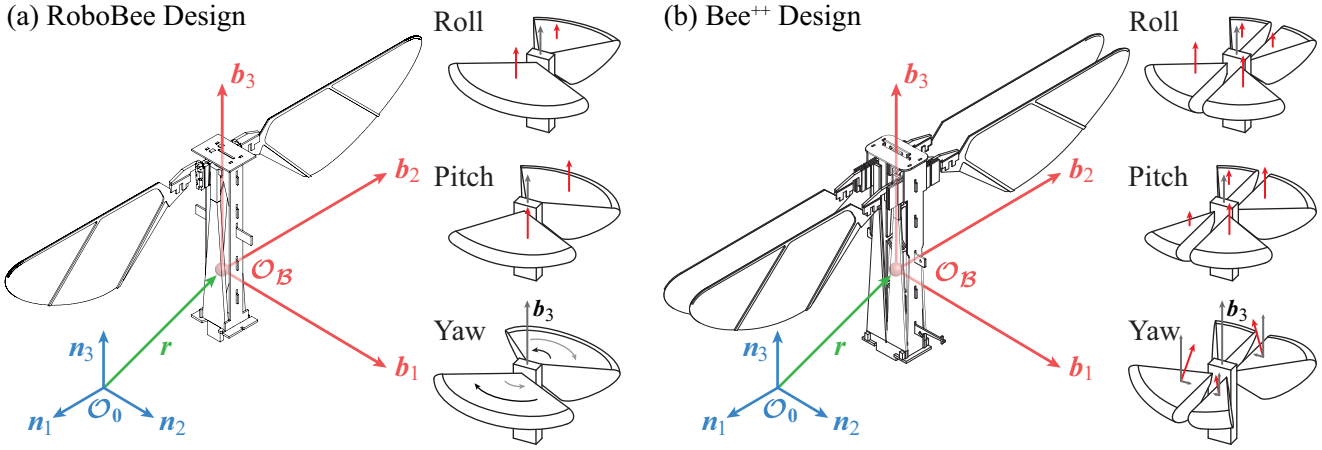

Fig. 5 Frames of reference and basic modes of flapping used to model the dynamics of RoboBee and Bee^{++} prototypes during flight. **(a)** The illustration on the left shows the RoboBee design and the schematics on the right show the modes of flapping used to generate torques about the roll, pitch, and yaw axes. **(b)** The illustration on the left shows the Bee^{++} design and the schematics on the right show the modes of flapping used to generate torques about the roll, pitch, and yaw axes. In both cases, the inertial frame of reference, $\{\mathcal{O}_0, \boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3\}$, is shown in blue and the body-fixed frame of reference, $\{\mathcal{O}_\mathcal{B}, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$, is shown in red. The vector position of the modeled robot's COM, $\boldsymbol{r}$, is shown in green

of $\boldsymbol{J}$, the coordinate vectors $\boldsymbol{\omega}$ and $\boldsymbol{\tau}$ are written with respect to $\mathcal{B}$.

In the case considered here, to describe the rotational kinematics of the modeled flyer, it is convenient to use unit quaternions due to their compact representation and computational robustness when compared to matrix-based descriptions of attitude [30, 31]. A unit quaternion is a four-component one-column array that stores the attitude of one frame of reference with respect to another. This avoids issues that commonly arise when using representations based on Euler angles, such as *gimbal lock*. Specifically, consistent with Euler's rotation theorem [31], in space, any rotation or sequence of rotations of a frame about a fixed point is equivalent to a single rotation, with a magnitude Θ, about a fixed axis defined by a unit vector $\hat{\boldsymbol{u}}$ that intersects the fixed point of rotation. Namely, a quaternion $\boldsymbol{q}$ stores this rotation information as

$$\boldsymbol{q} = \begin{bmatrix} \cos\frac{\Theta}{2} \\ \hat{\boldsymbol{u}}\sin\frac{\Theta}{2} \end{bmatrix}. \tag{3}$$

Accordingly, using quaternion algebra and assuming a perfectly-rigid body, the attitude dynamics of the modeled robotic insect can be described as

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{p}, \tag{4}$$

where $\boldsymbol{q}$ represents the orientation of the body-fixed frame relative to $\mathcal{N}$ and $\boldsymbol{p}$ is defined as

$$\boldsymbol{p} = \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}. \tag{5}$$

Thus, rearranging Eqs. 1, 2, and 4 yields a first-order nonlinear state-space representation with a total of thirteen states. Namely,

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{v}, \\ \dot{\boldsymbol{v}} &= \tfrac{f}{m}\boldsymbol{b}_3 - g\boldsymbol{n}_3, \\ \dot{\boldsymbol{q}} &= \tfrac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}, \\ \dot{\boldsymbol{\omega}} &= \boldsymbol{J}^{-1}\left(\boldsymbol{\tau} - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}\right), \end{aligned} \tag{6}$$

where $\boldsymbol{v}$ denotes the velocity of the robot's COM relative to $\mathcal{O}_0$.

3.2 Aerodynamic Force and Torque Generation

In the cases discussed here, the magnitude of the total thrust force, f, and the total aerodynamic-torque vector, $\boldsymbol{\tau}$, are both generated via the controlled flapping of the wings, which are attached to the transmission mechanisms through flexible hinges that function as rotational springs, as explained in Section 2. Consistently, during flapping, the wings interact with the surrounding air and, as a result, pitch passively, thus generating lift and drag force distributions on the wing surfaces that directly depend on the magnitude of the instantaneous angle of attack and velocity of the local flow field [32]. The total instantaneous forces and torques acting on the modeled robot due to sinusoidal flapping have high-frequency components in the $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $\boldsymbol{b}_3$ directions. Fortunately, as discussed in [13] and [25], the dynamics specified by Eq. 6 operate as a multi-channel lowpass filter and, therefore, for the purposes of system modeling and controller synthesis, we can estimate f and $\boldsymbol{\tau}$ using quasi-steady analysis and cycle averaging.

Specifically, assuming a sinusoidal flapping pattern, the magnitude of the cycle-averaged aerodynamic lift force generated by the ith wing, $\bar{f}_{\mathrm{l},i}$, can be modeled as

$$\bar{f}_{\mathrm{l},i} = C_{\mathrm{l}}(\bar{\alpha}) \nu_i^2 \varphi_{0,i}^2 S_{\mathrm{w}}, \tag{7}$$

where $C_{\mathrm{l}}(\bar{\alpha})$ is a lumped cycle-averaged aerodynamic lift-force coefficient that depends on the cycle-averaged angle of attack for the nominally dimensioned wing and hinge of the corresponding modeled robot, $\bar{\alpha}$; ν_i is the flapping frequency of the ith flapping wing and $\varphi_{0,i}$ is the amplitude of the sinusoidal flapping pattern $\varphi_i(t) = \varphi_{0,i} \sin(\nu_i t)$, corresponding to the ith wing, for $i \in \{1, 2\}$ in the case of the RoboBee and $i \in \{1, 2, 3, 4\}$ in the case of the Bee^{++}; and S_{w} is the total nominal area of a wing corresponding to the modeled robot. Further details can be found in [8]. Note that the same quasi-steady analysis followed to derive Eq. 7 can be reproduced in the more-realistic case in which the flapping patterns are not perfectly sinusoidal but periodic, by simply using Fourier-series decomposition and considering the principal components of the signals.

Furthermore, also using quasi-steady analysis, the magnitude of the cycle-averaged aerodynamic damping force acting on the ith wing, $\bar{f}_{\mathrm{d},i}$, can be modeled as

$$\bar{f}_{\mathrm{d},i} = C_2(\bar{\alpha}) \varphi_{0,i} \nu_i \omega_{\mathrm{w},i} S_{\mathrm{w}} + C_3(\bar{\alpha}) \dot{\omega}_{\mathrm{w},i} S_{\mathrm{w}}, \tag{8}$$

where $C_2(\bar{\alpha})$ and $C_3(\bar{\alpha})$ are lumped cycle-averaged aerodynamic drag-force coefficients that depend on $\bar{\alpha}$, derived according to the analysis presented in [24, 25]; and $\omega_{\mathrm{w},i}$ is the magnitude of the yaw component of the ith wing's angular velocity. Note that as first discussed in [8], given the symmetrical configurations of the robots of the class considered here, the damping forces acting on a flyer as a whole are, to a significant extent, mutually cancelled. Consistently, for the purpose of modeling for controller synthesis, we can assume that the magnitude of the total thrust force generated by the ith wing of the modeled flyer, $\bar{f}_{\mathrm{t},i}$, has the form in Eq. 7 and that the corresponding force vector $\bar{\boldsymbol{f}}_{\mathrm{t},i}$ is approximately perpendicular to the corresponding wing's stroke plane. Also, it is reasonable to assume that $\bar{\boldsymbol{f}}_{\mathrm{t},i}$ acts at the cycle-averaged center of force of the corresponding wing, whose position relative to the robot's COM is denoted by $\boldsymbol{d}_i$. For a periodic and symmetric flapping pattern, $\boldsymbol{d}_i$ is determined by the midpoint of the flapping stroke, the wing's center of force, and the position of the wing root.

For prototypes of the RoboBee class, the flapping wings generate stroke planes that are parallel to the $\boldsymbol{b}_1$-$\boldsymbol{b}_2$ plane, as shown using the schematic flapping mode at the bottom-right in Fig. 5(a); therefore, $\bar{\boldsymbol{f}}_{\mathrm{t},i}$, for $i \in \{1, 2\}$, is approximately aligned with the yaw axis, $\boldsymbol{b}_3$, of the flyer. On the other hand, the Bee^{++} was designed to generate wingstroke planes with angular offsets with respect to the $\boldsymbol{b}_1$-$\boldsymbol{b}_2$ plane, as explained using the schematic flapping mode at the bottom-right in Fig. 5(b). This results in the ability of each wing to generate a force component that is parallel to the $\boldsymbol{b}_1$-$\boldsymbol{b}_2$ plane in order to actively control the yaw DOF of the robot. We refer to this approach as the *inclined stroke plane* (ISP) method, which was first discussed in [8]. For the two types of flyers considered in this article, the total cycle-averaged thrust force aligned with the yaw axis and acting at the robot's COM can be estimated as

$$f = \sum_{i=1}^{n} \bar{\boldsymbol{f}}_{\mathrm{t},i}^{T} \boldsymbol{b}_3, \tag{9}$$

where n is the total number of wings of the modeled FWMAV. Consistently, the total cycle-averaged torque applied about the robot's COM can be estimated as

$$\boldsymbol{\tau} = \sum_{i=1}^{n} \boldsymbol{d}_i \times \bar{\boldsymbol{f}}_{\mathrm{t},i}. \tag{10}$$

From Eqs. 7, 9, and 10, it immediately follows that by varying the amplitude or frequency of the flapping pattern of each wing independently, the values of $\bar{f}_{\mathrm{t},i}$ and $\boldsymbol{d}_i$, and thus f and $\boldsymbol{\tau}$, can be varied. While the relationship between $\bar{f}_{\mathrm{t},i}$ and f is straightforward, more complex coordination is required to produce the desired components of $\boldsymbol{\tau}$.

For a two-wing FWMAV, torque about the roll axis, $\boldsymbol{b}_1$, of the flyer is generated by adjusting the difference between the values of $\bar{f}_{\mathrm{t},1}$ and $\bar{f}_{\mathrm{t},2}$, as explained using the schematic flapping mode at the top-right in Fig. 5(a). On the other hand, torque about the pitch axis, $\boldsymbol{b}_2$, of the flyer is generated by shifting the mean flapping angles of the wings to move the points of application of $\bar{\boldsymbol{f}}_{\mathrm{t},1}$ and $\bar{\boldsymbol{f}}_{\mathrm{t},2}$ out of the $\boldsymbol{b}_2$-$\boldsymbol{b}_3$ plane, as depicted using the schematic flapping mode at the middle-right in Fig. 5(a). Theoretically, torque about the yaw axis, $\boldsymbol{b}_3$, of the controlled flyer can be produced using a technique referred to as split cycling, according to which the flapping speed during the upstroke (in black) versus that of the downstroke (in gray) is varied and vice versa [2], as explained using the schematic flapping mode at the bottom-right in Fig. 5(a). However, thus far, effective experimental implementation of this method has been greatly limited by the narrow bandwidths of piezoelectric-based actuation mechanisms.

In the case of a four-wing FWMAV, roll torque is generated by adjusting the differences between the magnitudes of the $\bar{\boldsymbol{f}}_{\mathrm{t},i}$ forces produced by the wings located to the right and left sides of the $\boldsymbol{b}_1$-$\boldsymbol{b}_3$ plane, as explained using the schematic flapping mode at the top-right in Fig. 5(b). Similarly, pitch torque is generated by adjusting the differences between the magnitudes of the $\bar{\boldsymbol{f}}_{\mathrm{t},i}$ forces

produced by the wings to the front and rear sides of the $\boldsymbol{b}_2$-$\boldsymbol{b}_3$ plane, as explained using the schematic flapping mode at the middle-right in Fig. 5(b). As in the case of two-wing robots, the Bee^{++} can generate yaw torque via split cycling but also by using the ISP method mentioned above and discussed in [8], which is explained using the schematic flapping mode at the bottom-right in Fig. 5(b). Recent experimental results obtained at the AMSL using a Bee^{++} FWMAV have shown that the ISP method is significantly more effective than split cycling to control the yaw DOF because ISP-based torque generation is not limited by the narrow bandwidths of the piezoelectric actuation mechanisms typically used to drive FWMAVs.

3.3 Actuator Model

In general, when designing a controller for a system describable by Eq. 6, it is convenient to choose as control inputs f and $\boldsymbol{\tau}$; however, neither the RoboBee nor the Bee^{++} can generate these signals directly. For both microrobots, the true inputs to the modeled dynamical system are the oscillatory voltages used to drive the piezoelectric-based flapping mechanisms. The mapping between these voltages and the generalized forces generated by the flapping wings, f and $\boldsymbol{\tau}$, can be modeled using a linear static mapping with no dynamics or delays. In the case of the RoboBee, as depicted in Fig. 2(c), each bimorph actuator is excited with an instantaneous positive voltage signal of the form

$$e_i(t) = \gamma_i \sin(\nu_i t) + \epsilon_i, \tag{11}$$

where $i \in \{1, 2\}$; γ_i is the amplitude of the signal; ν_i is the frequency of the applied voltage and, due to linearity, the flapping frequency of the corresponding wing; and ϵ_i is an offset that determines the mean value of the flapping angle and maintains the instantaneous value of $e_i(t)$ positive. Immediately from Eq. 7, it follows that both γ_i or ν_i can be used to modulate f and $\boldsymbol{\tau}$. However, in the control scheme discussed here, ν_i is changed only for the purpose of yaw-torque modulation via split cycling, while γ_i and ϵ_i are continually adjusted for control during flight. In theory, the appropriate values for ν_i during the upstroke and downstroke can be found through open-loop trimming flapping experiments.

Thus, considering Eqs. 7 and 11, we define the actuation mapping-matrix for a RoboBee prototype as

$$\begin{bmatrix} f(t) \\ \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix} = \begin{bmatrix} k_{\text{amp}} & 0 & 0 & 0 \\ 0 & k_{\text{roll}} & 0 & 0 \\ 0 & 0 & k_{\text{pitch}} & 0 \\ 0 & 0 & 0 & k_{\text{yaw}} \end{bmatrix} \begin{bmatrix} \beta_{\text{amp}}(t) \\ \beta_{\text{roll}}(t) \\ \beta_{\text{pitch}}(t) \\ \beta_{\text{yaw}}(t) \end{bmatrix}, \tag{12}$$

where k_{amp}, k_{roll}, k_{pitch}, and k_{yaw} are empirically-identified constants; $\beta_{\text{amp}}(t)$ denotes the mean between the amplitudes of the signals $e_1(t)$ and $e_2(t)$, $\frac{1}{2}[\gamma_1(t) + \gamma_2(t)]$; $\beta_{\text{roll}}(t)$ denotes the difference between the amplitudes of both exciting voltages, $[\gamma_1(t) - \gamma_2(t)]$; $\beta_{\text{pitch}}(t)$ denotes the mean between the offsets of the signals $e_1(t)$ and $e_2(t)$, $\frac{1}{2}[\epsilon_1(t) + \epsilon_2(t)]$; and $\beta_{\text{yaw}}(t)$ denotes the deviation from the preset frequency of flapping during split-cycle-based yaw-torque generation. Thus, inverting the matrix specified by Eq. 12 yields the mapping from control inputs to actuator inputs that is used to control RoboBee prototypes. Namely,

$$\begin{bmatrix} \beta_{\text{amp}}(t) \\ \beta_{\text{roll}}(t) \\ \beta_{\text{pitch}}(t) \\ \beta_{\text{yaw}}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{k_{\text{amp}}} & 0 & 0 & 0 \\ 0 & \frac{1}{k_{\text{roll}}} & 0 & 0 \\ 0 & 0 & \frac{1}{k_{\text{pitch}}} & 0 \\ 0 & 0 & 0 & \frac{1}{k_{\text{yaw}}} \end{bmatrix} \begin{bmatrix} f(t) \\ \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix}. \tag{13}$$

In agreement with Eqs. 12 and 13, for the implementation and execution of the flight controller introduced in the next sections, the information contained in $\beta_{\text{amp}}(t)$, $\beta_{\text{roll}}(t)$, $\beta_{\text{pitch}}(t)$, and $\beta_{\text{yaw}}(t)$ is used to compute, according to Eq. 11, the voltages $e_1(t)$ and $e_2(t)$ that excite the two microactuators of the the controlled robot.

In the case of the Bee^{++}, as shown in Fig. 2(f), each unimorph actuator is excited with a voltage signal that is also described by Eq. 11, for $i \in \{1, 2, 3, 4\}$, but only the amplitude, γ_i, is varied for control purposes while the flapping frequency, ν_i, and offset, ϵ_i, are held constant. Borrowing from the literature on quadrotors [17, 18, 21], we assume that the averaged magnitude of the thrust force generated by the ith wing can be approximated as $\bar{f}_{\text{t},i} = k_{\text{f}}\gamma_i$, where k_{f} is an empirically-identified lumped constant. Additionally, as illustrated in Fig. 5(b), yaw torques in the steering plane are generated using the ISP method. Consistently, we estimate the component of the ith aerodynamic force projected on the steering plane as $\bar{f}_{\text{s},i} = k_{\text{s}}\gamma_i$, where k_{s} is also an identified lumped constant. Thus, the resultant actuator mapping-matrix takes the form

$$\begin{bmatrix} f(t) \\ \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix} = \begin{bmatrix} k_{\text{f}} & k_{\text{f}} & k_{\text{f}} & k_{\text{f}} \\ -k_{\text{f}}d_1 & -k_{\text{f}}d_1 & k_{\text{f}}d_1 & k_{\text{f}}d_1 \\ k_{\text{f}}d_2 & -k_{\text{f}}d_2 & k_{\text{f}}d_2 & -k_{\text{f}}d_2 \\ k_{\text{s}}d_3 & -k_{\text{s}}d_3 & -k_{\text{s}}d_3 & k_{\text{s}}d_3 \end{bmatrix} \begin{bmatrix} \gamma_1(t) \\ \gamma_2(t) \\ \gamma_3(t) \\ \gamma_4(t) \end{bmatrix}, \tag{14}$$

where d_j, for $j \in \{1, 2, 3\}$, are the magnitudes of the components of the vector locations of the wings' centers of force written with respect to $\mathcal{B}$, $\boldsymbol{d}_i$, for $i \in \{1, 2, 3, 4\}$. From the mechanics perspective, the values d_j can be thought of

as the magnitudes of the leverage arms associated with the cycle-averaged forces generated by the flapping wings of the flyer. Note that all the vectors $\boldsymbol{d}_i$ are defined in terms of only the three values d_j, which is explained by the geometrical symmetry of the Bee^{++} design. As with the RoboBee, inverting the matrix in Eq. 14 provides the mapping from the Bee^{++} control inputs, f and $\boldsymbol{\tau}$, to the actuator inputs, i.e.,

$$\begin{bmatrix} \gamma_1(t) \\ \gamma_2(t) \\ \gamma_3(t) \\ \gamma_4(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{4k_\mathrm{f}} & -\frac{1}{4k_\mathrm{f}d_1} & \frac{1}{4k_\mathrm{f}d_2} & \frac{1}{4k_\mathrm{s}d_3} \\ \frac{1}{4k_\mathrm{f}} & -\frac{1}{4k_\mathrm{f}d_1} & -\frac{1}{4k_\mathrm{f}d_2} & -\frac{1}{4k_\mathrm{s}d_3} \\ \frac{1}{4k_\mathrm{f}} & \frac{1}{4k_\mathrm{f}d_1} & \frac{1}{4k_\mathrm{f}d_2} & -\frac{1}{4k_\mathrm{s}d_3} \\ \frac{1}{4k_\mathrm{f}} & \frac{1}{4k_\mathrm{f}d_1} & -\frac{1}{4k_\mathrm{f}d_2} & \frac{1}{4k_\mathrm{s}d_3} \end{bmatrix} \begin{bmatrix} f(t) \\ \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix}. \quad (15)$$

For implementing and executing the flight controller presented in the following sections, the values of $\gamma_1(t)$, $\gamma_2(t)$, $\gamma_3(t)$, and $\gamma_4(t)$ are used to construct, according to Eq. 11, the voltages e_i, for $i \in \{1, 2, 3, 4\}$, that excite the four microactuators of the controlled Bee^{++}.

3.4 Position Controller

The control architecture introduced in this paper, depicted in Fig. 6, can be employed to automatically fly both the RoboBee and Bee^{++}, and also any other comparably rigid FWMAV driven by similar actuation methods. As shown in the diagram of Fig. 6, the position controller block takes as inputs the position control error, $\boldsymbol{r}_\mathrm{e} = \boldsymbol{r}_\mathrm{d} - \boldsymbol{r}$, the measured attitude quaternion, $\boldsymbol{q}$, and the desired instantaneous yaw angle, ψ_d. In this scheme, $\boldsymbol{r}_\mathrm{d}$ and $\boldsymbol{r}$ are the reference and measured positions written with respect to $\mathcal{N}$, respectively. The outputs generated by the position controller are the magnitude of the total thrust command, f, and the desired attitude quaternion, $\boldsymbol{q}_\mathrm{d}$. The process of control signal generation is executed as follows. First, the algorithm computes the instantaneous thrust-force vector written with respect to the inertial frame, $\boldsymbol{f}_\mathrm{a}$, required to compel the robot to track a desired trajectory, according to the LTI law

$$\begin{aligned} \boldsymbol{f}_\mathrm{a}(t) \;=\; & \boldsymbol{K}_\mathrm{p}\left[\boldsymbol{r}_\mathrm{d}(t) - \boldsymbol{r}(t)\right] + \boldsymbol{K}_\mathrm{i}\int_0^t \left[\boldsymbol{r}_\mathrm{d}(\tau) - \boldsymbol{r}(\tau)\right] d\tau \\ & + \boldsymbol{K}_\mathrm{d}\left[\dot{\boldsymbol{r}}_\mathrm{d}(t) - \dot{\boldsymbol{r}}(t)\right] + m\ddot{\boldsymbol{r}}_\mathrm{d}(t) + mg\boldsymbol{n}_3, \end{aligned} \quad (16)$$

where $\boldsymbol{K}_\mathrm{p}$, $\boldsymbol{K}_\mathrm{i}$, and $\boldsymbol{K}_\mathrm{d}$ are diagonal positive-definite gain matrices. Here, the first three terms correspond to a standard *proportional-integral-derivative* (PID) control architecture. The fourth term produces a direct open-loop force corresponding to the desired trajectory in the absence of control errors and external forces, and the last term provides direct constant gravity compensation.

Due to the fixed stroke planes according to which their wings are flapped during flight, RoboBee and Bee^{++} prototypes generate thrust forces that remain approximately aligned with the directions of their yaw axes, $\boldsymbol{b}_3$. As a consequence, an arbitrary force vector $\boldsymbol{f}_\mathrm{a}$ can not be produced instantaneously. However, by maneuvering the attitude of the controlled robot, $\boldsymbol{b}_3$ can be aligned with the direction of $\boldsymbol{f}_\mathrm{a}$, computed as specified by Eq. 16. Specifically, to control the position of a flyer during flight, the real-time controller in Fig. 6 computes the magnitude of $\boldsymbol{f}_\mathrm{a}$, f, and generates and tracks the unit quaternion $\boldsymbol{q}_\mathrm{d}$ that contains the information of the attitude required to align the yaw axis of the body frame, $\boldsymbol{b}_3$, with the direction of $\boldsymbol{f}_\mathrm{a}$. First, to extract the magnitude of the yaw component of $\boldsymbol{f}_\mathrm{a}$, the control algorithm simply calculates the inner product

$$f = \boldsymbol{f}_\mathrm{a}^T \boldsymbol{b}_3. \quad (17)$$

Next, in agreement with Euler's rotation theorem [33], using the yaw-angle reference signal, ψ_d, and the direction of $\boldsymbol{f}_\mathrm{a}$, the coordinate basis vectors, $\{\boldsymbol{b}_{\mathrm{d},1}, \boldsymbol{b}_{\mathrm{d},2}, \boldsymbol{b}_{\mathrm{d},3}\}$, of the desired body-fixed frame are computed according to

$$\begin{aligned} \boldsymbol{b}_{\mathrm{d},3} &= \frac{\boldsymbol{f}_\mathrm{a}}{\|\boldsymbol{f}_\mathrm{a}\|_2}, \\ \boldsymbol{b}_{\mathrm{d},1} &= \frac{[-\sin\psi_\mathrm{d} \;\; \cos\psi_\mathrm{d} \;\; 0]^T \times \boldsymbol{b}_{\mathrm{d},3}}{\|[-\sin\psi_\mathrm{d} \;\; \cos\psi_\mathrm{d} \;\; 0]^T \times \boldsymbol{b}_{\mathrm{d},3}\|_2}, \\ \boldsymbol{b}_{\mathrm{d},2} &= \boldsymbol{b}_{\mathrm{d},3} \times \boldsymbol{b}_{\mathrm{d},1}. \end{aligned} \quad (18)$$

As long as the result of the $\boldsymbol{b}_{\mathrm{d},1}$ calculation is nonzero, these basis vectors are used to form the rotation matrix $\boldsymbol{S}_\mathrm{d} = [\boldsymbol{b}_{\mathrm{d},1} \; \boldsymbol{b}_{\mathrm{d},2} \; \boldsymbol{b}_{\mathrm{d},3}]$, which maps any vector written with respect to the desired frame of coordinates to $\mathcal{N}$. Furthermore, the information in $\boldsymbol{S}_\mathrm{d}$ is expressed as a single rotation Θ_d about a unit axis $\hat{\boldsymbol{u}}_\mathrm{d}$. The rotation angle is calculated as

$$\Theta_\mathrm{d} = \arccos\left(\frac{\mathrm{tr}\{\boldsymbol{S}_\mathrm{d}\} - 1}{2}\right). \quad (19)$$

And, consistently, the associated unit rotation axis, $\hat{\boldsymbol{u}}_\mathrm{d}$, is the normalized eigenvector of $\boldsymbol{S}_\mathrm{d}$ corresponding to the eigenvalue $\sigma = 1$. For $\Theta_\mathrm{d} \neq 0$, this eigenvector is given by

$$\hat{\boldsymbol{u}}_\mathrm{d} = \frac{1}{\sqrt{(3 - \mathrm{tr}\{\boldsymbol{S}_\mathrm{d}\})\,(1 + \mathrm{tr}\{\boldsymbol{S}_\mathrm{d}\})}} \begin{bmatrix} S_{\mathrm{d},32} - S_{\mathrm{d},23} \\ S_{\mathrm{d},13} - S_{\mathrm{d},31} \\ S_{\mathrm{d},21} - S_{\mathrm{d},12} \end{bmatrix}, \quad (20)$$

where $S_{\mathrm{d},ij}$ denotes the $\{i, j\}$-entry of $\boldsymbol{S}_\mathrm{d}$. Last, these values are used to generate the attitude-reference quaternion, $\boldsymbol{q}_\mathrm{d}$, according to

$$\boldsymbol{q}_\mathrm{d} = \begin{bmatrix} m_\mathrm{d} \\ \boldsymbol{n}_\mathrm{d} \end{bmatrix} = \begin{bmatrix} \cos\frac{\Theta_\mathrm{d}}{2} \\ \hat{\boldsymbol{u}}_\mathrm{d}\sin\frac{\Theta_\mathrm{d}}{2} \end{bmatrix}. \quad (21)$$

Note that this quaternion represents exactly the same attitude of the desired frame of reference with respect to $\mathcal{N}$ as specified by Eq. 18.

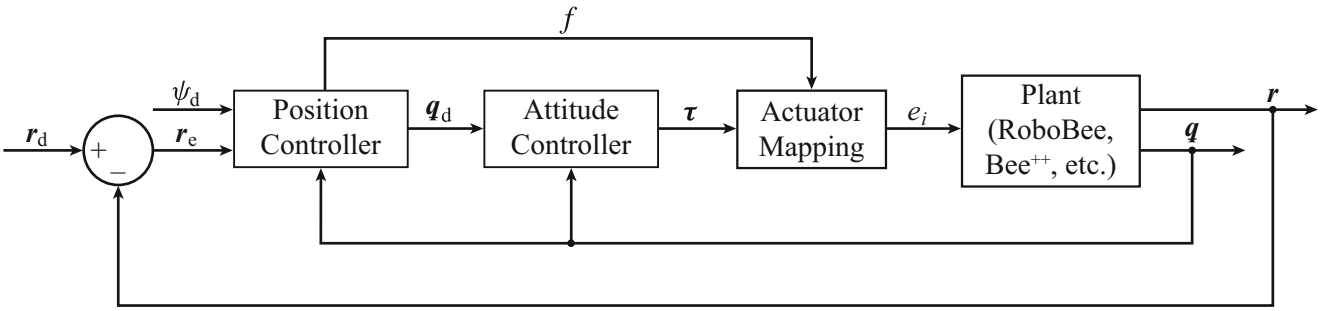

Fig. 6 Block diagram of the proposed control architecture for FWMAVs. In this scheme, the block labeled as *plant* represents the dynamics of the controlled FWMAV, which generates as outputs the position of the COM, $\boldsymbol{r}$, and the attitude quaternion of the body-fixed frame $\mathcal{B}$, $\boldsymbol{q}$. The inputs to this block are the actuator voltages e_i, with $i \in \{1, 2\}$ for the RoboBee and $i \in \{1, 3, 2, 4\}$ for the Bee^{++}. The *position controller* block receives as inputs the position control error $\boldsymbol{r}_\mathrm{e} = \boldsymbol{r}_\mathrm{d} - \boldsymbol{r}$, the desired yaw angle ψ_d, and the attitude quaternion $\boldsymbol{q}$. The outputs from this block are the total thrust-force magnitude f and the desired attitude quaternion $\boldsymbol{q}_\mathrm{d}$. The *attitude controller* block receives as inputs the measured and desired attitude quaternions, $\boldsymbol{q}$ and $\boldsymbol{q}_\mathrm{d}$, and produces as output the control torque $\boldsymbol{\tau}$. The *actuator mapping* block receives as inputs the control signals, f and $\boldsymbol{\tau}$, and produces as outputs the actuator voltages e_i

3.5 Attitude Controller

As shown in Fig. 6, the attitude controller takes as inputs the attitude-reference quaternion, $\boldsymbol{q}_\mathrm{d}$, and the quaternion $\boldsymbol{q}$, which represents the current measured attitude of the body-fixed frame with respect to the inertial frame. Using this information and quaternion algebra [31], the attitude controller algorithm first computes the attitude-error quaternion according to

$$\boldsymbol{q}_\mathrm{e} = \boldsymbol{q}^{-1} \otimes \boldsymbol{q}_\mathrm{d}, \tag{22}$$

where $\boldsymbol{q}_\mathrm{e} = [m_\mathrm{e} \ \boldsymbol{n}_\mathrm{e}^T]^T$ represents the attitude of the desired frame of reference relative to the body-fixed frame and $\boldsymbol{q}^{-1}$ is the quaternion inverse of $\boldsymbol{q}$ as defined in [31]. The terms m_e and $\boldsymbol{n}_\mathrm{e}$ respectively denote the scalar and vector components of the quaternion $\boldsymbol{q}_\mathrm{e}$, in agreement with the generic definition specified by Eq. 3. The direction of the vector $\boldsymbol{n}_\mathrm{e}$ is of particular importance because it represents the Euler rotation axis between the body-fixed frame and the desired frame of reference. After finding the error quaternion using Eq. 22, the control algorithm computes the angular velocity of the body-fixed frame relative to the inertial frame in body-fixed coordinates by inverting Eq. 4, which yields

$$\begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = 2\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}. \tag{23}$$

Similarly, the angular velocity of the desired frame relative to the inertial frame in the desired coordinates, $\hat{\boldsymbol{\omega}}_\mathrm{d}$, is given by

$$\begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}_\mathrm{d} \end{bmatrix} = 2\boldsymbol{q}_\mathrm{d}^{-1} \otimes \dot{\boldsymbol{q}}_\mathrm{d}, \tag{24}$$

which can be converted to body-fixed coordinates, using rotation matrices, according to

$$\boldsymbol{\omega}_\mathrm{d} = \boldsymbol{S}^T \boldsymbol{S}_\mathrm{d} \hat{\boldsymbol{\omega}}_\mathrm{d}, \tag{25}$$

where $\boldsymbol{S} = [\boldsymbol{b}_1 \ \boldsymbol{b}_2 \ \boldsymbol{b}_3]$ is the rotation matrix that converts vectors from body-fixed coordinates to inertial coordinates.

Last, utilizing m_e, $\boldsymbol{n}_\mathrm{e}$, $\boldsymbol{\omega}$, and $\boldsymbol{\omega}_\mathrm{d}$, the attitude controller calculates the torque in body-fixed coordinates required to compel the robot to track the desired attitude, specified by the quaternion $\boldsymbol{q}_\mathrm{d}$ in Eq. 21, using the LTI feedback law

$$\boldsymbol{\tau} = \boldsymbol{K}_{\boldsymbol{q}} \boldsymbol{n}_\mathrm{e} + \boldsymbol{K}_{\boldsymbol{\omega}} (\boldsymbol{\omega}_\mathrm{d} - \boldsymbol{\omega}) + \boldsymbol{J}\dot{\boldsymbol{\omega}}_\mathrm{d} + \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}, \tag{26}$$

where $\boldsymbol{K}_{\boldsymbol{q}}$ and $\boldsymbol{K}_{\boldsymbol{\omega}}$ are positive-definite controller gain matrices. Here, the first term generates a proportional torque component that is aligned with the rotation-error axis, $\boldsymbol{n}_\mathrm{e}$, thus directly acting to bring the body-fixed frame into alignment with the desired frame. To ensure that this contribution to the robot's rotation occurs in the direction aligned with the shortest rotational path, for real-time implementation, the first term in Eq. 26 can be adjusted to $\mathrm{sgn}\{m_\mathrm{e}\}\boldsymbol{K}_{\boldsymbol{q}}\boldsymbol{n}_\mathrm{e}$. The second term in Eq. 26 functions as a derivative feedback component that generates torque that directly depends on the rate of rotation error. The open-loop torque term, $\boldsymbol{J}\dot{\boldsymbol{\omega}}_\mathrm{d}$, enables faster trajectory tracking and the last term directly compensates for the gyroscopic coupling torque. In the absence of model uncertainty, these last two terms directly cancel the nonlinear dynamics of the system, thus performing a form of feedback linearization.

By combining the attitude controller presented in this section with the position controller discussed in Section 3.4, and an actuator mapping such as that specified by Eq. 13 or 15 (according to the scheme in Fig. 6), the proposed control architecture for flight control is completed. As discussed in Sections 3.6 and 4, the control scheme as a whole is capable of enforcing the closed-loop stability of the dynamics of the controlled robots without the need for the execution of open-loop tuning flight tests, which represents a significant advantage of the proposed architecture over other control methods that have been used to fly microrobots of the class considered in this article. In fact, thus far, open-loop system characterization, trimming, and tuning have been unavoidable steps in

the experimental implementation of flight controllers for insect-inspired flapping-wing microrobots [3–8].

3.6 Stability Analysis

The stability of the closed-loop system, as modeled by the block diagram in Fig. 6, can be assessed and enforced by independently determining the conditions for which the closed-loop attitude and position equilibrium points are asymptotically stable, using a combination of linear and nonlinear analysis techniques. The actuator mapping is static and, therefore, does not greatly impact the stability analysis. The proposed attitude controller in Fig. 6, as discussed in Section 3.5, was primarily conceived to fly FWMAVs. However, it shares many similarities with quaternion-based attitude controllers for spacecraft, already published in [30, 34, 35] and references therein. Consistently, the closed-loop attitude stability analysis presented here is mostly based on arguments in those sources. In the case considered in this paper, a complete state of the closed-loop attitude dynamics is composed of the attitude tracking error, $\boldsymbol{q}_\mathrm{e}$, and the angular-velocity tracking error, $\boldsymbol{\omega}_\mathrm{e} = \boldsymbol{\omega}_\mathrm{d} - \boldsymbol{\omega}$. Thus, after differentiating Eq. 22 and plugging Eq. 26 into Eq. 2, it is clear that a closed-loop state-space representation of the attitude dynamics of the system is given by

$$\begin{aligned} \dot{\boldsymbol{q}}_\mathrm{e} &= \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_\mathrm{e} \end{bmatrix} \otimes \boldsymbol{q}_\mathrm{e}, \\ \dot{\boldsymbol{\omega}}_\mathrm{e} &= -\boldsymbol{J}^{-1}\left(\boldsymbol{K}_{\boldsymbol{q}}\boldsymbol{n}_\mathrm{e} + \boldsymbol{K}_{\boldsymbol{\omega}}\boldsymbol{\omega}_\mathrm{e}\right). \end{aligned} \tag{27}$$

As shown in Appendix A, the vector function on the right side of Eq. 27 is locally Lipschitz continuous, and the seven-variable state-space system is autonomous and has two fixed points. The first is given by the vectors $\boldsymbol{q}_\mathrm{e}^* = [1\ 0\ 0\ 0]^T$ and $\boldsymbol{\omega}_\mathrm{e}^* = [0\ 0\ 0]^T$, and the second corresponds to the same zero angular velocity but with $\boldsymbol{q}_\mathrm{e}^\dagger = [-1\ 0\ 0\ 0]^T$. A detailed derivation of these two equilibrium points is provided in Appendix B. Note that both points represent the same attitude control error and, therefore, convergence to either of them represents convergence to the same physical orientation. However, as shown in Appendix C, for the attitude control scheme depicted in Fig. 6 and discussed in Section 3.5, regardless of the values of the positive-definite controller matrices $\boldsymbol{K}_{\boldsymbol{q}}$ and $\boldsymbol{K}_{\boldsymbol{\omega}}$, the second equilibrium point is always unstable and, for this reason, we analyze and discuss the stability and convergence to the first equilibrium point, $\left\{\boldsymbol{q}_\mathrm{e}^*, \boldsymbol{\omega}_\mathrm{e}^*\right\}$, only. The logic behind this approach is that since the fixed point corresponding to $m_\mathrm{e} = -1$ is unstable, if the system were to reach this state, any small disturbance would force the system to exit it and converge to $\left\{\boldsymbol{q}_\mathrm{e}^*, \boldsymbol{\omega}_\mathrm{e}^*\right\}$, provided that asymptotic stability of this first fixed point is guaranteed by the feedback controller. The apparently paradoxical situation in which two mathematical states describing the same physical condition of the system have contradictory stability properties is the result of the *quaternion sign ambiguity*, which has been thoroughly discussed in the literature [31]. However, note that, as demonstrated in Sections 3.7 and 4.1, this notational issue does not represent a problem from an engineering perspective because the proposed control scheme effectively and robustly functions in both simulations and experiments.

Here, we enforce the asymptotic stability of $\left\{\boldsymbol{q}_\mathrm{e}^*, \boldsymbol{\omega}_\mathrm{e}^*\right\}$ by invoking *Lyapunov's direct method* with *LaSalle's invariance principle* as presented in Corollary 4.2 of [36].

Proposition 1 *Let the user-defined reference signals, $\boldsymbol{r}_\mathrm{d}$ and ψ_d, be smooth functions of time, and let $\boldsymbol{K}_{\boldsymbol{q}}$ and $\boldsymbol{K}_{\boldsymbol{\omega}}$ be constant positive-definite matrices. Then, the equilibrium point $\left\{\boldsymbol{q}_\mathrm{e}^*, \boldsymbol{\omega}_\mathrm{e}^*\right\}$, corresponding to $\boldsymbol{q}_\mathrm{e}^* = [1\ 0\ 0\ 0]^T$ and $\boldsymbol{\omega}_\mathrm{e}^* = [0\ 0\ 0]^T$ of the closed-loop state-space representation of the attitude dynamics specified by* Eq. 27*, is asymptotically stable.*

Proof Let $V(\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e})$ be the *Lyapunov function* (LF) candidate defined as

$$V(\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e}) = \frac{1}{2}\boldsymbol{\omega}_\mathrm{e}^T \boldsymbol{K}_{\boldsymbol{q}}^{-1}\boldsymbol{J}\boldsymbol{\omega}_\mathrm{e} + 2(1 - m_\mathrm{e}). \tag{28}$$

It can be shown that, since $\boldsymbol{r}_\mathrm{d}$ and ψ_d are smooth functions and $\boldsymbol{K}_{\boldsymbol{q}} > 0$, $V(\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e})$ satisfies the requirements of an LF as specified in [36]. Namely, the smoothness of $\boldsymbol{r}_\mathrm{d}$ and ψ_d ensures that the function $V(\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e})$ remains continuously differentiable. Additionally, plugging $(\boldsymbol{q}_\mathrm{e}^*, \boldsymbol{\omega}_\mathrm{e}^*)$ into Eq. 28 yields

$$V(\boldsymbol{q}_\mathrm{e}^*, \boldsymbol{\omega}_\mathrm{e}^*) = 0. \tag{29}$$

Furthermore, recalling that the value of m_e remains bounded between ± 1 and $\boldsymbol{K}_{\boldsymbol{q}} > 0$, it immediately follows that

$$V(\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e}) > 0, \quad \forall\ \left\{\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e}\right\} \neq \left\{\boldsymbol{q}_\mathrm{e}^*, \boldsymbol{\omega}_\mathrm{e}^*\right\}. \tag{30}$$

By simple inspection, it can be shown that the norm of the state would go to infinity only when the norm of the angular velocity goes to infinity. Also, it can be shown that the chosen LF is radially unbounded because

$$\|\boldsymbol{\omega}_\mathrm{e}\|_2 \to \infty \ \Rightarrow\ V(\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e}) \to \infty, \tag{31}$$

thus satisfying all the conditions for an LF as specified in [36].

Next, by differentiating Eq. 28, we obtain that

$$\dot{V}\left(\boldsymbol{q}_\mathrm{e}, \boldsymbol{\omega}_\mathrm{e}\right) = \frac{1}{2}\dot{\boldsymbol{\omega}}_\mathrm{e}^T \boldsymbol{K}_{\boldsymbol{q}}^{-1}\boldsymbol{J}\boldsymbol{\omega}_\mathrm{e} + \frac{1}{2}\boldsymbol{\omega}_\mathrm{e}^T \boldsymbol{K}_{\boldsymbol{q}}^{-1}\boldsymbol{J}\dot{\boldsymbol{\omega}}_\mathrm{e} - 2\dot{m}_\mathrm{e}, \tag{32}$$

which can be simplified by recalling that $\boldsymbol{K}_{\boldsymbol{q}}^{-1}\boldsymbol{J}$ is Hermitian positive-definite and thus $\boldsymbol{K}_{\boldsymbol{q}}^{-1}\boldsymbol{J} = (\boldsymbol{K}_{\boldsymbol{q}}^{-1}\boldsymbol{J})^T$. Furthermore, by examining $\dot{\boldsymbol{q}}_\mathrm{e}$ in Eq. 27, it follows

that $\dot{m}_e = -\frac{1}{2}\boldsymbol{\omega}_e^T \boldsymbol{n}_e$. Thus, substituting these identities into Eq. 32 yields

$$\dot{V}\left(\boldsymbol{q}_e, \boldsymbol{\omega}_e\right) = \boldsymbol{\omega}_e^T \boldsymbol{K}_q^{-1} \boldsymbol{J} \dot{\boldsymbol{\omega}}_e + \boldsymbol{\omega}_e^T \boldsymbol{n}_e. \tag{33}$$

Additionally, using the second line of Eq. 27, this expression can be further simplified and written as

$$\dot{V}\left(\boldsymbol{q}_e, \boldsymbol{\omega}_e\right) = -\boldsymbol{\omega}_e^T \boldsymbol{K}_q^{-1} \boldsymbol{K}_\omega \boldsymbol{\omega}_e. \tag{34}$$

Last, since $\boldsymbol{K}_\omega$ is positive definite, the time derivative of the LF is negative semidefinite for any conceivable state. Namely,

$$\dot{V}\left(\boldsymbol{q}_e, \boldsymbol{\omega}_e\right) \leq 0, \quad \forall \left\{\boldsymbol{q}_e, \boldsymbol{\omega}_e\right\}, \tag{35}$$

where the set of states for which $\dot{V}(\boldsymbol{q}_e, \boldsymbol{\omega}_e) = 0$ is $\left\{\boldsymbol{q}_e, \boldsymbol{\omega}_e^*\right\}$, for any $\boldsymbol{q}_e$. However, by examining Eq. 27, it is clear that when $\boldsymbol{\omega}_e = \boldsymbol{\omega}_e^*$ and $\boldsymbol{n}_e \neq \boldsymbol{n}_e^*$, the expression for $\dot{\boldsymbol{\omega}}_e$ is nonzero. Therefore, no solution to $\dot{V}(\boldsymbol{q}_e, \boldsymbol{\omega}_e) = 0$ can remain in the set $\left\{\boldsymbol{q}_e, \boldsymbol{\omega}_e^*\right\}$ except at the two closed-loop equilibrium points. Note that since the equilibrium point corresponding to $m_e = -1$ is unstable, for any conceivable initial condition, the state of the attitude closed-loop system specified by Eq. 27 will converge to $\left\{\boldsymbol{q}_e^*, \boldsymbol{\omega}_e^*\right\}$. Therefore, this equilibrium point is asymptotically stable. □

To analyze the stability of the position closed-loop system, we assume that the rate of convergence of the attitude error is fast enough such that $\boldsymbol{b}_3 \approx \boldsymbol{b}_{d,3}$ and, as a consequence,

$$f\boldsymbol{b}_3 \approx \boldsymbol{f}_a. \tag{36}$$

This assumption has been validated through simulations and experiments. As shown in Fig. 6, the position tracking error is given by $\boldsymbol{r}_e = \boldsymbol{r}_d - \boldsymbol{r}$. Thus, using Eq. 36 and plugging Eq. 16 into Eq. 1 yields

$$m\ddot{\boldsymbol{r}}_e(t) = -\boldsymbol{K}_p \boldsymbol{r}_e(t) - \boldsymbol{K}_i \int_0^t \boldsymbol{r}_e(\tau) d\tau - \boldsymbol{K}_d \dot{\boldsymbol{r}}_e(t). \tag{37}$$

Since $\boldsymbol{K}_p$, $\boldsymbol{K}_i$, and $\boldsymbol{K}_d$ are diagonal matrices, this equation can be decoupled into a system of three third-order LTI ordinary differential equations of the form

$$m\dddot{\boldsymbol{x}} + \boldsymbol{K}_d \ddot{\boldsymbol{x}} + \boldsymbol{K}_p \dot{\boldsymbol{x}} + \boldsymbol{K}_i \boldsymbol{x} = 0, \tag{38}$$

where $\boldsymbol{x}(t) = \int_0^t \boldsymbol{r}_e(\tau) d\tau$. Clearly, the sole equilibrium point of this linear system is $\boldsymbol{x}^* = \dot{\boldsymbol{x}}^* = \ddot{\boldsymbol{x}}^* = [0\ 0\ 0]^T$ and its global asymptotic stability can be shown using classical linear systems theory. Here, we write this result in the form of a proposition.

Proposition 2 *Let $\boldsymbol{K}_p$, $\boldsymbol{K}_i$, and $\boldsymbol{K}_d$ be diagonal matrices that satisfy the inequalities $\boldsymbol{K}_d > \boldsymbol{0}$, $\boldsymbol{K}_p - m\boldsymbol{K}_i\boldsymbol{K}_d^{-1} > \boldsymbol{0}$, and $\boldsymbol{K}_i > \boldsymbol{0}$. Then, the equilibrium point, $\boldsymbol{x}^* = \dot{\boldsymbol{x}}^* = \ddot{\boldsymbol{x}}^* = \boldsymbol{0}$, of the closed-loop system described by* Eq. 38 *is globally asymptotically stable.*

Proof The linear system described by Eq. 38 is globally asymptotically stable if all the roots of the three associated characteristic polynomials have negative real parts. Directly from the Routh-Hurwitz stability criterion, it follows that this condition is satisfied as long as $\boldsymbol{K}_d > \boldsymbol{0}$, $\boldsymbol{K}_p - m\boldsymbol{K}_i\boldsymbol{K}_d^{-1} > \boldsymbol{0}$, and $\boldsymbol{K}_i > \boldsymbol{0}$. □

3.7 Simulation Results

Before experimentally testing the architecture for flight control of FWMAVs presented in Sections 3.4 and 3.5, we implemented and executed Simulink simulations, according to the block diagram in Fig. 6, to validate its functionality and evaluate its performance. The plant used for simulation is the rigid-body model specified by Eq. 6 in Section 3.1 with the parameters of the Bee^{++} ($m = 10^{-4}$ kg; $\boldsymbol{J} = \text{diag}\left\{3.86 \times 10^{-9}, 2.57 \times 10^{-9}, 3.99 \times 10^{-9}\right\}$ kg · m^2). To add realism to the simulation, we added actuator and sensor noise. We generated additive sensor noise signals by filtering white noise, which is produced using the uniform pseudorandom number generation block, through digital *finite impulse-response* (FIR) lowpass filters with order 2^{10} and a cutoff frequency of 200 Hz. The noise signals added to the simulated position signals r_1, r_2, and r_3 have *experimental standard deviations* (ESDs) of 10^{-5}, 2×10^{-5}, and 6×10^{-5} m, respectively; the noise signals added to the simulated attitude signals ψ, θ, and ϕ have ESD values of 0.5729, 0.2865, and 0.1719°, respectively. To generate actuator noise, we added periodic disturbances with time-varying amplitudes and biases to the four forces generated by the flapping wings. We defined these signals to vary their amplitudes and biases randomly, using the uniform pseudorandom number generation block, at intervals of 0.003 s. The resulting *root-mean-square* (RMS) value of each of these signals is about 0.068 mN. Using experimental data, we determined that this level of actuator disturbance is similar to that experienced by the physical Bee^{++} system. The simulations were implemented and run on a Lenovo ThinkPad laptop, using version 10.1 of Simulink (MATLAB R2020a). We set the numerical solver to use the Bogacki–Shampine method with a fixed step of 0.5 ms (2 kHz), the control algorithms were executed at a simulated sampling rate of 2 kHz, and the simulated measurements of the system outputs were updated at fixed steps of 2 ms (500 Hz) to correspond with the Vicon system used in the experiments.

A simulation result is shown as an example in Fig. 7. In this case, the position reference is the constant vector $\boldsymbol{r}_d^T = \left[r_{d,1}\ r_{d,2}\ r_{d,3}\right] = [0\ 0\ 0.15]$ m, which corresponds to normal hovering. As clearly seen in Fig. 7(a), in about 1.4 s, the system reaches the empirical steady state, which exhibits errors with ESD values of about 10, 10, and 30 mm along the $\boldsymbol{n}_1$ (blue), $\boldsymbol{n}_2$ (red), and $\boldsymbol{n}_3$ (green) directions,

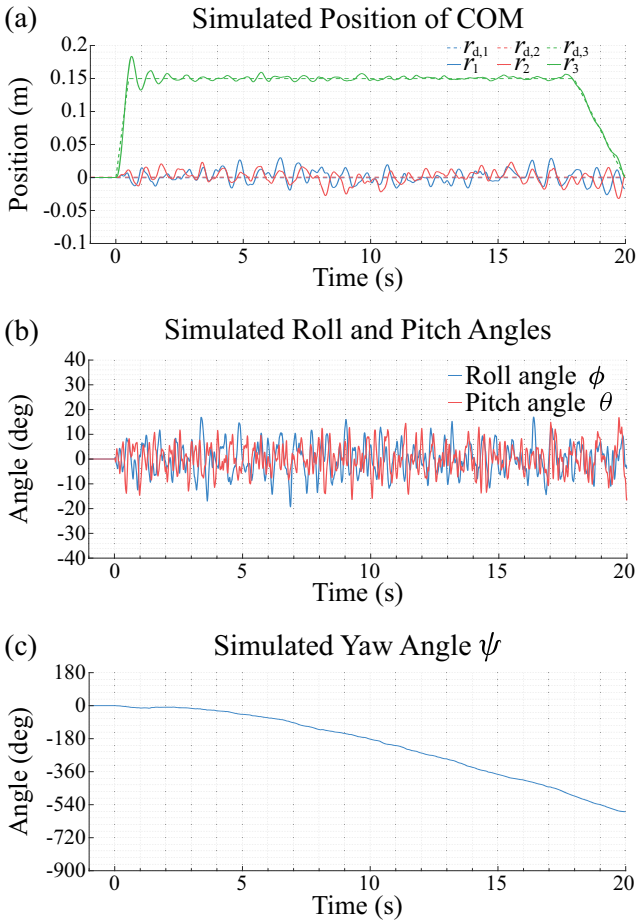

Fig. 7 Simulation results. In this numerical simulation, the Bee^{++} FWMAV was controlled to hover at the fixed inertial-position reference $\boldsymbol{r}_{\mathrm{d}} = [0\ 0\ 0.15]^T$ m. **(a)** Comparison of the position-reference vector $\boldsymbol{r}_{\mathrm{d}}(t)$ (whose three components are shown using dashed lines) with the simulated measured position vector $\boldsymbol{r}(t)$ (whose three components are shown using continuous lines). **(b)** Simulated measured roll and pitch angles, ϕ and θ. The references for ϕ and θ are not explicitly decided or computed but are implicitly determined by the desired attitude quaternion $\boldsymbol{q}_{\mathrm{d}}$, for a given yaw-angle reference ψ_{d} as specified by Eq. 21. **(c)** Simulated measured yaw angle ψ. In order to leave this DOF in open loop, for the computation of $\boldsymbol{S}_{\mathrm{d}}$ according to Eq. 18, we set $\psi_{\mathrm{d}} = \psi$

respectively. In agreement with the control scheme in Fig. 6, the roll and pitch angles shown in Fig. 7(b) correspond to the attitude required to reach the desired position $\boldsymbol{r}_{\mathrm{d}}$, $\boldsymbol{q}_{\mathrm{d}}$. For the computation of $\{\boldsymbol{b}_{\mathrm{d},1}, \boldsymbol{b}_{\mathrm{d},2}, \boldsymbol{b}_{\mathrm{d},3}\}$ as specified by Eq. 18, we used $\psi_{\mathrm{d}} = \psi$, which is equivalent to leaving the yaw DOF in open loop. The time evolution of the simulated yaw angle ψ is shown in Fig. 7(c). Overall, the three plots in Fig. 7 show a nearly perfect performance of the scheme in Fig. 6 as a position control method. Clearly, the controller quickly and effectively stabilizes the robot's position and attitude with both very small overshoot and steady-state error. This simulation alone is not a rigorous method for predicting experimental performance of the controller because it does not account for all the types of environmental disturbances affecting FWMAVs during flight. However, it does serve as an effective test platform for evaluating and debugging controller updates prior to implementation on the physical systems.

4 Experimental Results

4.1 Experimental Setup

To implement the proposed control architecture and test its ability to stably fly both the RoboBee and Bee^{++} FWMAVs, we built a flight arena instrumented with low-latency high-resolution sensors, fast digital signal processors, and high-precision actuator drivers, which in combination enable high-performance position and attitude tracking. A six-V5-camera Vicon motion capture system is used to measure the position and orientation of the tested robots during flight. These measured signals are sent to the real-time controller at a rate of 500 Hz. Before flight control experiments are performed, the tested robotic prototypes are outfitted with four reflective markers (see Fig. 1) to enable detection by the Vicon cameras and thus real-time capture of the rigid motions of the flyers. Specifically, by arranging the Vicon V5 cameras distributively about the flight test arena and using the Vicon Tracker 3.5 software, the three components of the position vector, $\boldsymbol{r}$, and the three Euler angles in the x-y-z body-fixed rotation convention are estimated in real time. For the purpose of control implementation, we further convert these three angles to the z-y-x body-fixed rotation convention, $\{\psi, \theta, \phi\}$, which is equivalent to the roll-pitch-yaw sequence of rotations with respect to the inertial frame. As shown in Fig. 6, $\boldsymbol{r}$ is directly fedback to be used by the position controller block while ϕ, θ, and ψ are used to compute the measured attitude quaternion, $\boldsymbol{q}$, as specified by the method described in [30, 31]. According to data obtained through calibration tests, the Vicon system generates highly-accurate measurements with observed levels of sensor noise with ESDs with values of less than 0.1 mm and 0.4° for translational and rotational motion, respectively.

During the performance of flight experiments, the controller in Fig. 6 is run on a host-target MathWorks Simulink Real-Time system (MATLAB R2016a). The target computer functions as a real-time digital signal processor and physically supports the board (National Instruments PCI-6229) used for *analog-to-digital* (AD) and *digital-to-analog* (DA) conversion at a sample-and-hold rate of 2 kHz. Before being processed by the control algorithms, the components of the measured position and attitude signals are filtered through lowpass filters, with the continuous-time form

$$H_1(s) = \frac{\omega_{\mathrm{c}}}{s + \omega_{\mathrm{c}}}, \tag{39}$$

with the purpose of eliminating high-frequency noise. Here, ω_c is the -3-dB cutoff frequency of the filter, which is chosen to be $200\,\text{rad}\cdot\text{s}^{-1}$ (31.83 Hz) in order to effectively cancel noise introduced by the upsampling processing of $\boldsymbol{r}$ and the Euler angles. Upsampling is required because the Vicon system functions at 500 Hz and the controller at 2 kHz. We chose to use a first-order filter to minimize the latency introduced by the filtering process. During the implementation of the algorithms, the continuous-time transfer function of the filter, $H_1(s)$, was discretized using the bilinear method.

As already explained, the Vicon-measured Euler angles are transformed into the measured attitude quaternion, $\boldsymbol{q}$, simply using quaternion algebra and the rotation matrix $\boldsymbol{S}$. However, since the Vicon system does not provide any form of attitude rate, the attitude quaternion's derivative, $\dot{\boldsymbol{q}}$, and the attitude-reference quaternion's derivative, $\dot{\boldsymbol{q}}_\text{d}$, are estimated using a *lowpass derivative filter* (LDF) with the continuous-time form

$$H_2(s) = \frac{\lambda s}{s+\lambda}, \tag{40}$$

where the parameter λ determines the speed of the filter. To implement the flight controller experimentally, through an iterative tuning process, λ was chosen to be $300\,\text{rad}\cdot\text{s}^{-1}$ (47.75 Hz), a value for which the filter was observed to estimate $\dot{\boldsymbol{q}}_\text{d}$ reasonably well, without significantly amplifying noise and with a very small delay. During the implementation of the algorithms, the continuous-time transfer function of the LDF filter, $H_2(s)$, was discretized using the bilinear method. A similar LDF was implemented to estimate the velocity vector, $\dot{\boldsymbol{r}}$, which is required by the position controller block in Fig. 6.

During operation, once all the state variables of the robot are measured and filtered, or estimated, the control algorithm calculates the actuator-exciting voltages, e_i, for $i \in \{1, 2\}$ in the RoboBee case and $i \in \{1, 2, 3, 4\}$ in the Bee^{++} case, as specified by Eq. 11 and depicted in Fig. 6. These voltages are transmitted from the DA board of the target computer to a set of drivers that provide the power required to properly excite the piezoelectric actuators of the tested FWMAV. The voltages generated by the drivers are transmitted to the actuators of the tested Robot using 52-AWG (20 µm in diameter) copper wires.

4.2 RoboBee Flight Tests

The experimental results presented in this section were obtained after a systematic tuning process of the control gain matrices for the scheme in Fig. 6, specified in Section 3. Here, we describe and present an analysis of the closed-loop results obtained during a typical representative flight test for a RoboBee prototype in terms of stability, performance, and robustness against model uncertainty and external perturbations. Dozens of similar experiments were performed and similar measurements were obtained in all of them.

In the particular experiment discussed in this section, the robot was initially at rest at approximately the inertial location $\boldsymbol{r} = [0\ 0\ 0]^T$ m and the position-reference vector was set at $\boldsymbol{r}_\text{d} = [0\ 0\ 0.15]^T$ m. The instantaneous yaw-angle reference signal, $\psi_\text{d}(t)$, was set equal to the current yaw-angle measurement, $\psi(t)$, which effectively disabled the yaw controller. This setting is necessary because the RoboBee design is not capable of reliably generating enough yaw torque for control via the split-cycling method to produce effective yaw tracking and stability [5–8, 37]. Fortunately, as expected from analyzing the control architecture shown in Fig. 6 and verified through experiments, the lack of sufficient yaw torque for control does not prevent

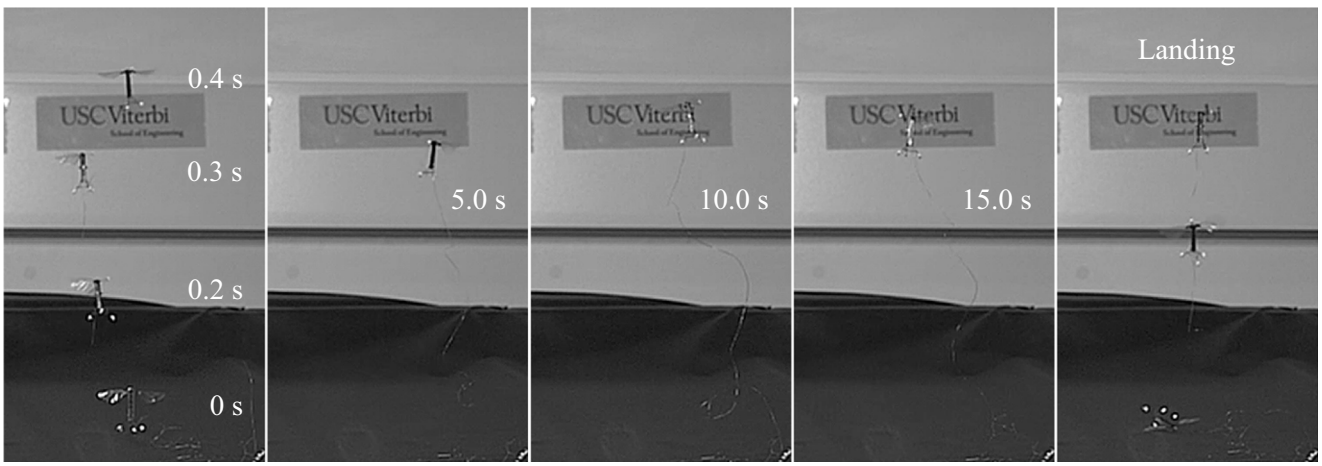

Fig. 8 Controlled flight experiment of the tested RoboBee prototype. This image composite shows the tested robot at various instants during a position control experiment. Here, the robot can be seen taking off from the inertial position $\boldsymbol{r} = [0\ 0\ 0]^T$ m to reach the inertial-position reference $\boldsymbol{r}_\text{d} = [0\ 0\ 0.15]^T$ m in about 0.5 s and fly stably for more than 15 s. A similar behavior in terms of stability and performance was observed in dozens of other flight tests

the position controller from achieving stability and high performance. Additionally, during the experimental controller tuning process, it was determined that, due to the tested RoboBee prototype's limited control authority, the integral control term in Eq. 16 induces actuator saturation, which significantly decreases the performance and stability robustness of the system. To eliminate this issue, the integral term was disabled in the experimental controller. In all the flight control tests, the tested RoboBee prototype was operated at a constant flapping frequency of 100 Hz.

As seen in the photographic composite shown in Fig. 8 and the plots in Fig. 9, during the flight test, the controller is turned on at time $t = 0$ s and the RoboBee prototype takes off to reach the reference height in about 0.5 s. Then, the robot stably maintains the desired position, $\boldsymbol{r}_{\mathrm{d}}$, with acceptable performance and moderate steady-state error until the controller is turned off at time $t = 19$ s. The plot in Fig. 9(a) compares the measured position, $\boldsymbol{r}$, with the reference, $\boldsymbol{r}_{\mathrm{d}}$, and Fig. 9(b) shows the measured roll and pitch angles, whose references correspond to the attitude-reference quaternion, $\boldsymbol{q}_{\mathrm{d}}(t)$, computed as specified by Eq. 21 with $\psi_{\mathrm{d}}(t) = \psi(t)$. The resulting measured yaw angle ψ is shown in Fig. 9(c). From these three plots, it is clear that the proposed controller successfully regulates both the lateral position, $\{r_1, r_2\}$, and altitude, r_3, of the controlled robot with a short transient period and an overshoot of about 27 % relative to the steady-state mean altitude. The lateral-position signals exhibit steady-state biases of about 0.010 m, which for all practical purposes can be considered to be negligible. Also, these two variables exhibit seemingly random lateral errors, with ESD values of about 0.020 m in the r_1 measurement and about 0.018 m in the r_2 measurement. The measured altitude, on the other hand, displays a more noticeable negative steady-state bias of about 0.027 m and a seemingly random error with an ESD value of about 0.012 m. Video footage of the complete experiment can be seen in the accompanying Supplementary Movie.

In the case of the RoboBee, positional errors are primarily caused by environmental disturbances and noise in the angular-velocity measurement of the controlled FWMAVs. In fact, small variations in the local airflows surrounding a RoboBee prototype and shifts in the actuator wiring can have a significant impact on the corresponding control performance due to the small mass and rotational inertia of the system. Note that lateral-position disturbances are more difficult to cancel than vertical ones because of the delay in the dynamic response between when the force vector in Eq. 16 is requested by the position controller and when it is achieved by the attitude controller. Furthermore, the estimation of the robot's angular velocity, $\boldsymbol{\omega}$, and the calculation of the desired angular velocity, $\boldsymbol{\omega}_{\mathrm{d}}$, both require the use of lowpass filters with the form specified by Eq. 40,

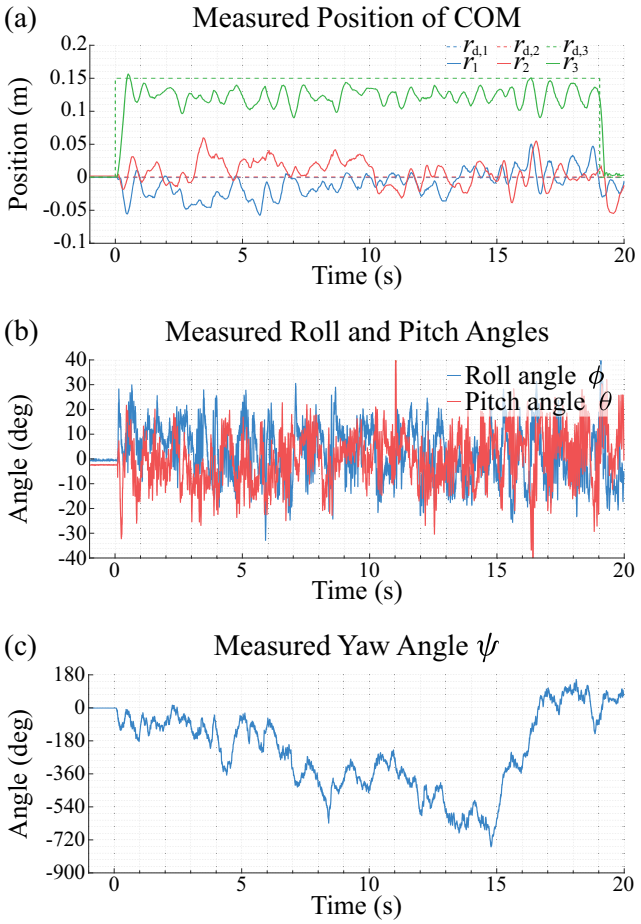

Fig. 9 Experimental results obtained using a RoboBee prototype. In this flight test, the tested FWMAV is controlled to hover at the fixed inertial-position reference $\boldsymbol{r}_{\mathrm{d}} = [0\ 0\ 0.15]$ m. **(a)** Comparison of the position-reference vector $\boldsymbol{r}_{\mathrm{d}}(t)$ (whose three components are shown using dashed lines) with the experimentally-measured position vector $\boldsymbol{r}(t)$ (whose three components are shown using continuous lines). **(b)** Experimental roll and pitch Euler angles, ϕ and θ, which were measured in real time using the Vicon motion capture system described in Section 4.1. During controlled flight, the references for ϕ and θ are not explicitly decided or computed but implicitly determined by the desired attitude quaternion $\boldsymbol{q}_{\mathrm{d}}$, for a given yaw-angle reference ψ_{d} as specified by Eq. 21. **(c)** Experimentally-measured yaw angle ψ. In order to leave this DOF in open loop, for the computation of $\boldsymbol{S}_{\mathrm{d}}$ according to Eq. 18, we set $\psi_{\mathrm{d}} = \psi$

which under certain conditions can introduce non-negligible amounts of noise into the calculation of the angular-velocity error, as specified by Eq. 26. We believe that this type of noise contributes to the steady-state oscillation observed in Fig. 9. The smaller oscillatory control errors in altitude are explained by the greater control authority over the r_3 DOF, which results from the near-perfect alignment of the generated thrust force with the body-fixed $\boldsymbol{b}_3$ direction. The steady-state altitude bias observed in Fig. 9(a) is explained by systematic errors introduced by the linear actuator

mapping specified by Eq. 13, which is an approximation, coupled with the lack of integral control action.

From Fig. 9(b), it can be deduced that the $\boldsymbol{b}_3$ axis of the tested RoboBee prototype reaches a stable alignment with the inertial $\boldsymbol{n}_3$ axis. Note, however, that the roll and pitch angles undergo significant variations during flight. These variations display quantified steady-state biases of about 2.7° in roll and about 0.1° in pitch. Also, these two angle signals exhibit oscillations with ESD values of about 10.4° and about 10.1°, respectively. According to the method described in Section 3.5, during flight, the proposed control scheme intentionally introduces some attitude deviation relative to the $\boldsymbol{n}_3$ axis because the position controller adjusts the attitude reference signal as specified by Eq. 18 in order to point the robot toward the position reference in space. However, the large error values seen in Fig. 9(b) are predominantly attributable to underactuation and the considerable effects of the aerodynamic disturbance forces and torques affecting the system. The impact of these disturbances on the attitude of the robot is inversely correlated with the magnitudes of the inertial parameters of the system. Namely, due to the small values of the mass and inertia matrix of the tested prototype, the relative impact of moderate external disturbances on the system is always significant. Furthermore, because the tested RoboBee prototype has a limited capability to generate aerodynamic control forces and torques, its associated thrust-to-weight ratio is low and, as a consequence, its actuators always operate near saturation. In general, if the control signals reach the saturation limits due to the influence of external disturbances, the system can begin to destabilize. Fortunately, according to the real-time experimental data, the external disturbances affecting the system tend to have short durations, enabling the controller to recover after destabilizing events.

4.3 Bee^{++} Flight Tests

In the flight experiments of the tested Bee^{++}, we also used the experimental setup described in Section 4.1. As in the RoboBee case, we present the results obtained through a single representative flight test. In general, the discussed experimental test is highly repeatable and the measured response can be considered typical in terms of performance and stability robustness. As seen in the photographic composite shown in Fig. 10 and plots in Fig. 11, during the flight test, the robot starts its trajectory from approximately $\boldsymbol{r} = [0\ 0\ 0]^T$ m. Then, commanded by the position reference $\boldsymbol{r}_\mathrm{d}(t)$, for $t \in [0:20]$ s, which is plotted using dashed lines in Fig. 11(a), at time $t = 0$ s, the robot takes off to reach the reference point $\boldsymbol{r}_\mathrm{d} = [0\ 0\ 0.15]^T$ m. Last, at time $t = 18$ s, commanded by the position reference, the robot starts to land. In summary, during the 20 s that the flight experiment lasted, the tested Bee^{++} prototype successfully took off, stably tracked a constant altitude-position reference, and performed a controlled landing maneuver. Video footage of the entire experiment can be seen in the accompanying Supplementary Movie.

As in the RoboBee case, yaw control was disabled by setting the yaw-angle reference to be equal to the measured instantaneous yaw angle, i.e., $\psi_\mathrm{d}(t) = \psi(t)$. While the four-wing design of the Bee^{++} enables effective yaw control, this novel feature of the robot was disabled in the implementation of the flight experiments in order to ensure consistency of comparison with the experimental results obtained with the RoboBee prototype. For the same reason, the integral action of the position controller specified by Eq. 16 was also disabled before the performance of the flight tests. In the experiment corresponding to the data shown in Fig. 10, the tested Bee^{++} prototype operated with a constant

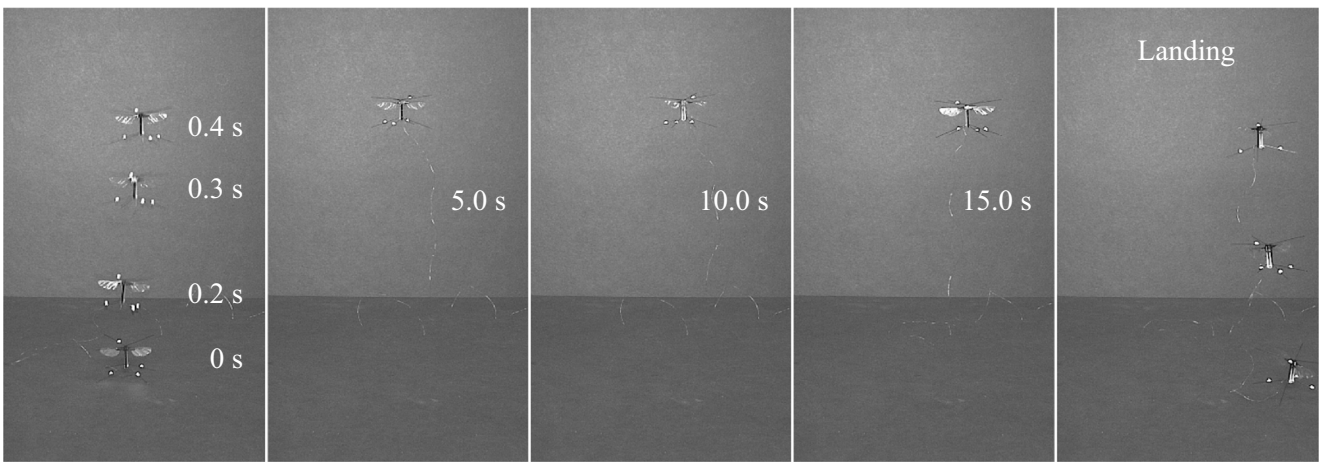

Fig. 10 Controlled flight experiment of the tested Bee^{++} prototype. This image composite shows the tested robot at various instants during a position control experiment. Here, the robot can be seen taking off from the inertial position $\boldsymbol{r} = [0\ 0\ 0]^T$ m to reach the inertial-position reference $\boldsymbol{r}_\mathrm{d} = [0\ 0\ 0.15]^T$ m in about 0.5 s and fly stably for more than 15 s. A similar behavior in terms of stability and performance was observed in dozens of other flight tests

flapping frequency of 150 Hz. This value was chosen to be lower than the estimated 165 Hz bandwidth of the lowpass mapping that receives as inputs the actuator excitations, e_i, for $i \in \{1, 2, 3, 4\}$, and produces as outputs the flapping motions, φ_i, for $i \in \{1, 2, 3, 4\}$, in order to increase the life expectancy of the compliant components of the robot by extending the time required to reach fatigue.

The plot in Fig. 11(a) clearly shows that the proposed control method is capable of regulating the lateral positions

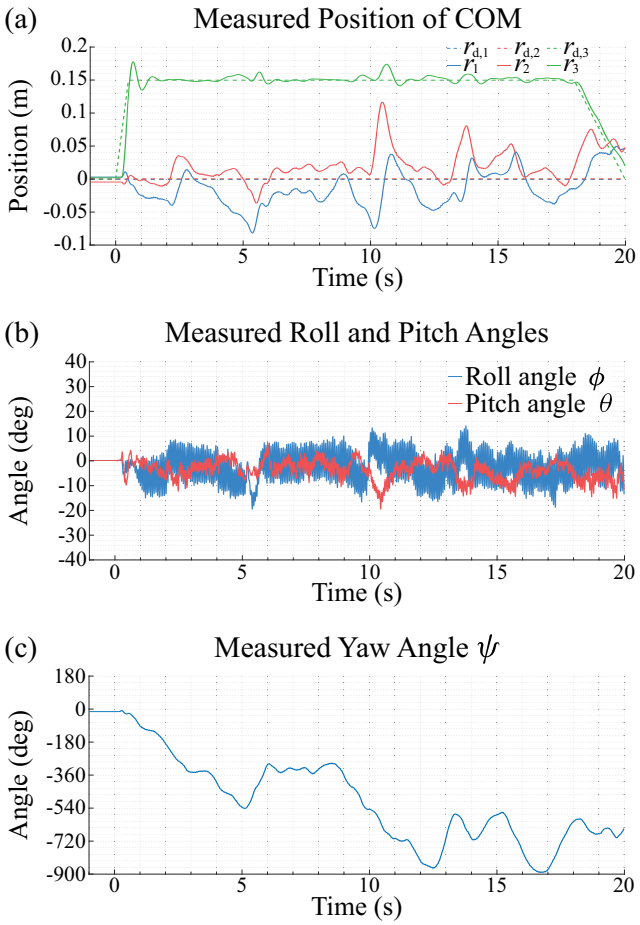

Fig. 11 Experimental results obtained using a Bee^{++} prototype. In this flight test, the tested FWMAV is controlled to hover at the fixed inertial-position reference $\boldsymbol{r}_{\mathrm{d}} = [0\ 0\ 0.15]^T$ m. **(a)** Comparison of the position-reference vector $\boldsymbol{r}_{\mathrm{d}}(t)$ (whose three components are shown using dashed lines) with the experimentally-measured position vector $\boldsymbol{r}(t)$ (whose three components are shown using continuous lines). During the experiment, all three measured components remained stable about the reference signals with reasonably small steady-state errors. **(b)** Experimental roll and pitch Euler angles, ϕ and θ, which were measured in real time using the Vicon motion capture system described in Section 4.1. During controlled flight, the references for ϕ and θ are not explicitly decided or computed but implicitly determined by the desired attitude quaternion $\boldsymbol{q}_{\mathrm{d}}$, for a given yaw-angle reference ψ_{d} as specified by Eq. 21. During the experiment, these two Euler angles remain stable and with reasonably small steady-state errors. **(c)** Experimentally-measured yaw angle ψ. In order to leave this DOF in open loop, for the computation of $\boldsymbol{S}_{\mathrm{d}}$ according to Eq. 18, we set $\psi_{\mathrm{d}} = \psi$

and altitude of the tested Bee^{++} prototype. The transient period after reaching the reference point [0 0 0.15] m is quite fast with an overshoot of only 17 % and, unlike in the RoboBee case, the altitude control error shows a negligible steady-state bias and the corresponding ESD value is in the order of only 0.005 m, which represents a significant improvement with respect to the performance achieved by the tested RoboBee prototype. The lateral-position error has a bias of about -0.020 m in the r_1 direction and of about 0.017 m in the r_2 direction, and ESD values of about 0.024 m and about 0.023 m in the r_1 and r_2 directions, respectively. These ESD values are slightly worse than those corresponding to the RoboBee experiments. As in the RoboBee case, the lateral-position control errors are the result of environmental disturbances and numerical errors introduced during the estimation of the angular-velocity state. We hypothesize that the electrical tether connecting the robot to the piezoelectric-actuator drivers causes an approximately-constant disturbance, which produces the steady-state lateral biases observed in Fig. 11(a). This issue could potentially be eliminated by activating the term corresponding to integral control in Eq. 16.

Also, from Fig. 11(b), it is clear that during flight, the tested Bee^{++} prototype is able to closely align its yaw axis with the inertial vertical axis, $\boldsymbol{n}_3$, and maintain this attitude throughout the flight experiment. In this case, the measured roll and pitch angles exhibit high-frequency oscillations with respective steady-state biases of about 2.6 and 3.2°, and respective ESD values of about 6.3 and 3.8°. Note that, in principle, for a constant position reference, the attitude should correspond to roll, pitch, and yaw values of zero. However, according to the control scheme in Fig. 6, a nonzero attitude orientation of the robot with respect to the inertial frame of reference during flight is inevitable because, as previously discussed and specified by Eq. 18, the attitude reference is continually changed for the purpose of position control. Despite this fact, the amount of vibration could potentially be reduced via iterative tuning and improved filtering. As already discussed, for the purpose of direct comparison with the RoboBee, we keep the yaw DOF in open loop by setting $\psi_{\mathrm{d}}(t) = \psi(t)$. The time evolution of ψ is shown in Fig. 11(c). As expected, in the absence of control, this angle drifts significantly during flight.

From the experimental data summarized in Figs. 10 and 11, we can conclude that although the impact of external disturbances is still significant in the Bee^{++} case, its two additional wings generate a considerable additional thrust force when compared to that produced by a RoboBee prototype, which is one of the key innovations of the Bee^{++} design. During hovering flight, this additional thrust mostly acts along the inertial $\boldsymbol{n}_3$ direction, resulting in an improved vertical control authority and a 58 % reduction in the ESD value of the altitude error when compared to

those of the RoboBee case. Similarly, the four-wing-based torque generation scheme positively impacted the measured attitude performance with a 39 % drop in roll error and a 62 % drop in pitch error in terms of ESD values.

In summary, the experimental results obtained with the tested Bee^{++} prototype demonstrate a significant superiority of this robot in terms of performance with respect to both the RoboBee prototype presented in Section 4.3 and the first Bee^{+} design in [8, 27]. Most importantly, this new improved version of the four-wing Bee^{+}, with the design updates described in Section 2, can be robustly controlled and destabilizing pitch-angle oscillations observed in previous iterations of this design (see [8] and [27]) have been effectively eliminated. Solving this problem was essential to enable the tested Bee^{++} prototype to fly for extended periods of time and achieve enhanced positional stability.

5 Conclusions and Future Work

We presented a mathematically-rigorous method for synthesizing and implementing in real time a multiplatform control architecture for flapping-wing flying robotic insects. As demonstrated through simulations and flight experiments, the resulting synthesized controllers are computationally simple, experimentally reliable, and highly effective from the stability and performance perspectives. The synthesis method is based on modern linear system theory and nonlinear Lyapunov analysis. The basic control structure used for design and real-time implementation was conceived with the objective of creating a highly-reusable easy-to-tune feedback controller for implementation on a wide variety of FWMAVs that are characterized by the generation of thrust along a main single direction. The functionality and performance of the proposed control method, as a position regulator devised to enable steady level flight, were demonstrated using experimental results obtained with two different platforms. One was a slightly-modified version of the two-wing Harvard RoboBee presented in [3] and the other was the Bee^{++}, which is an improved version of the four-wing microrobot we first presented in [8] in 2019.

Although the proposed control architecture and experimental results presented in this paper represent considerable progress with respect to the current state of the art of aerial microrobotics [27], there still is ample room for improvement from the control perspective in terms of overall performance, experimental implementation, and computational efficiency. For instance, with minor modifications, controller functionality and performance could be demonstrated across a much broader range of flight experiments. Given the potential widespread applicability of FWMAVs to perform tasks such as artificial pollination, search and rescue, and exploration, it is highly desirable to accomplish more complex control objectives in order to create a set of control primitives as defined in [38, 39]. Yaw-angle regulation and tracking, trajectory tracking, attitude pointing, and object following are all potential applications of the proposed control method. Furthermore, the steady-state performance can be improved through the use of a systematic control-gain tuning process based on system identification and a more precise real-time estimation of the robot's state via an observer.

Appendix A: Lipschitz Continuity of the Closed-Loop Attitude State-Space System

Here, we show that the closed-loop state equations specified by Eq. 27 satisfy the Lipschitz continuity condition. First, recall that all continuous functions with bounded first-order derivatives satisfy the Lipschitz continuity condition, although the converse is not necessarily true [36, 40]. By inspection, it is immediately clear that the vector function on the right side of Eq. 27 is continuous. To determine if the derivatives of this vector function are bounded, it is helpful to find the Jacobian matrix of the system, which is given by

$$\boldsymbol{A}\left(\boldsymbol{q}_{\mathrm{e}}, \boldsymbol{\omega}_{\mathrm{e}}\right)=\begin{bmatrix} 0 & -\frac{1}{2}\boldsymbol{\omega}_{\mathrm{e}}^{T} & -\frac{1}{2}\boldsymbol{n}_{\mathrm{e}}^{T} \\ \frac{1}{2}\boldsymbol{\omega}_{\mathrm{e}} & \frac{1}{2}\boldsymbol{\Omega}_{\mathrm{e}} & \frac{1}{2}(-\boldsymbol{N}_{\mathrm{e}}+m_{\mathrm{e}}\boldsymbol{I}_{3\times 3}) \\ \boldsymbol{0}_{3\times 1} & -\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}} & -\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{\omega}} \end{bmatrix}, \tag{41}$$

where $\boldsymbol{A}$ is 7×7; $\boldsymbol{\Omega}_{\mathrm{e}}$ is the skew-symmetric cross-product matrix, as defined in [30], for $\boldsymbol{\omega}_{\mathrm{e}}$; and $\boldsymbol{N}_{\mathrm{e}}$ is the skew-symmetric cross-product matrix, as defined in [30], for $\boldsymbol{n}_{\mathrm{e}}$.

Last, recalling that the components of the quaternion $\boldsymbol{q}_{\mathrm{e}}$ are bounded between ± 1, it follows that all the entries of the Jacobian matrix are bounded, except when one or more of the components of $\boldsymbol{\omega}_{\mathrm{e}} \rightarrow \pm\infty$. Therefore, the vector function on the right in Eq. 27 satisfies the Lipschitz continuity condition, locally.

Appendix B: Equilibrium Points of the Closed-Loop Attitude State-Space System

To find the fixed points of the system specified by Eq. 27, we first set $\dot{\boldsymbol{q}}_{\mathrm{e}} = [0\ 0\ 0\ 0]^{T}$ and $\dot{\boldsymbol{\omega}}_{\mathrm{e}} = [0\ 0\ 0]^{T}$, which yields

$$\begin{aligned} &-\frac{1}{2}\boldsymbol{n}_{\mathrm{e}}^{T}\boldsymbol{\omega}_{\mathrm{e}} = 0, \\ &-\frac{1}{2}\left(\boldsymbol{n}_{\mathrm{e}} \times \boldsymbol{\omega}_{\mathrm{e}} - m_{\mathrm{e}}\boldsymbol{\omega}_{\mathrm{e}}\right) = \boldsymbol{0}_{3\times 1}, \\ &-\boldsymbol{J}^{-1}\left(\boldsymbol{K}_{\boldsymbol{q}}\boldsymbol{n}_{\mathrm{e}} + \boldsymbol{K}_{\boldsymbol{\omega}}\boldsymbol{\omega}_{\mathrm{e}}\right) = \boldsymbol{0}_{3\times 1}. \end{aligned} \tag{42}$$

Then, we solve for $\boldsymbol{q}_{\mathrm{e}}$ and $\boldsymbol{\omega}_{\mathrm{e}}$.

From simple examination of the second line in Eq. 42, it can be deduced that $m_e\boldsymbol{\omega}_e$ and $\boldsymbol{n}_e \times \boldsymbol{\omega}_e$ are orthogonal to each other; therefore, their difference can only be zero if both terms are equal to zero. Also, the $\boldsymbol{n}_e \times \boldsymbol{\omega}_e$ term can only be zero if $\boldsymbol{n}_e$ and $\boldsymbol{\omega}_e$ are parallel, or one or both vectors are zero. Next, note that the first line in Eq. 42 can only be satisfied if $\boldsymbol{n}_e$ and $\boldsymbol{\omega}_e$ are orthogonal to each other, or one or both vectors are zero. Since $\boldsymbol{n}_e$ and $\boldsymbol{\omega}_e$ can not simultaneously be orthogonal and parallel, one or both vectors must be zero. Last, to satisfy the third equation in Eq. 42, it is clear that, since $\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}}$ and $\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{\omega}}$ are full rank, if either $\boldsymbol{n}_e$ or $\boldsymbol{\omega}_e$ is equal to zero, then both must be zero. Therefore, $\boldsymbol{n}_e = [0\ 0\ 0]^T$ and $\boldsymbol{\omega}_e = [0\ 0\ 0]^T$ constitute the only solution to Eq. 42.

Thus far, we have not imposed any restrictions on m_e; however, since $\boldsymbol{q}_e$ is a unit quaternion, the scalar component m_e takes either the value 1 or -1 when $\boldsymbol{n}_e = \boldsymbol{0}_{3\times1}$ and $\boldsymbol{\omega}_e = \boldsymbol{0}_{3\times1}$. This implies that the closed-loop system given by Eq. 27 has two equilibrium points. Namely, $[\pm1\ 0\ 0\ 0\ 0\ 0\ 0]^T$.

Appendix C: Instability of the Second Equilibrium Point

The equilibrium point corresponding to $m_e = -1$, $\boldsymbol{n}_e = [0\ 0\ 0]^T$, and $\boldsymbol{\omega}_e = [0\ 0\ 0]^T$ can be shown to be unstable using *Lyapunov's indirect method* as presented in Theorem 4.7 of [36]. In this case, this method is valid because the vector function on the right side of the closed-loop state-space system specified by Eq. 27 is continuously differentiable within a small neighborhood of the equilibrium point. Accordingly, we first evaluate the Jacobian matrix specified by Eq. 41 at the fixed point corresponding to $m_e = -1$, which yields

$$\boldsymbol{F} = \begin{bmatrix} 0 & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3} & -\frac{1}{2}\boldsymbol{I}_{3\times3} \\ \boldsymbol{0}_{3\times1} & -\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}} & -\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{\omega}} \end{bmatrix}. \tag{43}$$

The indirect method dictates that if any eigenvalue of $\boldsymbol{F}$ has a positive real part, then the equilibrium point is unstable. In this case, clearly, one of the eigenvalues of $\boldsymbol{F}$ is 0 and the other six are the eigenvalues of the submatrix

$$\bar{\boldsymbol{F}} = \begin{bmatrix} \bar{\boldsymbol{F}}_{11} & \bar{\boldsymbol{F}}_{12} \\ \bar{\boldsymbol{F}}_{21} & \bar{\boldsymbol{F}}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -\frac{1}{2}\boldsymbol{I}_{3\times3} \\ -\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}} & -\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{\omega}} \end{bmatrix}. \tag{44}$$

To find the eigenvalues of $\bar{\boldsymbol{F}}$, we use the Schur complement of $\bar{\boldsymbol{F}}_{22}$. Specifically, the determinant of the block matrix $\bar{\boldsymbol{F}}$ can be written as

$$\det\left(\bar{\boldsymbol{F}}\right) = \det\left(\bar{\boldsymbol{F}}_{22}\right)\det\left(\bar{\boldsymbol{F}}_{11} - \bar{\boldsymbol{F}}_{12}\bar{\boldsymbol{F}}_{22}^{-1}\bar{\boldsymbol{F}}_{21}\right). \tag{45}$$

Thus, by substituting the terms that define $\bar{\boldsymbol{F}}$, it follows that

$$\begin{aligned} \det\left(\bar{\boldsymbol{F}}\right) &= \det\left(-\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{\omega}}\right)\det\left(\tfrac{1}{2}\boldsymbol{K}_{\boldsymbol{\omega}}^{-1}\boldsymbol{K}_{\boldsymbol{q}}\right) \\ &= \det\left(-\tfrac{1}{2}\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}}\right) \\ &= -\tfrac{1}{8}\det\left(\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}}\right). \end{aligned} \tag{46}$$

Next, recalling that $\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}} > 0$ and that the determinant of a positive definite matrix is always positive, we immediately conclude that $\det\left(\boldsymbol{J}^{-1}\boldsymbol{K}_{\boldsymbol{q}}\right) > 0$ and, therefore, that

$$\det\left(\bar{\boldsymbol{F}}\right) = \prod_{i=1}^{6}\lambda_i < 0, \tag{47}$$

where λ_i denotes the ith eigenvalue of $\bar{\boldsymbol{F}}$. Since $\bar{\boldsymbol{F}}$ has an even number of eigenvalues, Eq. 47 can only be satisfied if at least one eigenvalue has a positive real part. Therefore, we conclude that $\boldsymbol{F}$ must have at least one eigenvalue with positive real part and that the equilibrium point is unstable.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10846-021-01556-2.

Author Contributions R. M. Bena designed flight controllers for the tested FWMAVs; developed most of the control theory required to prove stability; performed control experiments; analyzed the experimental and simulation data; and wrote the paper.

X.-T. Nguyen developed key methods used in the fabrication of the FWMAV platforms; fabricated FWMAVs for the experiments; and wrote the paper.

X. Yang developed the Bee^{++} platform; developed the ISP method for actuation and control; developed key methods used in the fabrication of the FWMAV platforms; fabricated FWMAVs for the experiments; performed control experiments; and analyzed the experimental data.

A. A. Calderón developed key methods used in the fabrication of the FWMAV platforms; fabricated FWMAVs for the experiments; performed control experiments; analyzed the experimental data; and wrote the paper.

Y. Chen designed flight controllers for the tested FWMAVs; developed part of the control theory required to prove stability; performed control experiments; and analyzed the experimental data.

N. O. Pérez-Arancibia conceived the line of research presented in the paper; developed part of the control theory required to prove stability; analyzed the experimental and simulation data; directed the research; performed administrative and funding acquisition tasks; and wrote and edited the paper.

References

1. Finio, B.M., Wood, R.J.: Open-loop roll, pitch and yaw torques for a robotic bee. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Vilamoura-Algarve, Portugal, pp. 113–119 (2012)
2. Ma, K.Y., Felton, S.M., Wood, R.J.: Design, fabrication, and modeling of the split actuator microrobotic bee. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Vilamoura-Algarve, Portugal, pp. 1133–1140 (2012)
3. 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)

4. Pérez-Arancibia, N.O., Duhamel, P.-E.J., Ma, K.Y., Wood, R.J.: Model-free control of a flapping-wing flying microrobot. In: Proc. 16th Int. Conf. Adv. Robot. (ICAR), Montevideo, Uruguay, pp. 1–8 (2013)
5. Pérez-Arancibia, N.O., Duhamel, P.-E.J., Ma, K.Y., Wood, R.J.: Model-free control of a hovering flapping-wing microrobot. J. Intell. Robot. Syst. **77**(1), 95–111 (2015)
6. Chirarattananon, P., Ma, K.Y., Wood, R.J.: Adaptive control of a millimeter-scale flapping-wing robot. Bioinspir. Biomim. **9**(2). Art. no. 025004 (2014)
7. Fuller, S.B.: Four wings: An insect-sized aerial robot with steering ability and payload capacity for autonomy. IEEE Robot. Autom. Lett. **4**(2), 570–577 (2019)
8. Yang, X., Chen, Y., Chang, L., Calderón, A.A., Pérez-Arancibia, N.O.: Bee$^+$: A 95-mg four-winged insect-scale flying robot driven by twinned unimorph actuators. IEEE Robot. Autom. Lett. **4**(4), 4270–4277 (2019)
9. Fearing, R.S., Chiang, K.H., Dickinson, M.H., Pick, D.L., Sitti, M., Yan, J.: Wing transmission for a micromechanical flying insect. In: Proc. IEEE Int. Conf. Robot. Automat. (ICRA), San Francisco, CA, USA, pp. 1509–1516 (2000)
10. Yan, J., Avadhanula, S.A., Birch, J., Dickinson, M.H., Sitti, M., Su, T., Fearing, R.S.: Wing transmission for a micromechanical flying insect. J. Micromechatronics **1**(3), 221–237 (2002)
11. Tanaka, H., Hoshino, K., Matsumoto, K., Shimoyama, I.: Flight dynamics of a butterfly-type ornithopter. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Edmonton, AB, Canada, pp. 2706–2711 (2005)
12. Wood, R.J.: The first takeoff of a biologically inspired at-scale robotic insect. IEEE Trans. Robot. **24**(2), 341–347 (2008)
13. 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). Art. no. 036009 (2011)
14. Wood, R.J., Avadhanula, S., Sahai, R., Steltz, E., Fearing, R.S.: Microrobot design using fiber reinforced composites. J. Mech. Des. **5**(130). Art. no. 052304 (2008)
15. Yang, X., Chang, L., Pérez-Arancibia, N.O.: An 88-milligram insect-scale autonomous crawling robot driven by a catalytic artificial muscle. Sci. Robot. **5**(45). Art. no. eaba0015 (2020)
16. Doman, D.B., Oppenheimer, M.W., Sigthorsson, D.O.: Wingbeat shape modulation for flapping-wing micro-air-vehicle control during hover. J. Guid. Control Dynam. **33**(3), 724–739 (2010)
17. Chen, Y., Pérez-Arancibia, N.O.: Generation and real-time implementation of high-speed controlled maneuvers using an autonomous 19-gram quadrotor. In: Proc. IEEE Int. Conf. Robot. Automat. (ICRA), Stockholm, Sweden, pp. 3204–3211 (2016)
18. Chen, Y., Pérez-Arancibia, N.O.: Lyapunov-based controller synthesis and stability analysis for the execution of high-speed multi-flip quadrotor maneuvers. In: Proc. Amer. Control Conf. (ACC), Seattle, WA, USA, pp. 3599–3606 (2017)
19. Chen, Y., Pérez-Arancibia, N.O.: Nonlinear adaptive control of quadrotor multi-flipping maneuvers in the presence of time-varying torque latency. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Madrid, Spain, pp. 7845–7852 (2018)
20. Chen, Y., Pérez-Arancibia, N.O.: Adaptive control of aerobatic quadrotor maneuvers in the presence of propeller-aerodynamic-coefficient and torque-latency time-variations. In: Proc. IEEE Int. Conf. Robot. Automat. (ICRA), Montreal, QC, Canada, pp. 6447–6453 (2019)
21. Chen, Y., Pérez-Arancibia, N.O.: Controller synthesis and performance optimization for aerobatic quadrotor flight. IEEE Trans. Control Syst. Technol. **28**(6), 2204–2219 (2020)
22. Ellington, C.P.: The aerodynamics of hovering insect flight. I. The quasi-steady analysis. Phil. Trans. R. Soc. Lond. **305**(1122), 1–15 (1984)
23. Karpelson, M., Wei, G.-Y., Wood, R.J.: Milligram-scale high-voltage power electronics for piezoelectric microrobots. In: Proc. IEEE Int. Conf. Robot. Automat. (ICRA), Kobe, Japan, pp. 2217–2224 (2009)
24. Chang, L., Pérez-Arancibia, N.O.: The dynamics of passive wing-pitching in hovering flight of flapping micro air vehicles using three-dimensional aerodynamic simulations. In: Proc. AIAA Atmos. Flight Mech. Conf., SciTech Forum, San Diego, CA, USA. Art. no. AIAA 2016–0013 (2016)
25. Chang, L., Pérez-Arancibia, N.O.: Time-averaged dynamic modeling of a flapping-wing micro air vehicle with passive rotation mechanisms. In: Proc. AIAA Atmos. Flight Mech. Conf., AVIATION Forum, Atlanta, GA, USA. Art. no. AIAA 2018–2830 (2018)
26. Singer, E.K., Chang, L., Calderón, A.A., Pérez-Arancibia, N.O.: Clip-brazing for the design and fabrication of micronewton-resolution, millimeter-scale force sensors, Smart Mater. Struct. **28**(5). Art. no. 055028 (2019)
27. Calderón, A.A., Chen, Y., Yang, X., Chang, L., Nguyen, X.-T., Singer, E.K., Pérez-Arancibia, N.O.: Control of flying robotic insects: A perspective and unifying approach. In: Proc. 19th Int. Conf. Adv. Robot. (ICAR), Belo Horizonte, Brazil, pp. 747–754 (2019)
28. Chen, Y., Ma, K., Wood, R.J.: Influence of wing morphological and inertial parameters on flapping flight performance. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Daejeon, Korea, pp. 2329–2336 (2016)
29. Jafferis, N.T., Graule, M.A., Wood, R.J.: Non-linear resonance modeling and system design improvements for underactuated flapping-wing vehicles. In: Proc. IEEE Int. Conf. Robot. Automat. (ICRA), Stockholm, Sweden, pp. 3234–3241 (2016)
30. Markley, F.L., Crassidis, J.L.: Fundamentals of Spacecraft Attitude Determination and Control. Springer, New Yorkk, NY, USA (2014)
31. Kuipers, J.B.: Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality. Princeton University Press, Princeton, NJ, USA (2002)
32. Whitney, J.P., Wood, R.J.: Aeromechanics of passive rotation in flapping flight. J. Fluid Mech. **660**, 197–220 (2010)
33. Murray, R.M.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton, FL, USA (1994)
34. Wie, B., Weiss, H., Arapostathis, A.: Quaternion feedback regulator for spacecraft eigenaxis rotations. J. Guid. Control Dynam. **12**(3), 375–380 (1989)
35. Wen, J.T.-Y., Kreutz-Delgado, K.: The attitude control problem. IEEE Trans. Autom. Control **36**(10), 1148–1162 (1991)
36. Khalil, H.K.: Nonlinear Systems. Prentice-Hall, Upper Saddle River, NJ, USA (2002)
37. Gravish, N., Wood, R.J.: Anomalous yaw torque generation from passively pitching wings. In: Proc. IEEE Int. Conf. Robot. Automat. (ICRA), Stockholm, Sweden, pp. 3282–3287 (2016)
38. Murray, R.M., Deno, D.C., Pister, K.S.J., Sastry, S.S.: Control Primitives for Robot Systems. IEEE Trans. Syst. Man Cybern. **22**(1), 183–193 (1992)
39. Schaal, S.: Dynamic movement primitives–A framework for motor control in humans and humanoid robotics. In: Kimura, H., Tsuchiya, K., Ishiguro, A., Witte, H. (eds.) Adaptive Motion of Animals and Machines, pp. 261–280. Springer, Tokyo, Japan (2006)
40. Reed, M.C.: Fundamental Ideas of Analysis. Wiley, Hoboken, NJ, USA (1998)

Ryan M. Bena received his B.S. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2012 and his M.S. degree in aerospace engineering from the University of Southern California (USC), Los Angeles, CA, USA, in 2018. From 2012 to 2019, he was an aerospace engineer with the US Air Force. Mr. Bena is currently pursuing his Ph.D. degree in aerospace engineering at USC, working in the Autonomous Microrobotic Systems Laboratory (AMSL). His main research areas of interest include nonlinear controller design and analysis, embedded automation of microrobotic systems, and optimization-based control techniques.

Xuan-Truc Nguyen earned her B.S. degree in mechanical engineering from Yale University, New Haven, CT, USA, in 2016, and her M.S. and Ph.D. degrees in mechanical engineering from the University of Southern California (USC), Los Angeles, CA, USA, in 2018 and 2021, respectively. As a member of the Autonomous Microrobotic Systems Laboratory (AMSL), Dr. Nguyen specialized in novel actuation methods to develop microscale artificial insects in the pursuit of fully autonomous microrobots.

Xiufeng Yang received his B.S. and M.S. degrees in mechanical design, manufacturing, and automation from Huazhong University of Science and Technology, Wuhan, China, in 2010 and 2012, respectively, and his M.S. and Ph.D. degrees in mechanical engineering from the University of Southern California, (USC), Los Angeles, CA, in 2015 and 2020, respectively. From August 2014 to May 2020, he was a graduate researcher with the USC Autonomous Microrobotic Systems Laboratory (AMSL). He is currently a research and development (R&D) engineer with Applied Materials, Santa Clara, CA. Dr. Yang is a member of the IEEE Robotics and Automation Society. His main research areas of interest include microrobotics, actuators and sensors, microfabrication, real-time feedback control, and mechatronics.

Ariel A. Calderón received his B.S. and Engineer degrees in mechanical engineering from Universidad de Chile, Santiago, Chile, in 2010 and 2012, respectively, and his M.S. and Ph.D. degrees in mechanical engineering from the University of Southern California (USC), Los Angeles, CA, USA, in 2017 and 2020, respectively. From August 2014 to September 2020, he was a graduate researcher with the USC Autonomous Microrobotic Systems Laboratory (AMSL). From October 2020 until June 2021, he was a postdoctoral scholar in the Department of Civil and Environmental Engineering at the University of California, Los Angeles (UCLA), Los Angeles, CA, USA. Presently, he is a senior mechanical design engineer at Intuitive Surgical Inc., Sunnyvale, CA, USA. His main research areas of interest include microrobotics, actuators and sensors, microfabrication, real-time feedback control, soft robotics, and medical devices.

Ying Chen received his B.S. degree in mechanical engineering from Beihang University, Beijing, China, in 2012, and his M.S. and Ph.D. degrees in mechanical engineering from the University of Southern California (USC), Los Angeles, CA, USA, in 2014 and 2019, respectively. From August 2014 to September 2019, he was a graduate researcher with the USC Autonomous Microrobotic Systems Laboratory (AMSL). Since October 2019, he has been with Intuitive Surgical Inc., Sunnyvale, CA, USA. His research interests include control theory, autonomous aerial robotic systems, microrobotics, and biologically-inspired engineering.

Néstor O. Pérez-Arancibia received his Ph.D. degree in mechanical engineering from the University of California, Los Angeles (UCLA), Los Angeles, CA, USA, in 2007. From October 2007 to March 2010, he was a Postdoctoral Scholar with the Laser Beam Control Laboratory and the Mechatronics and Controls Laboratory in the Mechanical and Aerospace Engineering Department at UCLA. From April 2010 to March 2013, he was a Postdoctoral Fellow, and from April 2013 to August 2013, he was a Research Associate with the Microrobotics Laboratory and the Wyss Institute for Biologically Inspired Engineering at Harvard University, Cambridge, MA, USA. From August 2013 to August 2021, he was a faculty member of the Department of Aerospace and Mechanical Engineering at the University of Southern California (USC), Los Angeles, CA, USA, where he founded and directed the Autonomous Microrobotic Systems Laboratory (AMSL). Since August 2021, Dr. Pérez-Arancibia has been a Flaherty Associate Professor in Engineering with the School of Mechanical and Materials Engineering at Washington State University (WSU), Pullman, WA, USA, where he continues to lead the AMSL. Currently, he is also an Associate Editor for Mechatronics, a journal of the International Federation of Automatic Control (IFAC). Many of Dr. Pérez-Arancibia's research papers have been nominated for or won best paper awards. Most recently, he won a 2020 IEEE Robotics and Automation Letters Best Paper Award. His current research and teaching interests include feedback control, mechatronics, microrobotics, soft robotics, and biologically-inspired engineering.