High-Performance Six-DOF Flight Control of the Bee++: An Inclined-Stroke-Plane Approach

Ryan M. Bena ⬤, Xiufeng Yang ⬤, Ariel A. Calderón ⬤, and Néstor O. Pérez-Arancibia ⬤

Abstract—We present a new method for synthesizing and implementing high-performance *six-degree-of-freedom* (6-DOF) flight controllers for the Bee++, an insect-scale flying robot driven by four independently-actuated flapping wings. Each wing of the Bee++ is installed with a preset orientation such that the stroke plane generated during flight is inclined, thus enabling reliable roll, pitch, and yaw torque generation. Leveraging this capability, we propose a Lyapunov-based nonlinear control architecture that enables closed-loop position and attitude regulation and tracking. The control algorithms presented in this article simultaneously stabilize position and attitude by independently varying the wingstroke amplitudes of the four flapping wings of the Bee++. We use this particular control architecture to exemplify the process of controller synthesis and real-time implementation; however, the aerodynamic design of the Bee++ is compatible with a great variety of control structures and performance objectives. As a main result, we present the first set of experimental data demonstrating sustained and robust high-performance tracking of a 6-DOF reference signal during flight at the insect scale, which has been a long-standing control problem in the field of flapping-wing microrobotics. Furthermore, using data obtained through a series of systematic flight tests, we show that the Bee++ can achieve the highest 6-DOF performance ever recorded for an insect-scale flapping-wing flying robot during sustained flight.

Index Terms—Automation at the micro/nano scales, biologically-inspired robots, micro/nano robots, underactuated robots.

I. INTRODUCTION

INSECT-SCALE flapping-wing flying robots have significant potential for executing a great variety of practical tasks, as

Ryan M. Bena is with the Department of Aerospace and Mechanical Engineering, University of Southern California (USC), Los Angeles, CA 90089-1453 USA (e-mail: bena@usc.edu).

Xiufeng Yang is with Applied Materials, Santa Clara, CA 95054-3299 USA (e-mail: xiufeng@usc.edu).

Ariel A. Calderón is with Intuitive Surgical Inc., Sunnyvale, CA 94086-5301 USA (e-mail: aacalder@usc.edu).

Néstor O. Pérez-Arancibia is with the School of Mechanical and Materials Engineering, Washington State University (WSU), Pullman, WA 99164-2920 USA (e-mail: n.perezarancibia@wsu.edu).
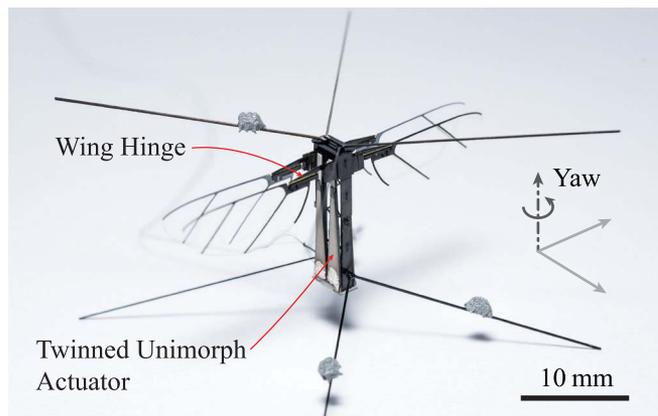
Fig. 1. Bee++ prototype used in the research presented in this article. This microrobot has an *inclined-stroke-plane* (ISP) configuration, weighs 95 mg, and measures 33 mm in wingspan. Eight anticollision protective spars and four 1.25-mg retroreflective markers were attached to the robot for motion tracking using a Vicon system.

can be inferred from observing the functionality and maneuverability of natural insects. However, state-of-the-art microrobotic insects have numerous fundamental limitations that must be overcome before their flight performance can match that exhibited by their biological counterparts. One of these limitations is the lack of high-performance control methods for all six degrees of freedom, corresponding to position in space and attitude, of the microflyers. Specifically, insect-scale flapping-wing flying robots have been unable to satisfactorily steer their yaw-angle *degree of freedom* (DOF) to desired values during sustained hovering flight. This is the primary challenge that motivated the research presented in this article, which we conducted using the Bee++, an improved new version of the Bee+ platform presented in [1] that can be effectively actuated and controlled using *inclined-stroke-plane* (ISP) methods of flapping. A photograph of a Bee++ prototype and a graphic definition of its yaw axis are shown in Fig. 1.

For flying insects such as flies and honeybees, yaw control plays an essential role in gaze stabilization during hovering flight, the execution of body saccades, and escape maneuvers [2]; and, since the flight control systems of these animals strongly rely on visual feedback [3], yaw maneuverability is one of their indispensable and more impressive capabilities. For example, a dipteran fly can adjust its heading direction by 90° in less than 50 ms, using visual and inertial feedback, by subtly varying the wingstroke amplitudes and stroke-plane angles of its flapping wings through thirteen pairs of μm-scale muscles located at

the wing articulations [4], [5]. Similar active variations of the wing motions are executed by most dipterans to produce the asymmetric drag forces required to generate complex yaw maneuvers [6], [7]. Likewise, complete attitude maneuverability is essential for insect-scale flapping-wing flying robots of the Bee++ type to achieve high performance, especially in real-life applications such as artificial pollination, aerial manipulation, and vision-based reconnaissance and surveillance, which require precise and fast control of the robot's pointing direction.

Unfortunately, even the most advanced microrobotic actuation technologies have yet to replicate the complex wing-hinge and transmission mechanisms of dipterans, hymenopterans, and many flying insects belonging to other orders, undoubtedly some of the most sophisticated mechanical joints found in nature [5], [8]. This technological limitation has compelled roboticists to develop simpler alternative solutions and, as a consequence, the vast majority of flapping-wing robotic insects reported to date have employed simplified flexure two-bar linkages as wing hinges and four-bar linkages as transmissions [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. As a result, the wings of these flyers can only pitch passively about their respective hinge axes. This underactuated flapping-wing design makes it impossible for state-of-the-art robotic insects to use bioinspired strategies for yaw-torque generation similar to those evolved by natural flying insects. In addition to the new ISP-based approach presented in this article, two other flapping-wing-based methods for yaw-torque generation and control at the insect scale have been proposed. The first and most widely discussed method is the use of asymmetric aerodynamic drag, an approach that is implementable via *split-cycling* [20], [21] and was first experimentally tested on the Harvard RoboBee [11], [13]. The second strategy is *wriggle-steering* [22], which consists of indirectly changing the yaw angle of the controlled flying vehicle through a combination of pitch and roll flight maneuvers. While the theory behind this proposed method is conceptually sound and intellectually attractive, to the best of our of knowledge, no flying robot has been demonstrated capable of generating reliable yaw torque using this technique during flight experiments. Therefore, we do not discuss this technique any further in this article.

The split-cycling technique consists of driving each wing of the controlled flyer with a time-asymmetric periodic signal, thus producing upstroke and downstroke velocities with different magnitudes within one flapping cycle, resulting in a nonzero cycle-averaged drag force acting on each wing. In this manner, body-yaw torque can be generated by varying the drag forces acting on the left and right wings of the robot. However, despite its elegance and apparent simplicity, the actual implementation of this method is quite complex and not entirely feasible with current microrobotic actuation technology. The main limitation in terms of actuation is the relatively narrow bandwidth of the piezoelectric-based flapping mechanisms employed by most flying microrobots, which causes the high-frequency content of the wingbeat, required by the split-cycling method, to be significantly damped. Furthermore, in studies of microrobotic thrust-generating mechanisms that rely on the passive pitching of flapping wings, it has been observed that the instantaneous drag forces acting on the wings do not reliably correlate with the corresponding flapping speeds [23]. Consistent with these

experimental findings, the difficulties of demonstrating repeatable controlled yaw maneuvers during real-time flight tests have been reported in [12] and [24]. To address this issue, the research presented in [16] proposed a modified version of the two-wing RoboBee design that can generate significantly larger yaw torques than those of the original robot in [11], and thus mitigate the high-frequency attenuation effects produced by the narrow bandwidths of the system's flapping mechanisms. However, in that approach, the larger measured yaw torques are generated using larger flapping amplitudes, at the cost of using lower flapping frequencies, resulting in a diminished thrust production that is not sufficient to achieve flight.

Following similar approaches based on asymmetric drag, roboticists have also developed and implemented yaw-motion control methods for several other insect-scale and larger platforms. The 143-mg microrobot presented in [15], a four-wing flyer that resembles the configuration of a quadrotor, employs the split-cycling method and simultaneously amplifies the generation of yaw torque by using wings placed at relatively long distances from the system's *center of mass* (COM), significantly lengthening the yaw-moment arms. The 155-mg modules that compose the insect-scale soft-actuated robots presented in [17] have a design that allows for the adjustment of the mean pitching of their flapping wings to produce a constant drag asymmetry during passively-stable flight. Unfortunately neither of these last two types of flyers, nor any other insect-scale flying robot, has conclusively demonstrated the ability to control the yaw DOF using these two asymmetric-drag-based methods in closed loop during flight. In contrast, both the 19-g nano hummingbird presented in [25] and the 12-g hummingbird-inspired robot presented in [26] use asymmetric-drag-based techniques to effectively and reliably produce yaw torques; however, the unique challenges at the insect size, such as limited actuator bandwidth, have prevented these approaches from being translated into the subgram scale.

Experimental data show that the most promising asymmetric-drag-based techniques for the insect-scale case are those introduced in [14] and [19]. Specifically, the *quad actuator bee* (QAB) in [14] is aerodynamically driven by two wings, and each wing is flapped by a power actuator and real-time adjusted by an additional independent actuator that regulates its mean pitching angle. Notably, the QAB was the first insect-scale flapping-wing flying robot to successfully demonstrate yaw stability; however, with an average *root-mean-square* (RMS) value of $38°$ for the control error during basic regulation, the flight performance achieved by this flyer is insufficient to accomplish most envisioned practical tasks. The Robofly-Expanded presented in [19] was experimentally demonstrated to control its yaw DOF using feedback and the standard split-cycling method during brief flight tests with durations of one to two seconds; however, results demonstrating this capability during sustained hovering flight were not reported. While the results in [19] are valuable and promising, the importance of data obtained during sustained flight should not be underestimated. Short-duration flight experiments are inadequate to assess the experimental stability of insect-scale flapping-wing flying robots because systems of this type may exhibit slow destabilizing behaviors [1].
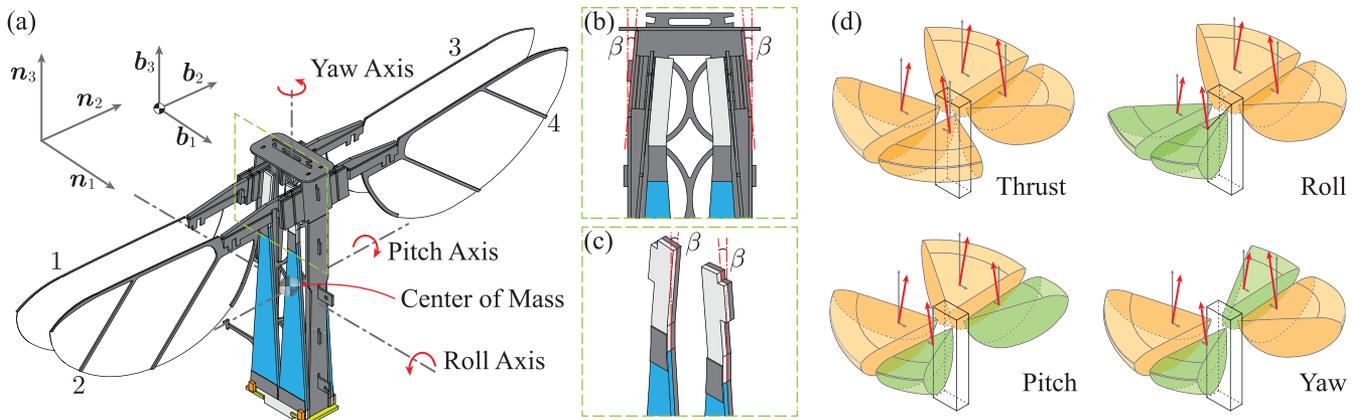
Fig. 2. Schematic diagrams that depict the design and functionality of the four-wing four-actuator Bee^{++}. (a) This illustration defines the inertial frame $\mathcal{N}$, with axes $\{n_1, n_2, n_3\}$, and the body-fixed frame $\mathcal{B}$, with axes $\{b_1, b_2, b_3\}$ (shifted for clarity), and labels the wings of the robot using numbers (from 1 to 4). As shown, the axes of $\mathcal{B}$ coincide with the three body axes of rotation, roll, pitch, and yaw. (b) Depiction of a zoomed-in side view of the top section of the Bee^{++} design. (c) Depiction of a zoomed-in view of the inclined tips of a pair of twinned unimorph actuators. (d) Flapping modes for thrust and torque generation. As shown, thrust is modulated by simultaneously varying the flapping amplitude of the four wings of the robot. Roll torque is generated and modulated by varying the difference in flapping amplitude between the right (from the flyer's perspective) pair of wings, $\{1, 2\}$, and the left (from the flyer's perspective) pair of wings, $\{3, 4\}$. Pitch torque is generated and modulated by varying the difference in flapping amplitude between the front pair of wings, $\{2, 4\}$, and the back pair of wings, $\{1, 3\}$. Torque about the yaw axis, b_3, is produced by simultaneously and coordinately varying the flapping amplitudes of the pairs of wings in the diagonals of the b_1-b_2 plane; specifically, to generate yaw torque, the flapping amplitudes of the wing pair $\{1, 4\}$ are varied in unison and maintained equal but with a difference with respect to those of the wing pair $\{2, 3\}$. This latter flapping pattern generates a yaw torque because, due to the ISP configuration of the Bee^{++} design, two force projections in the b_1-b_2 plane with equal magnitudes and opposite directions are produced at each side of the b_1-b_3 plane. The red arrows illustrate the idealized directions and relative magnitudes of the generated forces. These force vectors and corresponding flapping amplitudes are not to scale.

To effectively address the described long-standing problem of yaw-torque generation at the insect scale, here we introduce a new ISP-based method for the production and control of aerodynamic thrust via flapping. Essentially, with this proposed approach, we solved a control problem using both control theory and robotic design. The first insect-scale flyer capable of generating yaw torques using this method was the four-wing four-actuator Bee^{+}, which we introduced in [1]; however, despite obtaining promising preliminary results, we were unable to conclusively demonstrate ISP-based yaw control due to this robot's limited thrust-generation capabilities. As already mentioned, in the research presented in this article, we used a high-thrust Bee^{++} prototype, whose new features and capabilities we first presented and discussed in [27]. A detailed illustration of this robot's design is shown in Fig. 2(a). This robotic design was inspired by the observation that most natural insects, such as the fruit fly (*Drosophila melanogaster*), for control purposes during normal hovering and aerobatic flight, vary in real time the orientations of their stroke planes of flapping [2]. By using time-varying stroke planes, the aerodynamic forces produced by the flapping wings of a flyer can be directed to create thrust and body-torque vectors with infinitely many possible magnitudes and directions. Unfortunately, the technology required to fabricate micromechanisms capable of varying the wingstroke planes of insect-scale wings has yet to be developed.

Fortunately, as unequivocally shown in this article, Bee^{++} prototypes can effectively generate yaw torques sufficiently large to control the yaw DOF during flight, using fixed stroke planes. The robotic design of the Bee^{++} is endowed with this ability because it features four independently-actuated wings that are installed with fixed tilted angles to generate stroke

planes inclined with respect to the plane perpendicular to the main direction of thrust generation during flapping. As shown in the experimental section of this article, given the unique conceptual design and physical configuration of the Bee^{++}, simply by coordinately varying the flapping amplitudes of the system's wings, sufficient yaw torque for mobility and control can be generated. Furthermore, Table I clearly shows that the novel design of the Bee^{++} enables the best controlled yaw maneuvers performed by an insect-scale flapping-wing flying robot during sustained flight, published to date. In the past, torque-generation strategies for control that are based on ISP techniques have been developed for and implemented on flapping-wing flyers with larger-than-insect scales; for example, see [28]. Those robots, however, have sizes and designs vastly dissimilar to that of the Bee^{++}, and also employ significantly different actuation technologies; in fact, to the best of our knowledge, never before has an ISP-based approach been successfully implemented at the insect scale.

The rest of the article is organized as follows. First, Section II describes the most relevant aspects of the Bee^{++} design and the quasi-steady aerodynamic analysis that, to a significant extent, motivated and justified this design approach. Next, Section III presents the methods for the synthesis and implementation of the proposed *quaternion*-based flight controller that enables simultaneous position and yaw tracking for the Bee^{++} using ISP-based actuation. Section IV presents experimental results obtained to systematically assess the flight performance of the tested Bee^{++} prototype. These results conclusively validate the proposed method for controlling the yaw DOF of the Bee^{++}, and similar insect-scale flapping-wing flyers, using ISP-based actuation. Last, Section V presents conclusions and discusses

TABLE I
COMPARISON OF YAW-TORQUE-GENERATION METHODS FOR FLAPPING-WING FLYING MICROROBOTS AND ACHIEVED PERFORMANCE

Robot Name	Method	Number of Actuators	Open Loop	Closed Loop	In-Flight	Sustained	RMS Value of Error (Deg)
Bee⁺⁺	ISP	4	✓	✓	✓	✓	9°
QAB [14]	AD	4	✓	✓	✓	✓	38°
Robofly-Expanded [19]	AD	2	✓	✓	✓		10°
RoboBee [12], [16], [22]	AD, WS	2	✓	✓			NR
Four Wings [15]	AD	4	✓				NR
DEA Flyer [17]	AD	1, 2, 4	✓				NR
Bee⁺ [1]	ISP	4					NR
RoboBee XWing [18]	AD	2					NR

Note 1: This table compares all eight different state-of-the-art flapping-wing flying microrobots and the respective methods employed to produce yaw torque. The yaw-torque-generation methods are denoted as *inclined-stroke-plane* (ISP), *asymmetric drag* (AD), and *wriggle-steering* (WS). Performance is considered to be *sustained* if the reported flight test lasted longer than five seconds (5 s). The RMS value of the control error reported for the Bee⁺⁺ is the average over eight 20-s flight experiments. If a cited paper does not report quantifiable yaw-control data, the table indicates *not reported* (NR).

directions for future research.

Notation:

1) Regular lowercase symbols denote scalars, e.g., p; bold lowercase symbols denote vectors, e.g., $\boldsymbol{p}$; bold uppercase symbols denote matrices, e.g., $\boldsymbol{P}$; bold crossed lowercase symbols denote quaternions, e.g., $\boldsymbol{\mathsf{p}}$.

2) The variable t is used to denote continuous time.

3) The dot operator is used to denote differentiation with respect to time, e.g., $\dot{p} = \frac{dp}{dt}$. Consistently, multiple dots are used to indicate higher order derivatives.

4) The symbol $\times$ denotes the cross-product operation between vectors.

5) The symbol $\otimes$ denotes quaternion multiplication.

6) The symbol $\| \cdot \|_p$ denotes the p-norm of a vector.

7) The symbol $\mathrm{tr}\{\cdot\}$ denotes the trace operator for matrices.

8) The symbol $\mathrm{sgn}\{\cdot\}$ denotes the sign operator for scalars.

9) The symbols $>, <, \geq,$ and $\leq$ denote definiteness relationships when used with matrices. These same symbols denote ordering relationships when used with scalars.

10) The symbols $\sum$ and $\prod$ denote the summation and product of multiple elements, respectively.

II. ROBOTIC DESIGN AND ANALYSIS

A. Robotic Design

The Bee⁺⁺ design has many advantages regarding flight control, aerodynamic performance, robotic design, and microfabrication when compared to other insect-scale flapping-wing flying robots, as already discussed in [27]. Specifically, the unique four-wing configuration of the Bee⁺⁺ platform makes possible the implementation of ISP-based methods for complete *three-degree-of-freedom* (3-DOF) attitude-torque generation and control, even though this design is underactuated and generates aerodynamic forces using passive wing-pitching as most other flapping-wing flying microrobots. As seen in Fig. 1, the Bee⁺⁺ maintains a structural design and configuration very similar to those of the Bee⁺ first introduced in [1]. Also, the new robot still weighs 95 mg (excluding the reflective markers used for motion capture) and measures 33 mm in wingspan because, as explained in [27], the modifications applied to the original Bee⁺ design that resulted in the Bee⁺⁺, aimed to augment thrust

generation and control authority, did not cause any increases in body weight or overall size.

The most relevant features of the Bee⁺⁺ are depicted in the schematic views of Fig. 2. Here, Fig. 2(a) graphically defines the inertial frame of reference $\mathcal{N}$, with axes $\{\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3\}$, and the body-fixed frame $\mathcal{B}$, with axes $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$. With respect to their location in this figure, the wings to the left of the $\boldsymbol{b}_1$-$\boldsymbol{b}_3$ plane are labeled as 1 and 2, and the wings to the right of the $\boldsymbol{b}_1$-$\boldsymbol{b}_3$ plane are labeled as 3 and 4. Fig. 2(b) shows how the combined hinge-transmission mechanism that drives each wing is installed with an inwardly-inclined angle, β, which generates a symmetric structural feature with respect to the $\boldsymbol{b}_2$-$\boldsymbol{b}_3$ plane. This β-inclined installation of the wings is enabled by a simple design element, the center-top structural beam of the airframe with the shape of an isosceles trapezoid. Consistent with this configuration, the front and back pillars of the airframe are machined with kiss-cut folding lines in order to facilitate their smooth bending and thus create the inclined surfaces to which the transmissions are attached. Also, the airframe pillars have assembly slots to enforce the designed alignment of the components and a structural grid provides a stable base for ensuring symmetry, with respect to both the $\boldsymbol{b}_1$-$\boldsymbol{b}_3$ and $\boldsymbol{b}_2$-$\boldsymbol{b}_3$ planes, during the installation of the wings.

Additionally, since the transmission mechanisms and wings are installed with a preset desired inclination with respect to the $\boldsymbol{b}_1$-$\boldsymbol{b}_2$ plane, we designed the piezoelectric actuators of the robot with special characteristics that enable a robust mechanical connection between their distal ends and the transmissions that drive the wings of the system. Specifically, in contrast with the design of the original Bee⁺ that has piezoelectric actuators with straight tips, the new actuators developed for the Bee⁺⁺ are fabricated with tilted tips, as depicted in Fig. 2(c). As seen here, the tip of each Bee⁺⁺ actuator has exactly the same inclination β as the airframe pillars, which facilitates the adhesion of the surfaces in contact, thus creating robust mechanical connections between the actuators and transmissions of the robot. It is important to emphasize that this innovation in design was essential to achieving the yaw-torque production required for effective control of the yaw DOF. In fact, any geometric mismatch between the actuator tip and the linkage-transmission mechanism of a wing would invariably cause undesired twisting motions and thus introduce

unwanted nonlinear dynamics to the flapping-wing actuation system. The actuators of the Bee++ are unimorph and fabricated in twin pairs using exactly the same process we presented in [1]. This fabrication technology allows us to readily make unimorph actuators with single or twinned configurations, with rectangle or trapezoid shapes, and with inclined or straight tips.

The original Bee+ can achieve a maximum thrust-to-weight ratio of 1.4, which is sufficient for taking off and hovering but not enough for controlled positional maneuvering. To improve thrust and torque generation, we implemented the solution presented in [29], which is based on selecting the parameters of the wings and flapping mechanisms that maximize the operating frequency of the flyer. Specifically, by increasing the thickness of the main intermediate link of the four-bar-linkage transmission mechanism (for details, see [27]), we decreased its transmission ratio from $3509\,\mathrm{rad}\cdot\mathrm{m}^{-1}$ to $2694\,\mathrm{rad}\cdot\mathrm{m}^{-1}$. Additionally, we augmented the width of the passive flexure wing-hinges, thus increasing their stiffness to an estimated $2.3\times10^{-6}\,\mathrm{N}\cdot\mathrm{m}\cdot\mathrm{rad}^{-1}$ from $1.4\times10^{-6}\,\mathrm{N}\cdot\mathrm{m}\cdot\mathrm{rad}^{-1}$ according to the method in [7], which represents an increment of $64\,\%$. These two design modifications increased the overall stiffness of the flapping-wing actuation system, which resulted in a significant improvement in operating bandwidth. Specifically, the newly-designed Bee++ has a resonant frequency in excess of $165\,\mathrm{Hz}$, which represents an increase of about $50\,\%$ with respect to that of the original Bee+, whose measured natural frequency was in the order of $110\,\mathrm{Hz}$ [1].

Overall, the proposed ISP-based robotic design, as explained in Section II-B, enables the Bee++ to directly adjust the three-component aerodynamic-torque vector acting on its body by simply varying the flapping amplitudes of its four wings. In contrast with the wriggle-steering method in [22], ISP-based yaw-torque generation is, to a great extent, dynamically decoupled from the production of aerodynamic roll and pitch torques. More importantly, unlike in the case of split-cycling-based methods, ISP-based yaw-torque generation does not require the robot to be driven by signals with high-frequency harmonics. This characteristic of ISP-based methods is extremely relevant because its use significantly reduces the mechanical burden on the robotic components and, therefore, the expected time to reach structural fatigue is significantly extended [30].

B. Aerodynamic Analysis

The illustration in Fig. 2(d) explains the method employed to generate the roll, pitch, yaw, and altitude control inputs for the four-wing Bee++. This technique, in combination with the aforementioned improved aerodynamic and robotic designs, enables agile attitude maneuverability of the new microrobot, which has been thus far unattainable by its two-wing counterparts. In order to estimate the amount of yaw torque produced using the proposed ISP-based method, we leverage the simplified aerodynamic analysis discussed in [1], which we briefly review here. First, note that the magnitude of the cycle-averaged aerodynamic force generated by the ith flapping wing, $\bar{f}_i$, for $i\in\{1,2,3,4\}$, can be estimated, via the quasi-steady method, as

$$\bar{f}_i = C(\bar{\alpha})\nu_i^2 a_i^2 S_{\mathrm{w}}, \tag{1}$$

where $C(\bar{\alpha})$ is a generalized lumped aerodynamic coefficient; $\bar{\alpha}$ is the cycle-averaged angle of attack; ν_i is the frequency of flapping; a_i is the amplitude of flapping; and S_{w} is the total nominal area of the wing.

As depicted in Fig. 2(d), assuming symmetry for the purposes of analysis, the direction of the cycle-averaged aerodynamic-force vector of each wing is approximately orthogonal to the corresponding wingstroke plane. Thus, when the stroke plane is aligned with the b_1-b_2 plane, the direction of the aerodynamic-thrust vector is approximately aligned with the direction of b_3. Since, in the case of the Bee++, the stroke planes swept by the four wings are inclined by an angle β, a force component in the b_1-b_2 plane, with averaged magnitude $\bar{f}_i\sin\beta$, is produced by the ith wing of the flyer. Because this force component acts at a distance from the COM of the robot, it also generates a torque about the b_3 axis. Consequently, when a Bee++ prototype is in a stable horizontal orientation (i.e., b_3 is aligned with n_3) and its four wings are flapped in coordinated diagonal pairs, the result is a pure yaw rotation. Assuming that all four wings are symmetrically inclined, the value (signed magnitude) of the total yaw torque produced by the flyer is

$$\tau_3 = \left(\bar{f}_1 - \bar{f}_2 - \bar{f}_3 + \bar{f}_4\right) d_3\sin\beta, \tag{2}$$

where, assuming symmetry, d_3 is the perpendicular distance from the b_3 axis to the aerodynamic-force center of each wing.

For the particular design of the Bee++ prototype used in the experiments presented here, the ISP angle, β, was chosen to be $5°$, which is similar to the wing-tilting angle of a dipteran fly during yaw maneuvering [4]. Thus, at any given moment during flight, only about $9\,\%\,(\sin\beta)$ of the cycle-averaged aerodynamic force produced by each wing, $\bar{f}_i$, for $i\in\{1,2,3,4\}$, contributes to the generation of yaw torque. This implies that for the proposed ISP method to work, a sufficiently large amount of thrust force must be continually generated. Aerodynamic simulations based on the methods presented in [6], and calibrated using the experimental data and quasi-steady analyses in [1], indicate that the Bee++ platform can generate approximately $3.2\,\mathrm{mN}$ of thrust while flapping its four wings at a frequency of $165\,\mathrm{Hz}$ with wingstrokes of $65°$. Using these simulation data and (1), we further predict that, theoretically, a Bee++ prototype can generate thrust forces in the order of $4.3\,\mathrm{mN}$ for the four wings flapping with wingstrokes of $75°$ (the maximum experimental value observed at $150\,\mathrm{Hz}$); however, at $165\,\mathrm{Hz}$, the maximum observed wingstrokes do not surpass $65°$ and, therefore, for the rest of the article we consider the value of $3.2\,\mathrm{mN}$ to be the theoretical limit for the thrust force that the robot can generate during flight. This value represents an improvement of $125\,\%$ over earlier flyers of the Bee+ type, which were measured to produce maximum thrust forces in the order of $1.4\,\mathrm{mN}$ [1]. This increase in thrust-force production is explained by the shift of the resonant frequency of the flapping-wing system from about $110\,\mathrm{Hz}$ to more than $165\,\mathrm{Hz}$. These values are consistent with the data reported in [29].

Similarly, at the flapping frequency of $150\,\mathrm{Hz}$, aerodynamic simulations predict a total cycle-averaged thrust force of $2.2\,\mathrm{mN}$, for the four wings of the robot flapping with a wingstroke of
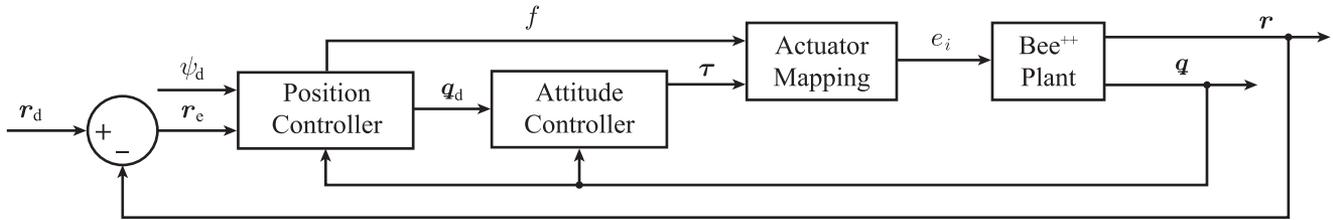
Fig. 3. Block diagram of the control architecture used to fly the tested Bee⁺⁺ prototype. This control scheme is composed of two main subsystems: a *position controller* and an *attitude controller*. The position controller receives as inputs the position error, $r_e = r_d - r$, the measured attitude quaternion, q, and the yaw-angle reference, ψ_d; then, this block generates as outputs the instantaneous force value, f, and attitude-reference quaternion, q_d, required to reach the position reference, r_d. The attitude controller receives as inputs the attitude-reference quaternion, q_d, and the measured attitude quaternion, q; then, this block generates as outputs the body torque, τ, required to align the yaw axis, b_3, of the robot with the desired thrust force, f_a, and also reach the yaw-angle reference, ψ_d. Last, the *actuator mapping* converts the control signals, f and τ, into the electrical voltages that are used to excite the actuators of the controlled Bee⁺⁺ prototype.

$65°$, and of 2.9 mN for the maximum experimentally-achievable wingstroke of $75°$. The yaw-torque lever-arm of the new Bee⁺⁺ prototype, d_3, is 9.3 mm; therefore, from (2), it directly follows that the maximum achievable yaw torque at a flapping frequency of 150 Hz is predicted to be approximately $1.18 \times 10^{-6} \, \text{N} \cdot \text{m}$. This condition is theoretically reached when wings 1 and 4 are flapped at their maximum achievable wingstroke angles ($75°$), and wings 2 and 3 stop flapping ($0°$), or vice versa. For the purpose of comparison, note that the experiment-based study in [16] predicts the generation of a maximum yaw torque of $2.5 \times 10^{-6} \, \text{N} \cdot \text{m}$ for the Harvard RoboBee, when the split-cycling method is employed during grounded experiments; unfortunately, conclusive in-flight results validating these findings have not been reported to date. Evidently, the maximum yaw torque achievable by a Bee⁺⁺ prototype cannot be sustained during stable controlled flight because the robot must use its flapping wings to simultaneously maintain altitude, roll, pitch, and yaw stability. This drawback arises because the Bee⁺⁺ is an underactuated system despite having two more actuators than the RoboBee. A brief assessment of the impact of underactuation on the flight performance of the flyer is discussed in Section IV.

Last, it should be noted that, as already hinted above, ISP-based yaw-torque generation is expected to inject a small cycle-averaged aerodynamic force that is not aligned with the yaw axis, b_3, of the robot. As seen in Fig. 2(d), in the pitch-torque generation mode, the net force component along the roll axis, b_1, resulting from adding the four aerodynamic forces produced by the flapping wings, is nonzero. Based on the already-discussed estimation of the total thrust force produced by the system during flight, this parasitic force along the direction of b_1 can reach maximum values in the order of 0.14 mN; fortunately, because of its direction, its effect is a small constructive addition to the generation of pitch torque. To see that the effect of this parasitic force is constructive and does not act as an undesired disturbance, consider the proposed control architecture presented in Section III. Specifically, note that body rolling and pitching are used to produce the horizontal forces required by the high-level position controller in the scheme shown in Fig. 3. Accordingly, since the parasitic force is always aligned with the aerodynamic force generated for control purposes, the result is always a net stabilizing effect.

III. SYNTHESIS OF 6-DOF FLIGHT CONTROLLER

To control the Bee⁺⁺ prototype developed for the flight experiments presented in this article, we designed and implemented the *six-degree-of-freedom* (6-DOF) feedback scheme shown in Fig. 3. The two main components in this configuration are the *position* and *attitude* controllers, which in combination control the six degrees of freedom of the robot, including the instantaneous yaw angle. The full implementation of this 6-DOF flight control architecture is enabled by the ISP-based method for the generation of yaw torque, already discussed in Section II. As seen in the block diagram of Fig. 3, the proposed control architecture uses coupled position and attitude algorithms that were derived using a rigid-body model of the Bee⁺⁺ dynamics. During flight, the desired forces and torques generated by the control algorithms are processed by a static *actuator mapping* that calculates and amplifies the voltage signals that are fed into the piezoelectric actuators of the robot. As a result of the control action of this feedback configuration, the closed-loop system can robustly operate in a stable and reliable manner in flight tests during which position-vector and yaw-angle references are effectively followed.

A. Rigid-Body Dynamics

Overall, the dynamic model for the Bee⁺⁺ that we present here is a reduced-complexity version of the more-generic non-vertical thrust model used in [31] to describe the dynamics of flyers. This simplified representation of the Bee⁺⁺ dynamics is intended for use in controller synthesis and has the same structure as that of the quadrotor *uncrewed aerial vehicle* (UAV) discussed in [32] and [33]. As conclusively demonstrated in Section IV, and also reported in [32] and [33], this rigid-body modeling approach is sufficiently accurate for designing controllers for four-propeller flyers. Thus, recalling the definitions of the inertial and body-fixed reference frames, $\mathcal{N}$ and $\mathcal{B}$, in Section II-A, invoking Newtonian mechanics to describe the flyer's translational dynamics, and using a formulation based on quaternions to describe the flyer's rotational motion, we can write a state-space representation of the system with the form

$$\dot{\boldsymbol{r}} = \boldsymbol{v},$$

$$\dot{\boldsymbol{v}} = \frac{f}{m}\boldsymbol{b}_3 - g\boldsymbol{n}_3,$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix},$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}\left(\boldsymbol{\tau} - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}\right),$$

(3)

where $\boldsymbol{r} = [r_1 \ r_2 \ r_3]^T$ and $\boldsymbol{v}$ are the instantaneous position and velocity, relative to $\mathcal{N}$, of the robot's COM, respectively; f is the component of the total thrust force produced by the robot that is aligned with the yaw axis, $\boldsymbol{b}_3$; m is the total mass of the robot; g is the acceleration of gravity; $\boldsymbol{q}$ is the quaternion used to describe the instantaneous orientation of the body-fixed frame $\mathcal{B}$ relative to the inertial frame $\mathcal{N}$; $\boldsymbol{\omega}$ is the angular velocity of $\mathcal{B}$ relative to $\mathcal{N}$, written with respect to $\mathcal{B}$; $\boldsymbol{J}$ is the inertia matrix of the system, written with respect to $\mathcal{B}$; and $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \tau_3]^T$ is the total aerodynamic torque produced by the robot, written with respect to $\mathcal{B}$. As defined in Section I, the symbol $\otimes$ denotes the standard quaternion multiplication. According to this model, as depicted in Fig. 3, the variables f and $\boldsymbol{\tau}$ are also the control inputs generated by the proposed feedback-control algorithms discussed in Section III-B. Last, note that to facilitate the process of controller synthesis, this dynamic model neglects the component of the cycle-averaged thrust force generated by the robot that is not aligned with $\boldsymbol{b}_3$, which consequently should be interpreted as a disturbance affecting the system.

B. Control Algorithms

The real-time algorithms proposed to simultaneously control the position and attitude of the Bee++ during flight are based on a two-stage method that sequentially computes the control signals f and $\boldsymbol{\tau}$, as depicted in Fig. 3. In this scheme, during the position-control stage, an algorithm uses a modified *proportional–integral–derivative* (PID) filter to attenuate the error between the desired instantaneous position, $\boldsymbol{r}_d$, and measured instantaneous position of the robot, $\boldsymbol{r}$, according to

$$\boldsymbol{f}_a(t) = \boldsymbol{K}_p\left[\boldsymbol{r}_d(t) - \boldsymbol{r}(t)\right] + \boldsymbol{K}_i\int_0^t \left[\boldsymbol{r}_d(\tau) - \boldsymbol{r}(\tau)\right]d\tau$$
$$+ \boldsymbol{K}_d\left[\dot{\boldsymbol{r}}_d(t) - \dot{\boldsymbol{r}}(t)\right] + m\ddot{\boldsymbol{r}}_d(t) + mg\boldsymbol{n}_3,$$

(4)

where $\boldsymbol{f}_a$ is the instantaneous force vector required to track $\boldsymbol{r}_d$, for time $t \geq 0$; and $\boldsymbol{K}_p$, $\boldsymbol{K}_i$, and $\boldsymbol{K}_d$ are positive diagonal controller-gain matrices. Note that, as already discussed in Section II, the Bee++ can reliably modulate only aerodynamic thrust generated along the $\boldsymbol{b}_3$ direction; therefore, for control purposes, we compute the component of $\boldsymbol{f}_a$ in the $\boldsymbol{b}_3$ direction, f, as

$$f(t) = \boldsymbol{f}_a^T(t)\boldsymbol{b}_3(t).$$

(5)

As shown in Fig. 3, f, computed according to (5), is the first control signal inputted to the actuator mapping in the control block diagram and assumed to be the *true* signed magnitude of the total aerodynamic thrust produced by the robot.

Next, in the attitude-control stage, an algorithm computes the spacial orientation required to align the yaw axis of the robot, $\boldsymbol{b}_3$, with the instantaneous direction of $\boldsymbol{f}_a$ and simultaneously reach the instantaneous yaw-angle reference, ψ_d. Specifically, the coordinate basis vectors, $\{\boldsymbol{b}_{1,d}, \boldsymbol{b}_{2,d}, \boldsymbol{b}_{3,d}\}$, representing the desired attitude of the robot are calculated according to

$$\boldsymbol{b}_{3,d} = \frac{\boldsymbol{f}_a}{\|\boldsymbol{f}_a\|_2},$$
$$\boldsymbol{b}_{1,d} = \frac{\left[-\sin\psi_d \ \cos\psi_d \ 0\right]^T \times \boldsymbol{b}_{3,d}}{\|\left[-\sin\psi_d \ \cos\psi_d \ 0\right]^T \times \boldsymbol{b}_{3,d}\|_2},$$
$$\boldsymbol{b}_{2,d} = \boldsymbol{b}_{3,d} \times \boldsymbol{b}_{1,d}.$$

(6)

Then, the desired attitude specified by (6) is converted into the equivalent reference-attitude quaternion, $\boldsymbol{q}_d$, which is employed to compute the attitude control error as

$$\boldsymbol{q}_e = \boldsymbol{q}^{-1} \otimes \boldsymbol{q}_d,$$

(7)

where $\boldsymbol{q}_e = [m_e \ \boldsymbol{n}_e^T]^T$ represents the attitude of the *desired* body-fixed frame of reference, $\mathcal{I}$, relative to the *true* body-fixed frame of reference, $\mathcal{B}$. Last, the aerodynamic torque in (3), $\boldsymbol{\tau}$, used for feedback control as shown in the scheme of Fig. 3, is computed according to the feedback law

$$\boldsymbol{\tau} = \boldsymbol{K}_q\boldsymbol{n}_e + \boldsymbol{K}_\omega\left(\boldsymbol{\omega}_d - \boldsymbol{\omega}\right) + \boldsymbol{J}\dot{\boldsymbol{\omega}}_d + \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega},$$

(8)

where $\boldsymbol{K}_q$ and $\boldsymbol{K}_\omega$ are controller-gain positive-definite matrices; and $\boldsymbol{\omega}_d$ is the angular-velocity reference for $\mathcal{B}$.

C. Actuator Mapping

As depicted in Fig. 3, after the control algorithms compute the force f and torque $\boldsymbol{\tau}$, the control inputs required to excite the dynamics of the controlled Bee++ during stable flight, these signals are mapped into the voltages that excite the four unimorph piezoelectric actuators of the flyer, using a static matrix operator as explained in [1] and [27]. Each unimorph actuator of the robot is excited using a sinusoidal voltage, e_i, for $i \in \{1, 2, 3, 4\}$, with a fixed frequency, ν_i, and a variable amplitude, γ_i. The output motion produced by the ith actuator of the system is mapped to the corresponding ith wing through a transmission, as described in [1]. The matrix-based approach used to model the actuator mapping of the closed-loop system combines the ideas on quasi-steady aerodynamic analysis in Section II-B with an empirical method commonly used in quadrotor research [32], [33]. Namely, the cycle-averaged thrust force generated by the ith flapping wing, for $i \in \{1, 2, 3, 4\}$, is estimated as

$$\bar{f}_i(t) = k_f\gamma_i(t),$$

(9)

where k_f is an experimentally-identified positive real constant.

Thus, assuming symmetry and recalling the variables defined in Section II-B, we obtain that for the Bee++, the *inverse* actuator mapping is specified by

$$\begin{bmatrix} f \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ -k_f d_1 & -k_f d_1 & k_f d_1 & k_f d_1 \\ k_f d_2 & -k_f d_2 & k_f d_2 & -k_f d_2 \\ k_s d_3 & -k_s d_3 & -k_s d_3 & k_s d_3 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix},$$

(10)

where d_1 and d_2 are the lever arms corresponding to the torques τ_1 and τ_2; and d_3 is the lever arm corresponding to the torque τ_3, already defined in Section II-B. Note that k_s directly depends on the angle β that determines the inclined stroke plane of each wing, according to $k_s = k_f \tan\beta$. Then, by simply inverting the

matrix specified by (10), we obtain the *direct* actuator mapping for the robot. Namely,

$$
\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k_\mathrm{f}} & -\frac{1}{4k_\mathrm{f}d_1} & \frac{1}{4k_\mathrm{f}d_2} & \frac{1}{4k_\mathrm{s}d_3} \\ \frac{1}{4k_\mathrm{f}} & -\frac{1}{4k_\mathrm{f}d_1} & -\frac{1}{4k_\mathrm{f}d_2} & -\frac{1}{4k_\mathrm{s}d_3} \\ \frac{1}{4k_\mathrm{f}} & \frac{1}{4k_\mathrm{f}d_1} & \frac{1}{4k_\mathrm{f}d_2} & -\frac{1}{4k_\mathrm{s}d_3} \\ \frac{1}{4k_\mathrm{f}} & \frac{1}{4k_\mathrm{f}d_1} & -\frac{1}{4k_\mathrm{f}d_2} & \frac{1}{4k_\mathrm{s}d_3} \end{bmatrix} \begin{bmatrix} f \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} .
\tag{11}
$$

It is important to reiterate that this mapping is utilized for real-time implementation of the control algorithms that enable Bee++ prototypes to fly; specifically, this mapping receives as inputs the control signals, $\{f, \tau_1, \tau_2, \tau_3\}$, and outputs the amplitudes γ_i required to generate the voltages e_i, for $i \in \{1, 2, 3, 4\}$, used to excite the four actuators of the system.

D. Stability Analysis of the Closed-Loop System

Motivated by the analyses of spacecraft dynamics presented in [34] and [35], and the discussion on the general attitude-stability control problem in [36], we employ a standard Lyapunov-based methodology to analyze and enforce the closed-loop flight-attitude stability of the Bee++ prototype used in the control experiments presented here. Following this approach, the closed-loop attitude states are chosen to be the attitude tracking error, q_e, and the angular-velocity tracking error, $\omega_\mathrm{e} = \omega_\mathrm{d} - \omega$. Thus, taking the time-derivative of the expression specified by (7) yields a state-space representation of the attitude tracking-error dynamics of the system. Similarly, plugging the expression of the aerodynamic torque used for control, specified by (8), into the open-loop state-space representation, specified by (3), and then solving for $\dot{\omega}_\mathrm{e}$ yields a state-space representation of the angular-velocity tracking-error dynamics of the system. As a result, a state-space representation of the complete closed-loop attitude dynamics of the system can be specified as

$$
\dot{q}_\mathrm{e} = \frac{1}{2} \begin{bmatrix} 0 \\ \omega_\mathrm{e} \end{bmatrix} \otimes q_\mathrm{e},
$$
$$
\dot{\omega}_\mathrm{e} = -J^{-1} \left(K_q n_\mathrm{e} + K_\omega \omega_\mathrm{e} \right).
\tag{12}
$$

It can be easily shown that the multivariable functions on the right side of the autonomous system specified by (12) are continuous with a locally-bounded 7×7 Jacobian matrix, A, defined as

$$
A\left(q_\mathrm{e}, \omega_\mathrm{e}\right) = \begin{bmatrix} 0 & -\frac{1}{2}\omega_\mathrm{e}^T & -\frac{1}{2}n_\mathrm{e}^T \\ \frac{1}{2}\omega_\mathrm{e} & -\frac{1}{2}\Omega_\mathrm{e} & \frac{1}{2}\left(N_\mathrm{e} + m_\mathrm{e}I_{3\times3}\right) \\ 0_{3\times1} & -J^{-1}K_q & -J^{-1}K_\omega \end{bmatrix},
\tag{13}
$$

where Ω_e is the skew-symmetric cross-product matrix, as defined in [34], formed from the components of ω_e; and N_e is the skew-symmetric cross-product matrix, as defined in [34], formed from the components of n_e. The continuity of the functions on the right side of (12) and local boundedness of the matrix in (13) imply that the multivariable functions on the right side of the closed-loop state-space representation of the complete attitude dynamics of the system, specified by (12), are locally Lipschitz. Furthermore, by setting $\dot{q}_\mathrm{e}$ and $\dot{\omega}_\mathrm{e}$ equal to

zero and solving for all possible values of q_e and ω_e, it can be determined that there are only two equilibrium points, one unstable and one stable. The instability of the first equilibrium point, corresponding to $q_\mathrm{e}^\dagger = [-1 \ 0 \ 0 \ 0]^T$ and $\omega_\mathrm{e}^\dagger = [0 \ 0 \ 0]^T$, can be easily proven by invoking *Lyapunov's indirect method* [37]. Specifically, plugging $q_\mathrm{e}^\dagger$ and $\omega_\mathrm{e}^\dagger$ into the Jacobian matrix in (13) yields

$$
A\left(q_\mathrm{e}^\dagger, \omega_\mathrm{e}^\dagger\right) = \begin{bmatrix} 0 & 0_{1\times3} & 0_{1\times3} \\ 0_{3\times1} & 0_{3\times3} & -\frac{1}{2}I_{3\times3} \\ 0_{3\times1} & -J^{-1}K_q & -J^{-1}K_\omega \end{bmatrix}.
\tag{14}
$$

This matrix, as explicitly shown in [27], always has at least one eigenvalue with positive real part, from which it follows that $\{q_\mathrm{e}^\dagger, \omega_\mathrm{e}^\dagger\}$ is an unstable fixed point of the state-space system described by (12).

The second equilibrium point of the system specified by (12) is $q_\mathrm{e}^* = [1 \ 0 \ 0 \ 0]^T$ and $\omega_\mathrm{e}^* = [0 \ 0 \ 0]^T$. The asymptotic convergence to this equilibrium point can be readily proven using *Lyapunov's direct method* and *LaSalle's invariance principle*, in the form stated in [37, Corollary 4.2]. Here, we state this result as a proposition.

Proposition 1: Let the controller references in Fig. 3, r_d and ψ_d, be smooth functions of time, and let matrices K_q and K_ω be time invariant and positive definite, as stated in Section III-C. Then, the fixed point $\{q_\mathrm{e}^*, \omega_\mathrm{e}^*\}$ of the state-space system in (12) is asymptotically stable.

Proof: First, we define the function

$$
V(q_\mathrm{e}, \omega_\mathrm{e}) = \frac{1}{2}\omega_\mathrm{e}^T K_q^{-1} J \omega_\mathrm{e} + 2\left(1 - m_\mathrm{e}\right),
\tag{15}
$$

which satisfies the conditions of a Lyapunov function, as stated in [37]. Namely, $V(q_\mathrm{e}, \omega_\mathrm{e})$ is continuously differentiable because both r_d and ψ_d are smooth, $V(q_\mathrm{e}, \omega_\mathrm{e})$ is radially unbounded since its magnitude goes to infinity when at least one of the components of ω_e goes to infinity, and by inspection it can be determined that $V(q_\mathrm{e}, \omega_\mathrm{e})$ is always positive within the limits of q_e, except at $\{q_\mathrm{e}^*, \omega_\mathrm{e}^*\}$, where $V(q_\mathrm{e}^*, \omega_\mathrm{e}^*) = 0$. By differentiating (15) with respect to time, we obtain that

$$
\dot{V}(q_\mathrm{e}, \omega_\mathrm{e}) = \frac{1}{2}\dot{\omega}_\mathrm{e}^T K_q^{-1} J \omega_\mathrm{e} + \frac{1}{2}\omega_\mathrm{e}^T K_q^{-1} J \dot{\omega}_\mathrm{e} - 2\dot{m}_\mathrm{e}. \tag{16}
$$

Because $K_q^{-1}J > 0$, $K_\omega > 0$, and $\dot{m}_\mathrm{e} = -\frac{1}{2}\omega_\mathrm{e}^T n_\mathrm{e}$, substituting $\dot{\omega}_\mathrm{e}$ as specified in (12) into (16) yields

$$
\dot{V}(q_\mathrm{e}, \omega_\mathrm{e}) = -\omega_\mathrm{e}^T K_q^{-1} K_\omega \omega_\mathrm{e} \leq 0, \quad \forall \{q_\mathrm{e}, \omega_\mathrm{e}\}, \tag{17}
$$

where the set of states for which $\dot{V}(q_\mathrm{e}, \omega_\mathrm{e}) = 0$ is $\{q_\mathrm{e}, \omega_\mathrm{e}^*\}$. Then, employing LaSalle's invariance principle, we can immediately conclude that no state of the closed-loop system in (12) can stay in the set $\{q_\mathrm{e}, \omega_\mathrm{e}^*\}$, except at the two equilibrium points. Since the first equilibrium point, $\{q_\mathrm{e}^\dagger, \omega_\mathrm{e}^\dagger\}$, is unstable, the response of the closed-loop state-space system specified by (12), starting from any other conceivable state, converges to the second equilibrium point, $\{q_\mathrm{e}^*, \omega_\mathrm{e}^*\}$. Therefore, this second closed-loop attitude equilibrium point is asymptotically stable. ∎

Last, the positional stability of the closed-loop system, as defined by Fig. 3, can be established by assuming that the
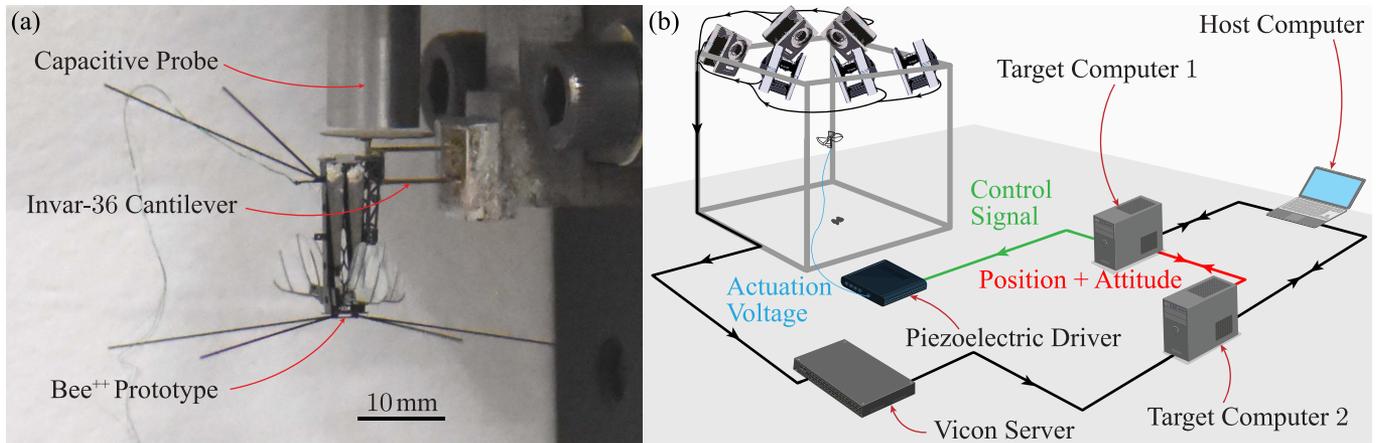
Fig. 4. Experimental setups. (a) Force sensor used to measure thrust in open loop with the tested Bee[++] prototype attached. This sensing device consists of a double-cantilever structure made of Invar-36 (FeNi$_{36}$), which deforms along the negative vertical direction when the microrobot generates aerodynamic forces by flapping. This deformation is measured using a capacitive displacement sensor, as described in [38]. Last, using basic beam theory, we map the instantaneous measured deformation of the double-cantilever structure into a force signal. (b) Instrumented flying arena used in the feedback-controlled flight tests. In this setup, the six degrees of freedom of the controlled Bee[++] prototype are measured using a Vicon motion-capture system. The sensor data (position and attitude) are transmitted to a real-time digital controller at a rate of 500 Hz. The control algorithms are run in real time on a host-target system, using MathWorks SRT software, at a fixed clock-speed of 2 kHz. The control signals are mapped into actuator voltages using specialized piezoelectric drivers, as described in [1] and [27]. Physically, the real-time digital controller was implemented on Target Computer 1, and Target Computer 2 functions as an interface between the Vicon system and Target Computer 1. The controller and data-collection algorithms were programmed using a regular laptop host computer.

response of the attitude dynamics of the system are considerably faster than those of the translational dynamics. Thus, since we already proved the closed-loop stability of the attitude dynamics of the system, we simply replace the actual thrust-force vector in (3), $f\boldsymbol{b}_3$, with the desired thrust-force vector $\boldsymbol{f}_a$ specify by (4) to prove positional stability. Using this approximation, the position and attitude controllers in Fig. 3 are decoupled from each other and the global asymptotic stability of the only closed-loop position equilibrium point of the system, $\{\boldsymbol{r}_e = \boldsymbol{r}_d - \boldsymbol{r} = 0; \dot{\boldsymbol{r}}_e = \dot{\boldsymbol{r}}_d - \dot{\boldsymbol{r}} = 0\}$, can be easily shown using classical linear-analysis techniques [27].

IV. EXPERIMENTAL RESULTS

A. Experimental Setups

To fully test the 6-DOF control capabilities of the Bee[++], we utilized several custom-built experimental apparatuses and environments. First, for characterizing the generation of aerodynamic forces and torques in open loop, we used a modified version of the capacitive force sensor that we presented in [38]. This force sensor is simply a double-cantilever structure made of Invar-36 (FeNi$_{36}$), designed to deform in a predefined manner and thus map periodic forces into linear deflections that can be measured using a capacitive displacement sensor. In this case, we use an 8-mm-diameter cylindrical Physik Instrumente PISeca D510.021 capacitive probe with a 20-μm range and 0.2-nm resolution. The voltage output from this capacitive probe is read using a MathWorks *Simulink Real-Time* (SRT) host-target system and a National Instruments PCI-6229 data acquisition board, at a sampling rate of 10 kHz. A photograph of the force-sensing apparatus, with a Bee[++] prototype attached to it, is shown in Fig. 4(a). During grounded open-loop

force-characterization experiments, as with all other tests presented in this article, we operated the Bee[++] prototype at a sinusoidal flapping frequency of 150 Hz. We chose this value because operating below the natural frequency of the flapping system, which we experimentally estimated to be slightly larger than 165 Hz, significantly prolongs the lifespan of the mechanical components of the robot.

In closed-loop flight tests, we used the flying arena depicted in Fig. 4(b). This experimental setup consists of a Bee[++] prototype, a measurement system, a digital real-time controller, and a signal generator. The measurement system uses a set of six Vicon motion-capture cameras and the specialized Vicon software *Tracker*, which is necessary to accurately collect the instantaneous position and attitude data of the controlled Bee[++] during flight. The data collected by the Vicon-based measurement system are transmitted at a frequency of 500 Hz to the digital real-time controller. Then, the controller processes the position and attitude data in order to calculate the actuator voltages, according to the scheme in Fig. 3, as already discussed in Section III. The control and data-processing algorithms were implemented and are run in real time on Target Computer 1, using MathWorks SRT software, at a fixed clock-speed of 2 kHz. The controller outputs, in the form of actuator signals mapped according to the diagram depicted in Fig. 3, are then analogously transmitted from Target Computer 1 to the signal generator. This generator, using a piezoelectric driver, amplifies the actuator voltages and provides the power required to properly drive the Bee[++] prototype during controlled flight. Last, the driving signals are relayed to the robot via a bundle of five 52-gauge enamel-coated copper wires, thus generating the 150-Hz flapping output that enables controlled flight. Target Computer 2 functions as an interface between the Vicon system and Target Computer 1.

During controlled flight, Bee++ prototypes do not have the capability to collect angular-velocity data using onboard devices, and the Vicon system cannot measure this variable directly. Therefore, the angular-velocity and angular-acceleration signals used in the computation of the control torque specified by (8), $\boldsymbol{\tau}$, must be calculated by numerically differentiating $\boldsymbol{q}$ and $\boldsymbol{q}_{\mathrm{d}}$, a process that significantly exacerbates the negative impacts of sensor noise on the system response and, therefore, greatly limits the flight performance of the controlled Bee++ prototypes. To mitigate these undesirable effects, we implemented the experimental real-time controller with the nonlinear gyroscopic coupling term, $\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}$, off. This modification to the controller is reasonable because, according to numerical simulations implemented using the scheme in Fig. 3, the contribution of the $\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}$ term to the entire control signal is of only about 1%; therefore, its elimination has a negligible effect on the functionality and performance of the system. Further evidence supporting this claim can be found, in the form of raw experimental data, in the supplementary material accompanying this paper.

Additionally, to further improve performance and attenuate the effect of sensor noise, we modified the method used to compute the angular-acceleration reference, $\dot{\boldsymbol{\omega}}_{\mathrm{d}}$. Specifically, when the yaw-angle reference, ψ_{d}, is a smooth function known *a priori*, the signal $\ddot{\psi}_{\mathrm{d}}$ is computed analytically and $\dot{\boldsymbol{\omega}}_{\mathrm{d}}$ is replaced with the approximation

$$\widehat{\dot{\boldsymbol{\omega}}}_{\mathrm{d}} = \boldsymbol{S}^T \begin{bmatrix} 0 \\ 0 \\ \ddot{\psi}_{\mathrm{d}} \end{bmatrix}, \tag{18}$$

where $\boldsymbol{S} = [\boldsymbol{b}_1 \ \boldsymbol{b}_2 \ \boldsymbol{b}_3]$. Note that since the Vicon system provides the attitude of the flyer using Euler angles in the Z-Y-X convention, (18) constrains the selection of the angular-acceleration reference to the set of vectors aligned with the yaw axis, $\boldsymbol{b}_3$, because it has the form $\ddot{\psi}_{\mathrm{d}}\boldsymbol{b}_3$. In cases in which ψ_{d} is required to follow a real-time trajectory in order for the robot to execute a high-level flight task, such as in the experiments discussed in Section IV-E, the controller term $\boldsymbol{J}\dot{\boldsymbol{\omega}}_{\mathrm{d}}$ is turned off. All the modifications applied during the real-time implementation of the flight controllers used in the experiments presented in this article were exhaustively tested in simulation to ensure robust stability under standard experimental conditions. Further evidence supporting this statement, in the form of raw experimental data, can be found in the supplementary material accompanying this paper.

B. Characterization of Aerodynamic Force and Torque

As a first step in evaluating the 6-DOF performance of the Bee++ during flight, we directly measured the aerodynamic thrust generated in open loop by a grounded prototype, using the setup described in Section IV-A. Subsequently, using these thrust-force measurements, we estimated the yaw torque generated in open loop by the tested robot. In this section, we compare these thrust and yaw-torque experimental results to those obtained via the aerodynamic analysis presented in Section II-B. The plot in Fig. 5 shows the measured instantaneous and cycle-averaged thrust force corresponding to an instantaneous sinusoidal flapping signal with a frequency of 150 Hz and
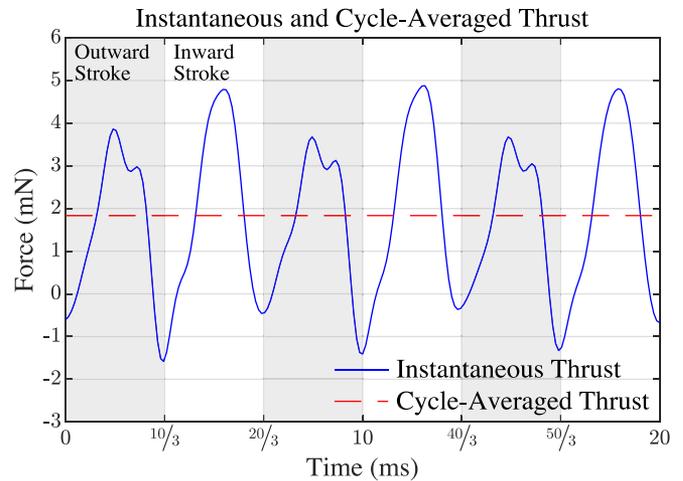

Fig. 5. Instantaneous thrust generated by the tested Bee++ prototype during a grounded experiment performed using the setup shown in Fig. 4(a). In this plot, we show 20 ms of steady-state data obtained through a test during which the wings of the robot were flapped sinusoidally in open loop at 150 Hz for several seconds, with a resulting constant wingstroke angle of about $60°$ (corresponding to an amplitude of about $30°$ relative to the midpoint of flapping). The measured cycle-averaged thrust is approximately 1.8 mN. Using these data and (1), we estimate that at a flapping frequency of 150 Hz and the maximum achievable wingstroke angle of $75°$, the maximum achievable cycle-averaged thrust is in the order of 2.8 mN (0.7 mN for a single wing). Consistently, for a flapping frequency of 150 Hz and wingstroke angle of $45°$ (a typical value of operation during hovering flight), we estimate a cycle-averaged thrust in the order of 1.0 mN (0.25 mN for a single wing). From these force estimates, it immediately follows that a typical value for the cycle-averaged yaw torque, τ_3, available for control during hovering is in the order of 7.3×10^{-7} N·m. Similarly, it follows that the maximum achievable cycle-averaged yaw torque is in the order of 1.13×10^{-6} N·m, which corresponds to a diagonal wing pair flapping with a wingstroke of $75°$ and the other wing pair completely stalled. The 20 ms of data used to make this plot were low-pass filtered using a digital zero-phase Butterworth filter with a -3-dB cutoff frequency of 1040 Hz.

wingstroke angle of about $60°$ (equivalent to an amplitude of about $30°$ relative to the midpoint of flapping). In this experiment, we excited the approximately-linear mechanism that flapped the wings in open loop; therefore, the observed instantaneous wingstroke angle of $60°$ was not precisely prescribed via feedback control and, as a consequence, could have experienced small variations during operation. This flapping angle corresponds to 80% of the maximum wingstroke at which a Bee++ prototype can flap its wings at 150 Hz. As also seen in Fig. 5, during a flapping cycle, the aerodynamic thrust produced by the tested robot oscillates at twice the flapping frequency, in this case 300 Hz. This pattern of thrust-force generation is expected and consistent with prior research [6].

As can be inferred from the results in [10], or deduced by simply inspecting (3), the flight dynamics of the Bee++ prototype act as a low-pass filter with a relatively low 0-dB cutoff frequency (less than 10 Hz for a 95-mg prototype subjected to body drag); therefore, since the fundamental frequency of the generated thrust force significantly exceeds the bandwidth of the Bee++ dynamics, the system primarily responds to the cycle-averaged thrust force. In the experiment corresponding to Fig. 5, the tested prototype generated a cycle-averaged thrust force of about 1.8 mN, which equates to a thrust-to-weight ratio of 1.9. Using these experimental data in combination with the formulas specified by (1) and (2), and considering a wingstroke
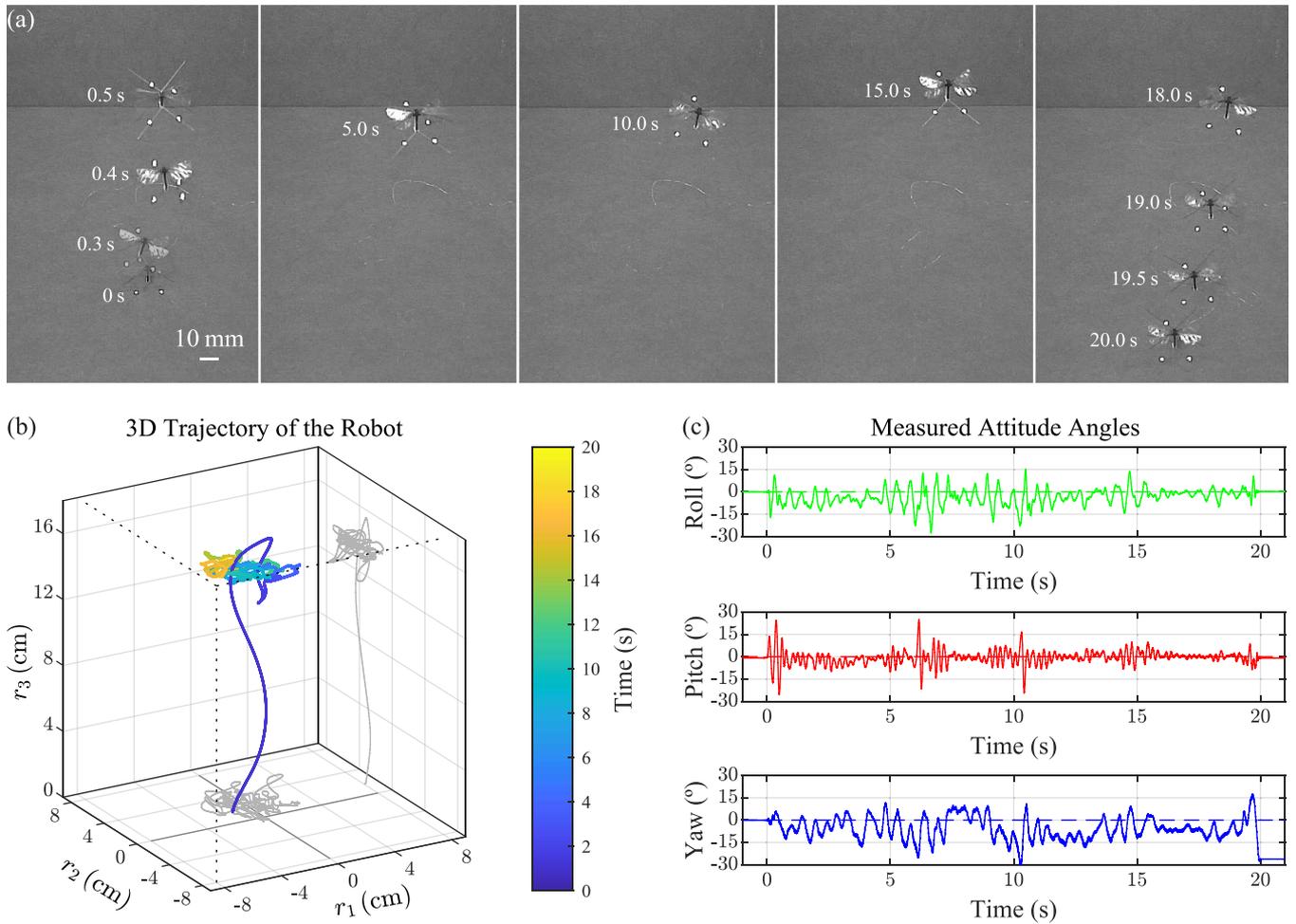
Fig. 6. A feedback-controlled 6-DOF stable hovering flight of the Bee^{++}. (a) Composite of video frames showing a 20-s 6-DOF regulation flight test. As seen, the robot first lifts off and reaches the reference state, then it hovers exhibiting robust stability, and finally executes a landing maneuver. (b) 3D trajectory followed by the controlled Bee^{++} prototype during the 20-s flight test. As seen, the robot maintained a stable altitude with small lateral control errors. (c) Attitude of the controlled Bee^{++} prototype during the 20-s hovering flight. As seen, the values of the roll, pitch, and yaw angles oscillate with small RMS values about the neutral attitude, $\mathbf{0}_{3\times 1}^{\circ}$, plotted using dashed lines. As shown in Fig. 3, the reference for the instantaneous yaw angle, ψ_{d}, was explicitly set to zero while the instantaneous references for the roll and pitch angles were implicitly set in order to satisfy (6) for the 6-DOF regulation case.

angle of $45°$ (a typical flapping angle of operation during hovering flight), we estimate the expected yaw torque available for control to be in the order of $7.3 \times 10^{-7}\,\mathrm{N} \cdot \mathrm{m}$, for a flapping frequency of $150\,\mathrm{Hz}$. Similarly, for this flapping frequency, we estimate a maximum achievable cycle-averaged yaw torque in the order of $1.13 \times 10^{-6}\,\mathrm{N} \cdot \mathrm{m}$, which corresponds to one diagonal wing pair flapping with a wingstroke of $75°$ and the other diagonal wing pair completely stalled. Notably, this value is only about $4.24\,\%$ lower than that we originally predicted through aerodynamic simulations and quasi-steady analyses, employing the methods in [1] and [6], as already discussed in Section II-B.

The slight discrepancy between the simulation-based and experiment-based yaw-torque estimations can be explained as the result of simulation miscalibration and an initial overestimation of the numerical value of the coefficient of force used to specify (1). It is important to mention that, over time, the ability of Bee^{++} prototypes to flap their wings with very large stroke angles diminishes and, therefore, the data in Fig. 5 (corresponding to wingstrokes of $60°$) can be considered to be more

representative of the true thrust-generation capabilities of a controlled Bee^{++} than estimations obtained through aerodynamic simulations and quasi-steady formulas. This observation indicates that there still is room for optimization of both the design and fabrication process of robots of the Bee^{++} class. Despite these shortcomings, also using the grounded-test thrust-force data, we further predict the generation of sustainable yaw torque in the range of $5.8 \times 10^{-7}\,\mathrm{N} \cdot \mathrm{m}$ to $6.8 \times 10^{-7}\,\mathrm{N} \cdot \mathrm{m}$ during flight, while maintaining altitude stability. Through control simulations, such as those presented in [27], we determined that the measured and estimated forces and torques for typical operation here presented are sufficiently large to achieve robust position and attitude stability.

Last in this section, note that another salient characteristic of the measured instantaneous thrust-force signal in Fig. 5 is the shape difference (asymmetry) between the force values corresponding to the outward and inward wingstrokes, defined relative to the resting position of a wing, as depicted in Fig. 2(a). Specifically, during a flapping cycle, the outward stroke produces a smaller local-maximum peak of thrust when compared to

that produced by the inward stroke. Also, during a flapping cycle, the outward stroke produces a deeper negative notch of thrust when compared to that produced by the inward stroke, which barely reaches values below zero. We believe that these two asymmetries are a result of *clap-and-fling* phenomena generated by the aerodynamic interaction of the adjacent wing pairs $\{1, 2\}$ and $\{3, 4\}$. At the end of each inward stroke, the two wings of a pair clap together, thus disturbing their respective local airflows, which has been postulated to alter the generation of flapping forces at the insect scale [39], [40]. While clapping seems to reduce the generation of instantaneous negative thrust, it also induces air vortices that increase drag during the outward stroke of a flapping cycle, thus decreasing the maximum positive thrust generated during this semiperiod. Whether the net impact of clap-and-fling phenomena is positive or negative is difficult to ascertain from the experimental data shown in Fig. 5; however, it is clear that it has a negligible impact on the overall stability and performance of the controller proposed to fly the Bee++.

C. Six-DOF Regulation

To demonstrate that the Bee++, using the ISP method of actuation, is capable of generating sufficiently large forces and torques for controlled flight, and also to verify that the proposed control method can stabilize the position and attitude of the flyer simultaneously, we set the position reference to $r_\mathrm{d} = [0\ 0\ 15]^T$ cm and yaw-angle reference to $\psi_\mathrm{d} = 0°$. We collected data through five different 20-s flight experiments, under identical test conditions. During these five experiments, the tested Bee++ prototype was able to successfully take off, reach the desired position, stabilize all attitude degrees of freedom, and subsequently execute a prescribed landing maneuver. Fig. 6 summarizes the results obtained through one of these experiments. Here, Fig. 6(a) shows an image composite created using frames from video footage of the flight test, Fig. 6(b) shows the 3D trajectory followed by the Bee++ prototype during the 20 s that the flight test lasted, and Fig. 6(c) shows the measured instantaneous roll, pitch, and yaw angles during flight. The raw data and video footage corresponding to this flight experiment and all the other tests discussed in this section are part of the accompanying supplementary material.

As clearly seen in Fig. 6, the proposed feedback controller can effectively regulate all six degrees of freedom of the Bee++ about their respective constant references during hovering flight; we consistently observed a similar performance in dozens of experiments. Averaged across the five different 20-s flight tests, the position of the robot was robustly stabilized with a relatively small RMS value of the control error for each positional DOF. The *mean ± standard deviation* for these five RMS values is 2.9 ± 1.3 cm, 2.3 ± 0.6 cm, and 1.1 ± 0.4 cm in the n_1, n_2, and n_3 directions, respectively. Similarly, in the five different 20-s flight tests mentioned above, the attitude of the flyer was also robustly stabilized with a relatively small RMS value of the control error for each attitude DOF. The *mean ± standard deviation* for these five RMS values is $7.0 \pm 0.6°$, $4.2 \pm 1.2°$, and $9.6 \pm 0.8°$ for the roll, pitch, and yaw degrees of freedom, respectively. To the best of our knowledge, the stability and performance achieved by the controller along the yaw DOF is comparable to those
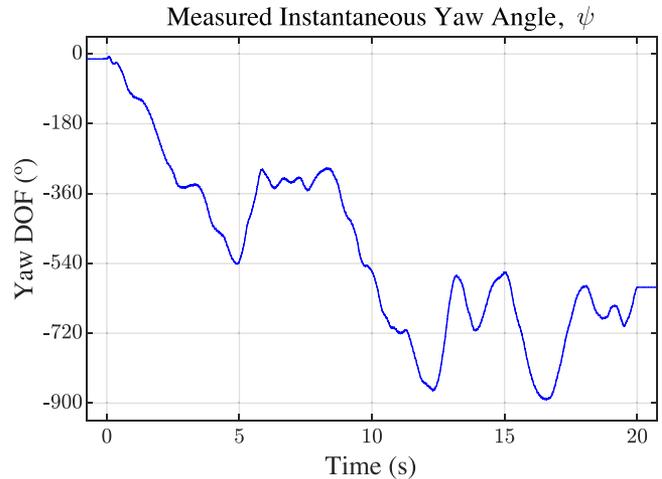

Fig. 7. Measured instantaneous yaw angle, ψ, of the tested Bee++ prototype during hovering flight with yaw control disabled. As seen, ψ drifts seemingly randomly, at one point accumulating almost $-900°$ of uncontrolled rotation.

corresponding to the roll and pitch degrees of freedom for the first time in any sustained flight of an insect-scale flapping-wing flying robot.

To verify that the observed yaw robust stability and measured performance of the tested Bee++ during flight were the result of the feedback-control action and not due to passive damping, we ran an experiment in which all the references were set to the same values as those of the test corresponding to Fig. 6, with the exception of the yaw-angle reference which was set to be equal to the measured instantaneous yaw angle; namely, $\psi_\mathrm{d}(t) = \psi(t)$. Note that by setting $\psi_\mathrm{d}(t) = \psi(t)$, the yaw-control effort is effectively disabled and, therefore, this DOF operates in open loop. The instantaneous yaw angle measured during this flight experiment is shown in Fig. 7. This plot clearly shows that, in the absence of feedback yaw control, the instantaneous yaw angle of the robot drifts significantly during flight and does not reach a stable equilibrium point. This behavior is explained by the aggregated effects of yaw-torque disturbances exerted by the surrounding airflow and electrical tether. The latter disturbance is especially relevant because the inertial and elastic forces applied by the power wires are considerable compared to the scale of the flyer. The marked differences between the feedback-controlled and open-loop yaw-angle measurements provide conclusive evidence that the ISP-based method for yaw-torque generation and control is highly effective for flying insect-scale flapping-wing robots of the Bee++ class.

D. Hovering With Yaw Tracking

Because sustained high-performance yaw control of the Bee++ during flight is the most significant and novel result we present in this article, we conducted a systematic experimental evaluation of the flight controller capabilities while tracking some relevant representative yaw-angle reference signals. To accomplish this objective, we ran experiments in which the position reference was set to $r_\mathrm{d} = [0\ 0\ 15]^T$ cm while the yaw-angle reference was set to a predefined time-varying signal. The reference and measured instantaneous yaw angles, ψ_d and
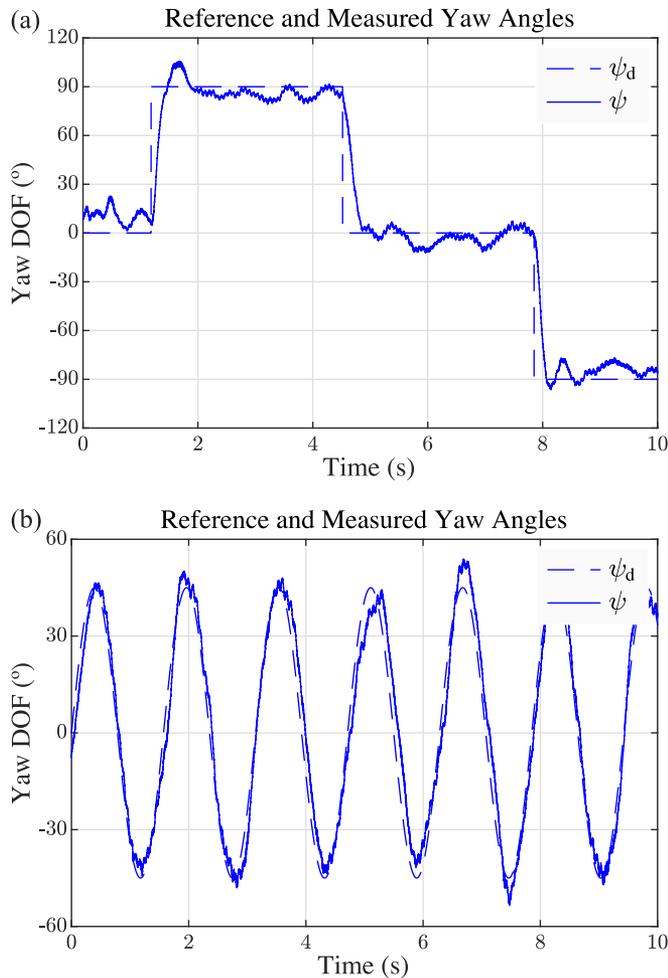
Fig. 8.   Yaw-angle reference, ψ_{d}, and measured instantaneous yaw angle, ψ, of the tested Bee^{++} prototype during hovering flight with active yaw-tracking control. (a) Yaw response to a step reference with variable amplitude. In this flight test, a constant yaw-angle reference was aggressively reset three times with changes of $90°$. (b) Yaw response to a sinusoidal reference with constant amplitude. In this flight test, the reference has an amplitude of $45°$ and a frequency of $4\,\mathrm{rad}\cdot\mathrm{s}^{-1}(0.64\,\mathrm{Hz})$.

ψ, for two of these experiments are shown in Fig. 8. In the first experiment, as seen in Fig. 8(a), ψ_{d} is composed of four consecutive steps with values of $0°$, $90°$, $0°$, and $-90°$. In this case, the measured yaw response, ψ, transitions between the constant set points relatively fast (in less than $300\,\mathrm{ms}$) and with minor overshoots. As expected, but notably, the robot accomplished this control task while maintaining stable hovering flight, which provides conclusive evidence that the tested prototype has the control authority to simultaneously achieve multiple time-varying position and attitude control objectives. As in the yaw-regulation case, in multiple yaw-tracking experiments, the robot exhibited robust stability with an RMS value of the control error in the order of $6.7°$.

It is important to mention that the speeds reached by the controlled robot during the yaw turns shown in Fig. 8(a) are comparable, in order of magnitude, to those reported for the fruit fly (*Drosophila melanogaster*), which can execute a 90-degree yaw turn in about $50\,\mathrm{ms}$ with a rate of change of about $1800°\cdot\mathrm{s}^{-1}$ [4]. Specifically, during a transition between

two set points, the tested Bee^{++} prototype can complete a $90°$ yaw turn in about $200\,\mathrm{ms}$, which corresponds to a yaw-angle rate of change of about $382°\cdot\mathrm{s}^{-1}$. These measurements demonstrate that the flight capabilities of the Bee^{++} represent a significant step forward in the quest to match the maneuverability of natural insects. Further analysis of the data in Fig. 8(a) shows that the tested Bee^{++} prototype can achieve maximum yaw angular accelerations in the order of $143\,\mathrm{rad}\cdot\mathrm{s}^{-2}$. Thus, considering that the moment of inertia of the robot about the $\boldsymbol{b}_3$ axis is estimated to be $4\times10^{-9}\,\mathrm{kg}\cdot\mathrm{m}^2$, we calculate that the maximum yaw torque that the flyer can generate during a step response, while maintaining stable flight, is about $5.7\times10^{-7}\,\mathrm{N}\cdot\mathrm{m}$. This calculated figure represents only $50.44\,\%$ of the maximum achievable yaw torque at a flapping frequency of $150\,\mathrm{Hz}$, estimated using the open-loop grounded-test data shown in Fig. 5, as described in Section IV-B; also, this number is below the predicted yaw-torque range of operation during flight. Considering that, during feedback-controlled flight, the robot must cope with aerodynamic and elastic disturbances, this observed discrepancy in yaw-torque generation is reasonable and to be expected because a sufficient amount of control authority must be allocated to maintain position, roll, and pitch stability.

In the second experiment, as seen in Fig. 8(b), ψ_{d} is a sinusoidal signal with an amplitude of $45°$ and frequency of $4\,\mathrm{rad}\cdot\mathrm{s}^{-1}(0.64\,\mathrm{Hz})$. This flight test was conceived to demonstrate the ability of the Bee^{++} to track a continuously-changing yaw reference. Specifically, we defined the oscillatory function

$$\psi_{\mathrm{d}}(t) = \frac{\pi}{4}\sin(4t), \tag{19}$$

while, as already explained, the position reference remained set at $\boldsymbol{r}_{\mathrm{d}} = [0\ 0\ 15]^T$ cm. As seen in Fig. 8(b), during this flight test, the controlled robot was able to closely track the yaw-angle command while maintaining positional stability. Quantitatively, the RMS value of the measured control error is $7.1°$. In summary, the proposed attitude controller, as specified by (8) and with the additional features introduced in Section IV-A, achieved high tracking performance with a small associated phase lag. The slight discrepancies at the peaks and valleys of the signals result from the trade-off between the control authority allocated for the yaw DOF and altitude, respectively.

Overall, the experimental results presented in this section provide conclusive evidence that the tested Bee^{++} prototype can effectively and accurately track yaw-angle commands defined *a priori* while maintaining high performance and robust stability during hovering flight. In conclusion, when the yaw-angle reference is perfectly known, the proposed control method functions as intended, thus producing the desirable results in terms of stability and performance.

E. Hovering With Yaw Pointing Control

Insect-scale aerial robotic technology has a broad spectrum of potential applications in areas such as terrestrial and extraterrestrial planetary exploration, reconnaissance, search and rescue, environmental monitoring, and artificial pollination. In all these applications, a main common requirement for the

robots involved is the ability to perform positional navigation while maintaining an object in their field of view. To test and demonstrate this ability, a Bee⁺⁺ prototype must be able to point its b_1 axis at a fixed location in space with some level of specified accuracy while subjected to positional variations. With this notion in mind, given the excellent results presented in Sections IV-C and IV-D, the natural progression was to assess the degree to which the Bee⁺⁺ can complete a pointing task during flight. To this end, using yaw control, we commanded the tested prototype to direct its b_1 axis toward a fixed focal point in space, $r_p = [r_{1,p} \ r_{2,p} \ r_{3,p}]^T$, as depicted in Fig. 9(a). Specifically, we defined the instantaneous yaw-angle reference as

$$\psi_d(t) = \tan^{-1}\left(\frac{r_{2,p}(t) - r_2(t)}{r_{1,p}(t) - r_1(t)}\right). \tag{20}$$

Note that when this formula is used to define the yaw-angle reference, the desired yaw trajectory is no longer a function known *a priori*; instead, its instantaneous value, $\psi_d(t)$, is continually computed in real time using as inputs the focal-point reference, $r_p(t)$, and measured time-changing lateral position, $\{r_1(t), r_2(t)\}$, of the robot. An obvious interpretation of this method is that the yaw-angle reference is dynamically adjusted using a feedback loop.

During the yaw-pointing flight test corresponding to the plots in Figs. 9(b) and (c), the position reference of the robot was maintained at $r_d = [0 \ 0 \ 15]^T$ cm while its b_1 axis was directed toward the point $r_p = [10 \ 0 \ 15]^T$ cm. In this case, the dynamically-adjusted reference is turned on at time $t = 2\,\text{s}$. The plots in Fig. 9 clearly show that the controlled Bee⁺⁺ prototype was capable of tracking the yaw-angle reference with a relatively small control error. Quantitatively, the RMS value of the control error is in the order of $11°$, which is equivalent to an RMS value of the focal-point misalignment in the order of $1.9\,\text{cm}$ at the 10-cm focal length. This measured control error indicates that, during the yaw-pointing flight test, the robot was capable of keeping the reference point within the field of view of most optical sensors (about $90°$). Consistently, we can conclude that, in this experimental test, the robot clearly demonstrated the ability to fly stably while maintaining a given point in its field of view. It is important to note that, as intended, the proposed yaw-pointing flight-control method exhibited a quantifiable improvement in performance over standard yaw regulation, in this specific task. For example, if, instead of dynamically adjusting the yaw-angle reference according to (20) to point the b_1 axis of the robot toward the coordinate $r_p = [10 \ 0 \ 15]^T$ cm, we would have chosen the yaw-angle reference $\psi_d = 0°$ and position reference $r_d = [0 \ 0 \ 15]^T$ cm, as in the hovering experiment presented in Section IV-C, the RMS value of the yaw-angle control error would have been in the order of $13°$ instead of $11°$ (approximately 18% larger). The experimental results in Fig. 9 also suggest that the yaw-control performance can be further improved by refining the algorithms used for signal processing and reference generation.

F. Real-Time Tracking of an Infinity Symbol (∞)

Last in this section, we present results obtained through flight tests conceived to determine the impact of complete 3-DOF robustly-stable attitude control on the performance of the Bee⁺⁺
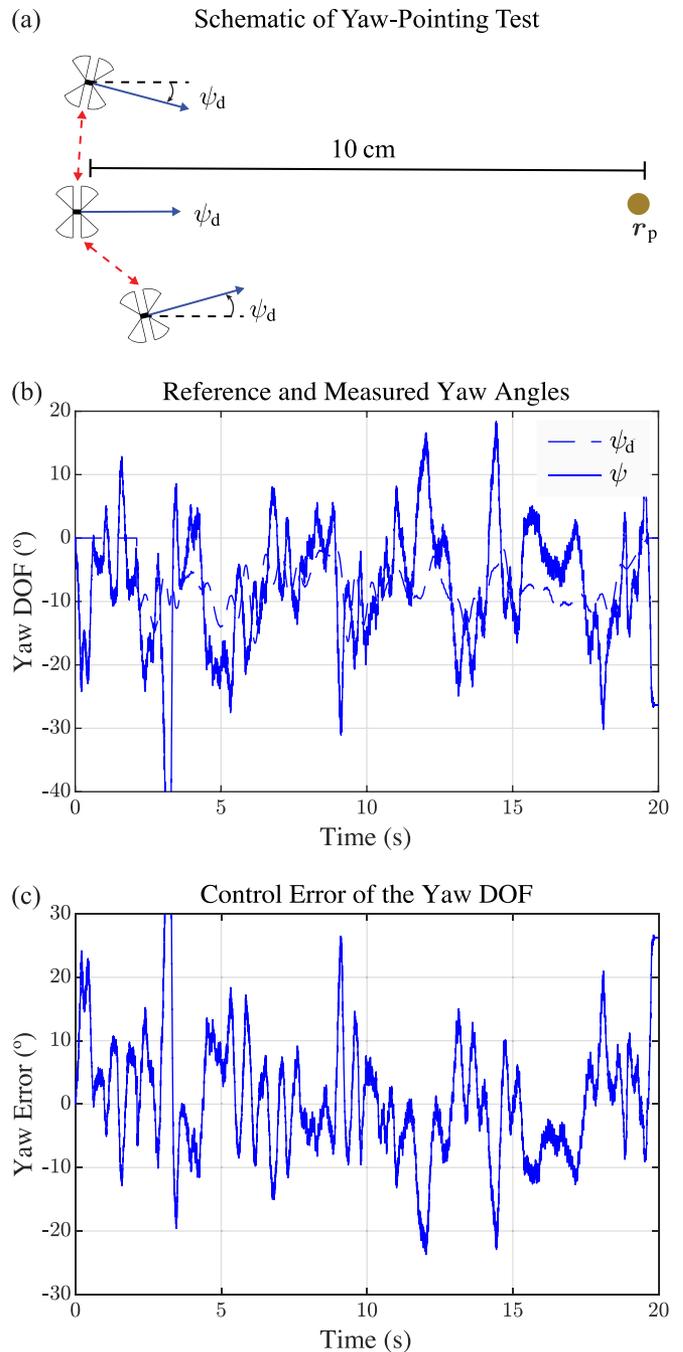

Fig. 9. Yaw pointing of the tested Bee⁺⁺ prototype during 6-DOF controlled flight. (a) Schematic illustration that explains the execution of yaw-pointing maneuvers during a flight test. (b) Comparison of the yaw-angle reference, ψ_d, with the measured instantaneous yaw angle, ψ, of the controlled Bee⁺⁺ prototype while pointing at a fixed location in space, $r_p = [10 \ 0 \ 15]^T$ cm. (c) Control error of the yaw DOF, $\psi_d - \psi$, corresponding to the same yaw-pointing experiment through which we obtained the data in (b).

position controller during position-trajectory tracking. Specifically, we commanded the controlled Bee⁺⁺ prototype with exactly the same position reference, under identical experimental conditions, during two different flight tests. In the first test, we flew the robot with the yaw controller off and, in the second test, with the yaw controller on. In these two experiments, the flyer was commanded to track an *infinity symbol* (∞) in space,
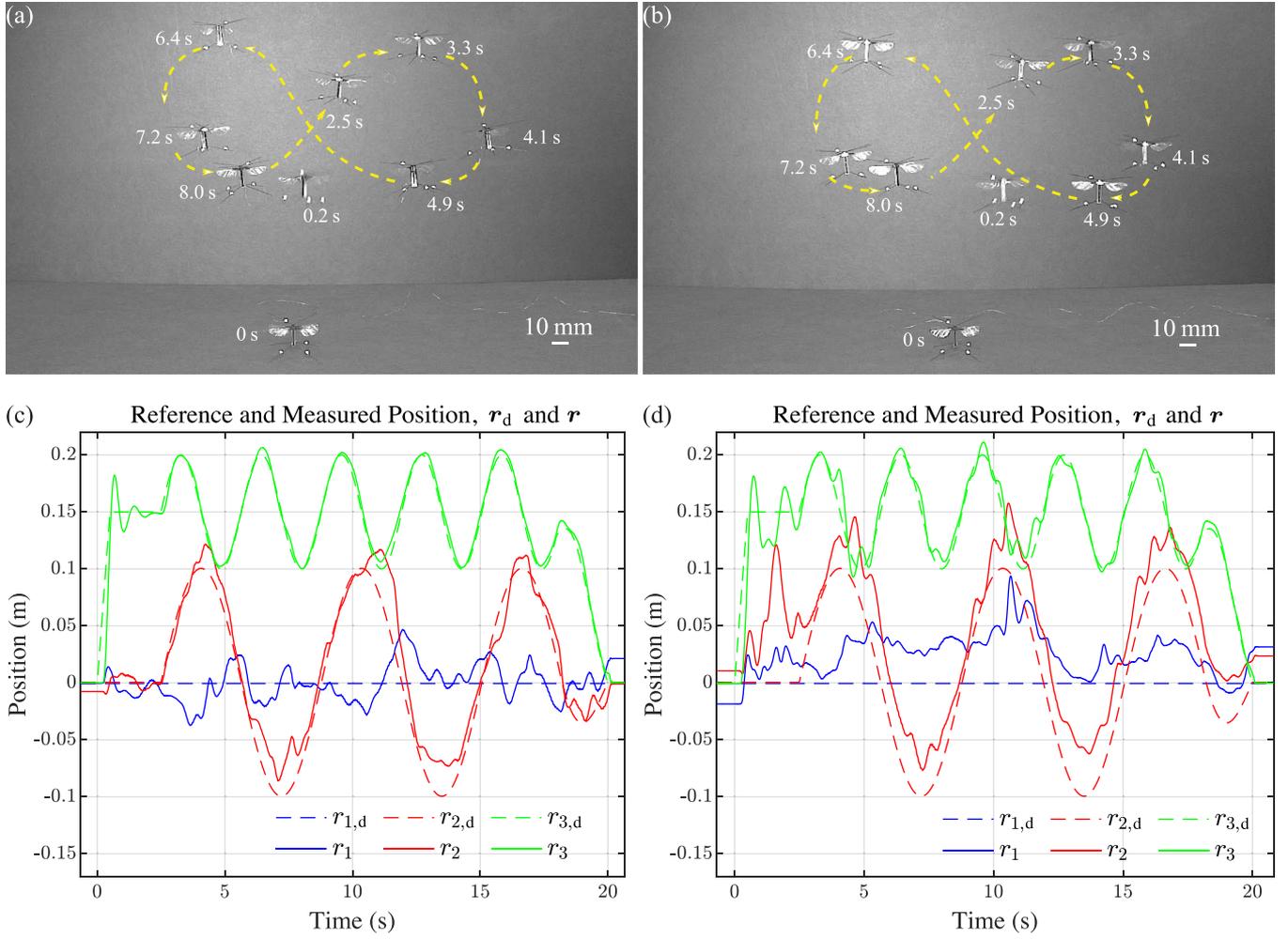
Fig. 10. Real-time tracking of an infinity symbol (∞) by the controlled Bee^{++} prototype. (a) Composite of sequential video-frames showing a flight experiment during which the robot tracks the trajectory defined by an infinity symbol with yaw control disabled. (b) Composite of sequential video-frames showing a flight experiment during which the robot tracks the trajectory defined by an infinity symbol with yaw control enabled. (c) Position reference, $\boldsymbol{r}_\mathrm{d} = [r_{1,\mathrm{d}} \; r_{2,\mathrm{d}} \; r_{3,\mathrm{d}}]^T$, and measured instantaneous position, $\boldsymbol{r} = [r_1 \; r_2 \; r_3]^T$, of the robot's COM during flight with yaw control disabled. (d) Position reference, $\boldsymbol{r}_\mathrm{d} = [r_{1,\mathrm{d}} \; r_{2,\mathrm{d}} \; r_{3,\mathrm{d}}]^T$, and measured instantaneous position, $\boldsymbol{r} = [r_1 \; r_2 \; r_3]^T$, of the robot's COM during flight with yaw control enabled.

mathematically defined as

$$\boldsymbol{r}_\mathrm{d}(t) = \begin{bmatrix} 0 \\ 0.1 \sin t \\ 0.05 \sin(2t) + 0.15 \end{bmatrix}, \qquad (21)$$

which is a vector with a constant zero value along the $\boldsymbol{n}_1$ direction and that oscillates twice as fast along the $\boldsymbol{n}_3$ direction as along the $\boldsymbol{n}_2$ direction, thus tracing the desired infinity-symbol trajectory in the $\boldsymbol{n}_2$-$\boldsymbol{n}_3$ plane. A photographic composite showing the trajectory followed by the robot with the yaw controller disabled is shown in Fig. 10(a); a photographic composite showing the trajectory followed by the robot with the yaw controller enabled is shown in Fig. 10(b). The corresponding position data are shown in Figs. 10(c) and (d), respectively.

The data in Fig. 10 clearly show that the controlled robot adequately followed the position-trajectory reference during both flight tests, which indicates that 3-DOF attitude control is not necessary to achieve high-performance position-trajectory tracking. In fact, the comparison in Table II shows that the tracking performance achieved by the robot during flight slightly

TABLE II
RMS VALUES OF THE TRACKING CONTROL ERRORS FOR THE BEE^{++}
PROTOTYPE FLYING WITH AND WITHOUT ACTIVE YAW CONTROL

Inertial Axis	Without Yaw Control (cm)	With Yaw Control (cm)
$\boldsymbol{n}_1$	1.7	3.3
$\boldsymbol{n}_2$	1.9	3.5
$\boldsymbol{n}_3$	0.5	0.8

decreases when the yaw DOF is actively controlled. These empirical results are reasonable and expected when considering that actively controlling the yaw DOF decreases the control authority available for the other controlled states of the system, a direct consequence of the restricted ability to produce thrust and body torques. As shown in Fig. 3, the yaw-angle reference is one out of four control commands. Consistently, by disabling the yaw-control effort, the real-time controller can allocate more actuation resources to achieving robust positional stability and high performance during flight. In contrast, with yaw control enabled, the controller must allocate part of the control and

actuation efforts toward achieving yaw stability, thus decreasing the force and torque available for positional tracking. We see this reallocation of resources as a necessary tradeoff because vision-enabled controlled yaw maneuverability is a critical function of robotic insects envisioned capable of executing tasks useful for humans. Fortunately, this limitation can be lessened and potentially eliminated in future design iterations by further increasing the thrust force and body torques produced by the robot in order to prevent actuator saturation.

V. CONCLUSION

We proposed a new ISP-based method for yaw actuation and control for insect-scale flapping-wing flyers. Also, we presented a comprehensive study of the method and discussed its design and real-time implementation on a new four-wing four-actuator platform, the Bee⁺⁺. After describing the robotic-design elements that enable 6-DOF robustly-stable high-performance flight control, we introduced a procedure for synthesizing and implementing the nonlinear control scheme devised to achieve these flight objectives. Subsequently, we presented experimental results obtained through a series of flight tests that demonstrate and quantify the stability and performance that the Bee⁺⁺ platform can achieve during flight, with a particular focus on the dynamic response of the yaw DOF. We started the flight tests with an example of basic 6-DOF regulation and finished with an example of complex-trajectory track following in space. In these experiments, we obtained the best results reported to date for an insect-scale flapping-wing flying robot during 6-DOF feedback-controlled sustained flight in terms of stability and performance. These results not only show unparalleled attitude stability for an extended duration (20 s); they also demonstrate the unique position and attitude-tracking capabilities of the Bee⁺⁺ design, which still is the only insect-scale four-wing flyer driven by four independent actuators. Furthermore, these results allow us to anticipate future advances in control and robotic design.

The flight performance achieved by the Bee⁺⁺ with the proposed ISP-based actuation method and 6-DOF flight control scheme represents a fundamental breakthrough in the field of insect-scale flapping-wing aerial robotics. While some previous two-actuator and four-actuator insect-scale flying robots have demonstrated closed-loop yaw control during brief flights, no other subgram flyer has achieved a similar level of feedback-controlled yaw stability during sustained flight. This accomplishment is relevant in the quest to achieve full autonomy at the subgram scale because 6-DOF (position + attitude) stabilization is a crucial capability required to execute complex tasks such as object tracking, obstacle avoidance, and multirobot formation flight. These performance objectives, which were previously unattainable with microrobotic fliers, are now worthy of future investigation. Furthermore, the results presented in this article represent a significant step toward the integration of sensors with limited fields of view into insect-scale flyers with the purpose of implementing vision-based feedback control. Considering that we were able to conclusively demonstrate the efficacy of the proposed ISP-based actuation method for flight control, which is enabled by the novel four-actuator design of

the Bee⁺⁺ platform, we believe that a rigorous quantification of the body torques generated by flyers of this type would significantly advance the field of subgram aerial robotics. Similarly, we believe that there is still room for improvement regarding thrust generation by developing new actuators, transmission mechanisms, and wings. Improvement in thrust production is essential to reduce, or eliminate, actuator saturation and optimize trajectory tracking with simultaneous yaw stabilization. These open-research issues are a matter of current and further research at the *Autonomous Microrobotic Systems Laboratory* (AMSL).

REFERENCES

[1] X. Yang, Y. Chen, L. Chang, A. A. Calderón, and N. O. Pérez-Arancibia, "Bee⁺: A 95-mg four-winged insect-scale flying robot driven by twinned unimorph actuators," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4270–4277, Oct. 2019.

[2] M. H. Dickinson and F. T. Muijres, "The aerodynamics and control of free flight manoeuvres in *Drosophila*," *Philos. Trans. R. Soc. B: Biol. Sci.*, vol. 371, no. 1704, Sep. 2016, Art. no. 20150388.

[3] J.-C. Zufferey, *Bio-Inspired Flying Robots: Experimental Synthesis of Autonomous Indoor Flyers*. Lausanne, Switzerland: EPFL Press, 2008.

[4] S. N. Fry, R. Sayaman, and M. H. Dickinson, "The aerodynamics of free-flight maneuvers in *Drosophila*," *Science*, vol. 300, no. 5618, pp. 495–498, Apr. 2003.

[5] S. M. Walker et al., "*In vivo* time-resolved microtomography reveals the mechanics of the blowfly flight motor," *PLoS Biol.*, vol. 12, no. 3, Mar. 2014, Art. no. e1001823.

[6] L. Chang and N. O. Pérez-Arancibia, "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, Jan. 2016, Art. no. AIAA2016-0013.

[7] L. Chang and N. O. Pérez-Arancibia, "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, Jun. 2018, Art. no. AIAA2018-2830.

[8] I. Cohen, S. C. Whitehead, and T. Beatus, "Fluid dynamics and control of insect flight," *Nat. Rev. Phys.*, vol. 1, no. 11, pp. 638–639, Nov. 2019.

[9] R. J. Wood, "Design, fabrication, and analysis of a 3DOF, 3 cm flapping-wing MAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, San Diego, CA, USA, Oct.–Nov. 2007, pp. 1576–1581.

[10] N. O. Pérez-Arancibia, K. Y. Ma, K. C. Galloway, J. D. Greenberg, and R. J. Wood, "First controlled vertical flight of a biologically inspired microrobot," *Bioinspir. Biomim.*, vol. 6, no. 3, Sep. 2011, Art. no. 036009.

[11] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, "Controlled flight of a biologically inspired, insect-scale robot," *Science*, vol. 340, no. 6132, pp. 603–607, May 2013.

[12] E. F. Helbling, S. B. Fuller, and R. J. Wood, "Pitch and yaw control of a robotic insect using an onboard magnetometer," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Hong Kong, China, May–Jun. 2014, pp. 5516–5522.

[13] P. Chirarattananon, K. Y. Ma, and R. J. Wood, "Adaptive control of a millimeter-scale flapping-wing robot," *Bioinspir. Biomim.*, vol. 9, no. 2, Jun. 2014, Art. no. 025004.

[14] Z. E. Teoh, "Design of hybrid passive and active mechanisms for control of insect-scale flapping-wing robots," Ph.D. dissertation, Harvard University, Cambridge, MA, USA, Aug. 2015.

[15] S. B. Fuller, "Four wings: An insect-sized aerial robot with steering ability and payload capacity for autonomy," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 570–577, Apr. 2019.

[16] R. Steinmeyer, N.-s. P. Hyun, E. F. Helbling, and R. J. Wood, "Yaw torque authority for a flapping-wing micro-aerial vehicle," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Montreal, Canada, May 2019, pp. 2481–2487.

[17] Y. Chen et al., "Controlled flight of a microrobot powered by soft artificial muscles," *Nature*, vol. 575, no. 7782, pp. 324–329, Nov. 2019.

[18] N. T. Jafferis, E. F. Helbling, M. Karpelson, and R. J. Wood, "Untethered flight of an insect-sized flapping-wing microscale aerial vehicle," *Nature*, vol. 570, no. 7762, pp. 491–495, Jun. 2019.

[19] Y. M. Chukewad and S. Fuller, "Yaw control of a hovering flapping-wing aerial vehicle with a passive wing hinge," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1864–1871, Apr. 2021.

[20] D. B. Doman, M. W. Oppenheimer, and D. O. Sigthorsson, "Dynamics and control of a minimally actuated biomimetic vehicle: Part I. Aerodynamic model," in *Proc. AIAA Guid. Navig. Control Conf.*, Chicago, IL, USA, Aug. 2009, Art. no. AIAA2009-6160.

[21] M. W. Oppenheimer, D. B. Doman, and D. O. Sigthorsson, "Dynamics and control of a minimally actuated biomimetic vehicle: Part II. Control," in *Proc. AIAA Guid. Navig. Control Conf.*, Chicago, IL, USA, Aug. 2009, Art. no. AIAA2009-6161.

[22] S. B. Fuller, J. P. Whitney, and R. J. Wood, "Rotating the heading angle of underactuated flapping-wing flyers by wriggle-steering," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 1292–1299.

[23] N. Gravish and R. J. Wood, "Anomalous yaw torque generation from passively pitching wings," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Stockholm, Sweden, May 2016, pp. 3282–3287.

[24] P. Chirarattananon, K. Y. Ma, and R. J. Wood, "Perching with a robotic insect using adaptive tracking control and iterative learning control," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1185–1206, Sep. 2016.

[25] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov, "Development of the nano hummingbird: A tailless flapping wing micro air vehicle," in *Proc. 50th AIAA Aerosp. Sci. Meeting*, Nashville, TN, USA, Jan. 2012, Art. no. AIAA2012-0588.

[26] Z. Tu, F. Fei, J. Zhang, and X. Deng, "An at-scale tailless flapping-wing hummingbird robot. I. Design, optimization, and experimental validation," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1511–1525, Oct. 2020.

[27] R. M. Bena, X.-T. Nguyen, X. Yang, A. A. Calderón, Y. Chen, and N. O. Pérez-Arancibia, "A multiplatform position control scheme for flying robotic insects," *J. Intell. Robot. Syst.*, vol. 105, no. 1, May 2022, Art. no. 19.

[28] C. De Wagter, M. Karásek, and G. de Croon, "Quad-thopter: Tailless flapping wing robot with four pairs of wings," *Int. J. Micro Air Veh.*, vol. 10, no. 3, pp. 244–253, Sep. 2018.

[29] N. T. Jafferis, M. A. Graule, and R. J. Wood, "Non-linear resonance modeling and system design improvements for underactuated flapping-wing vehicles," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Stockholm, Sweden, May 2016, pp. 3234–3241.

[30] R. Malka, A. L. Desbiens, Y. Chen, and R. J. Wood, "Principles of microscale flexure hinge design for enhanced endurance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Chicago, IL, USA, Sep. 2014, pp. 2879–2885.

[31] A. Franchi, "Interaction control of platforms with multi-directional total thrust," in *Aerial Robotic Manipulation*, A. Ollero and B. Siciliano, Eds. Cham, Switzerland: Springer, 2018, pp. 175–189.

[32] Y. Chen and N. O. Pérez-Arancibia, "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, Oct. 2018, pp. 7845–7852.

[33] Y. Chen and N. O. Pérez-Arancibia, "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, May 2017, pp. 3599–3606.

[34] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY, USA: Springer, 2014.

[35] B. Wie, H. Weiss, and A. Arapostathis, "Quaternion feedback regulator for spacecraft eigenaxis rotations," *J. Guid. Control Dyn.*, vol. 12, no. 3, pp. 375–380, May–Jun. 1989.

[36] J. T.-Y. Wen and K. Kreutz-Delgado, "The attitude control problem," *IEEE Trans. Autom. Control*, vol. 36, no. 10, pp. 1148–1162, Oct. 1991.

[37] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[38] E. K. Singer, L. Chang, A. A. Calderón, and N. O. Pérez-Arancibia, "Clip-brazing for the design and fabrication of micronewton-resolution, millimeter-scale force sensors," *Smart Mater. Struct.*, vol. 28, no. 5, May 2019, Art. no. 055028.

[39] X. Cheng and M. Sun, "Wing-kinematics measurement and aerodynamics in a small insect in hovering flight," *Sci. Rep.*, vol. 6, May 2016, Art. no. 25706.

[40] A. Santhanakrishnan et al., "Clap and fling mechanism with interacting porous wings in tiny insect flight," *J. Exp. Biol.*, vol. 217, no. 21, pp. 3898–3909, Nov. 2014.

Ryan M. Bena received the B.S. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2012 and the 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. He is currently pursuing a Ph.D. degree in aerospace engineering at USC, working in the Autonomous Microrobotic Systems Laboratory and the Dynamic Robotics and Control Laboratory. His research interests include nonlinear controller design and analysis, embedded automation of micro aerial vehicles, and optimization-based control techniques.

Xiufeng Yang received the 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 the M.S. and Ph.D. degrees in mechanical engineering from the University of Southern California (USC), Los Angeles, CA, USA, in 2015 and 2020, respectively. From August 2014 to May 2020, he was a Graduate Researcher with the Autonomous Microrobotic Systems Laboratory at USC. He is currently a Research and Development Engineer with Applied Materials, Santa Clara, CA, USA. His research interests include microrobotics, actuators and sensors, microfabrication, real-time feedback control, and mechatronics.

Ariel A. Calderón received the B.S. and Engineer degrees in mechanical engineering from Universidad de Chile, Santiago, Chile, in 2010 and 2012, respectively, and the 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 Autonomous Microrobotic Systems Laboratory at USC. From October 2020 to June 2021, he was a Postdoctoral Scholar with the Department of Civil and Environmental Engineering at the University of California, Los Angeles (UCLA), Los Angeles, CA, USA. He is currently a Senior Mechanical Design Engineer with Intuitive Surgical Inc., Sunnyvale, CA, USA. His research interests include microrobotics, actuators and sensors, microfabrication, real-time feedback control, soft robotics, and medical devices.

Néstor O. Pérez-Arancibia received the 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. Also, he is currently an Associate Editor for Mechatronics, a journal of the International Federation of Automatic Control (IFAC). Many of Dr. Pérez-Arancibia's research articles have been nominated for or won best paper awards. Most recently, he was the recipient of a 2020 IEEE Robotics and Automation Letters Best Paper Award. His research and teaching interests include feedback control, mechatronics, microrobotics, soft robotics, and biologically-inspired engineering.