

Adaptive control of a millimeter-scale flapping-wing robot

Pakpong Chirarattananon, Kevin Y Ma and Robert J Wood

Harvard University School of Engineering and Applied Sciences, Cambridge, MA 02138, USA

E-mail: chirarat@fas.harvard.edu


Received 11 September 2013, revised 24 March 2014

Accepted for publication 8 April 2014

Published 22 May 2014

Abstract

Challenges for the controlled flight of a robotic insect are due to the inherent instability of the system, complex fluid–structure interactions, and the general lack of a complete system model. In this paper, we propose theoretical models of the system based on the limited information available from previous work and a comprehensive flight controller. The modular flight controller is derived from Lyapunov function candidates with proven stability over a large region of attraction. Moreover, it comprises adaptive components that are capable of coping with uncertainties in the system that arise from manufacturing imperfections. We have demonstrated that the proposed methods enable the robot to achieve sustained hovering flights with relatively small errors compared to a non-adaptive approach. Simple lateral maneuvers and vertical takeoff and landing flights are also shown to illustrate the fidelity of the flight controller. The analysis suggests that the adaptive scheme is crucial in order to achieve millimeter-scale precision in flight control as observed in natural insect flight.

 Online supplementary data available from stacks.iop.org/BB/9/025004/mmedia

Keywords: biologically-inspired robots, flapping wing, adaptive control, unmanned aerial vehicles

(Some figures may appear in colour only in the online journal)

1. Introduction

Inspired by the agility of flying insects and motivated by myriad engineering challenges and open scientific questions, the RoboBees project is developing a colony of autonomous robotic insects. The effort to create such millimeter-scale robots is motivated by numerous possible applications such as assisted agriculture, reconnaissance, search and rescue in hazardous environments, and the potential to aid entomologists in the study of insect flight. In [1], the controlled flight of a millimeter-scale flapping-wing robot was first demonstrated empirically. This result was the culmination of research in meso-scale actuation [2] and advances in manufacturing [3]. These developments enabled the creation of insect-scale flapping-wing vehicles that are able to generate torques about all three body axes [4, 5], a requirement for flapping-wing *micro aerial vehicles* (MAVs) due to their inherent instability [6, 7].

From a control perspective, flapping-wing MAVs [8, 9] bear some resemblance to helicopters and quadrotors. Both types of aircraft are typically underactuated with four inputs and six outputs. While flying insects are able to perform extraordinary aerodynamic feats, small quadrotors are also known to possess great maneuverability [10, 11]. Given the extensive research on controlling quadrotors using various techniques [10–12], similar strategies may also be suitably applied to the emerging flapping-wing MAV prototypes.

In addition to swift dynamics, a primary challenge in controlling the robotic insect shown in figure 1(a) is due to the lack of comprehensive knowledge of the system and the variation in system properties as a result of complex aerodynamics and manufacturing imperfections. Empirical characterization and system identification are not currently feasible since a multi-axis force/torque sensor with appropriate range and resolution for the robots of interest does not exist. To compensate, in previous work [1], predictions of the system's characteristics were made based on theoretical

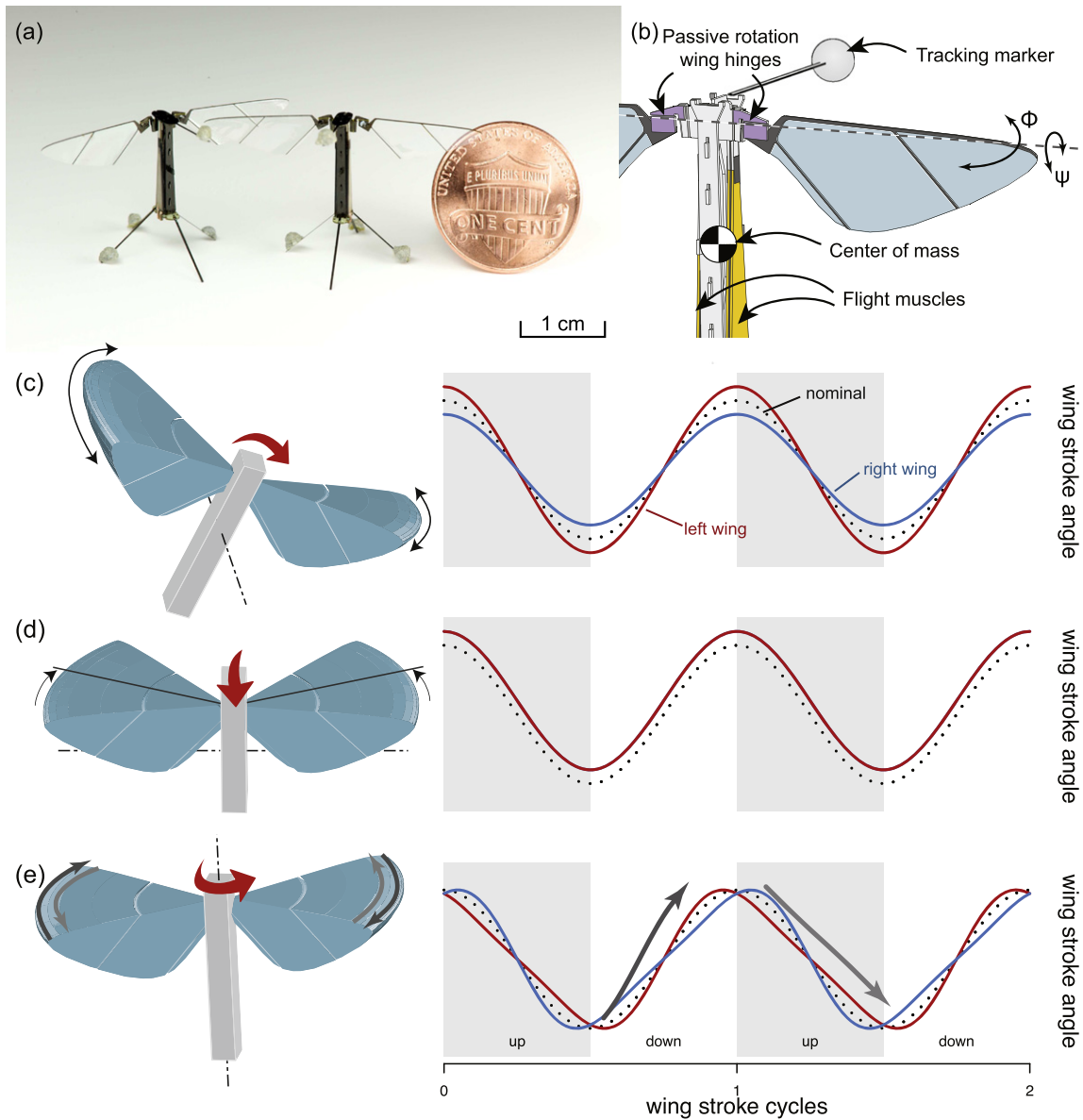


Figure 1. (a) Photograph of two biologically-inspired flapping-wing robots next to a US penny. (b) A schematic model highlights key components of the insect-scale robot and the rotational axes of the flapping strokes and the passive rotation axes. (c) Roll torque generation. (d) Pitch torque generation. (e) Yaw torque generation.

models [13, 14]. In order to achieve sustained flight, it is necessary to account for uncertain parameters arising from manufacturing errors (e.g. torque offsets); this cannot be done by modeling alone. One possible approach to account for model uncertainties is to use an adaptive controller.

The controllers used in [1] were not inherently adaptive. Instead, an integral part was added to deal with parameter uncertainty. It is conceivable that the use of adaptive controllers with proven convergence properties could improve flight performance. Additionally, the results allow us to gain further insights into the flight dynamics of the vehicle and obtain more realistic models for control purposes. In this paper we revisit the problem of controlling the robotic insect by employing an adaptive approach. The flight controller, initially presented in [15], has been designed based on proposed Lyapunov functions using sliding mode control

techniques. The control laws and adaptive laws are derived such that the stability can be guaranteed in a Lyapunov sense. The use of adaptive sliding mode control is not novel in MAV applications, nevertheless, earlier work often relies on a small-angle assumption [16, 17]. In this work, the proposed controller possesses a large domain of attraction. Another major benefit is the reliability of the adaptive parts that allow us to efficiently obtain estimates of uncertain parameters. The performance of the proposed controller was preliminarily tested in a few hovering flights and vertical takeoff and landing flights in [15]. In this paper, we have expanded the work to include more flight experiments by performing more than 20 hovering flights using adaptive and non-adaptive controllers. Demonstration of simple maneuvers is also presented, together with a more thorough analysis of the results. The outcomes suggest the importance of the adaptive scheme

and its potential roles in the flight control system of real insects.

The rest of the paper is organized as follows. For completeness and clarity, the description of the microrobot used in the experiments and its relevant flight dynamics, including details on thrust and torque generation first reported in [1] are reproduced in section 2. The derivations of the controllers are given in section 3. Section 4 contains the implementation and flight experiments. Finally, a conclusion and further considerations on the connections to insect flight control are discussed in section 5.

2. Robot description and flight dynamics

2.1. Robot design

The robot in this study (illustrated in figure 1(a)) is an 80 mg flapping-wing microrobot fabricated using the *smart composite microstructures* process as detailed in [3, 4]. The robot's airframe is made of layers of carbon fiber, laminated under heat and pressure to form a rigid and lightweight composite, and laser machined. Resilient flexure joints are fabricated from polyimide film. The assembly process takes inspiration from origami to create complex 3D structures and mechanisms by folding. Taking inspiration from *Diptera* (flies), the robot has two wings and is equipped with two piezoelectric bimorph actuators. The addition of the second actuator distinguishes this robot from the earlier prototype in [18], allowing two wings to be driven independently, resembling direct flight muscles found at the base of the wings in *Odonata* (dragonflies) [19]. This design also deviates from the robot presented in [5, 20], where two small control actuators were mounted at the base of each wing to independently fine tune the stroke amplitudes of the wings, similar to the thoracic topology of a Dipteran insect [19]. The physical parameters of the robot reported in [1] are reproduced in table 1.

In this robot, linear displacement of the actuator tip is amplified and converted into a rotational motion of the wing (described the angle Φ in figure 1(b)) by a flexure-based four-bar transmission, creating an actuator-transmission-wing system. The angle of attack of both wings is not directly controlled and relegated to a passive mechanism by the incorporation of compliant flexures at the wing hinges as shown in figure 1(b). Lift force is then produced as a result of the wing rotation along the angle Ψ . In operation, the flapping frequency is typically fixed at a value between 110 – 120 Hz, near the system's resonant frequency. The robotic insect is capable of modulating the thrust force that is nominally aligned with the robot's vertical axis by altering its flapping amplitude and able to generate torques along its three body axes using different flapping schemes as detailed in figures 1(c)–(e). Theoretically, this allows the robot to be controllable over the $SO(3)$ space. Similar to most insects, which lack the ability to generate lateral thrust without banking [21], lateral maneuvers can be achieved by re-

Table 1. Various physical parameters of the robotic insect.

Robot properties		
Total mass	80	mg
Flight muscle mass	50	mg
Tracking marker mass	5	mg
Wire tether mass	0.5	mg cm ⁻¹
Roll axis inertia	1.42	mg mm ²
Pitch axis inertia	1.34	mg mm ²
Yaw axis inertia	0.45	mg mm ²
Reynolds number	<1200	
Flapping frequency	120	Hz
Flapping amplitude	110	degrees
Power consumption during hover	19	mW
Robot geometry		
Height	14	mm
Body width	3.5	mm
Wing span	35	mm
Wing properties		
Wing length	15	mm
Mean chord length	3.46	mm
Area	52	mm ²
Inertia (flapping axis)	45.3	mg mm ²
Mass	1	mg

orienting the body such that the net thrust vector takes on a lateral component as modeled in the literature [22, 23].

Without power source, microcontroller, and sensors, the robot currently draws power from an offboard power source via a wire tether. Sensing is provided by a camera-based motion capture system and computation for control is performed off-board as illustrated in figure 2(a).

2.2. Flight dynamics

Owing to the relatively small inertia of the wings (relative to the body) and rapid but low-amplitude motion of the actuators, for the time scales of interest, these small oscillations are neglected and only stroke-averaged dynamics are considered. Stroke-averaged models were found to be sufficiently accurate to capture the dynamics of insects and robots of similar scales [18, 24]. Herein, the robotic insect is regarded as a rigid body in three-dimensional space, based on the standard aircraft model—the approach often taken in modeling the flight dynamics of flapping-wing MAVs [6, 7]. In the body attached coordinates, the roll, pitch, and yaw axes are aligned with the \hat{x} , \hat{y} , and \hat{z} axes as presented in figure 2(b).

Due to symmetry, it is reasonable to assume that the cross terms in the moment of inertia matrix \mathbf{J} are negligible. The orientation between the body frame and the inertial frame is defined by the rotation matrix R , which is rotating at an angular velocity ω with respect to the body frame. As a result,

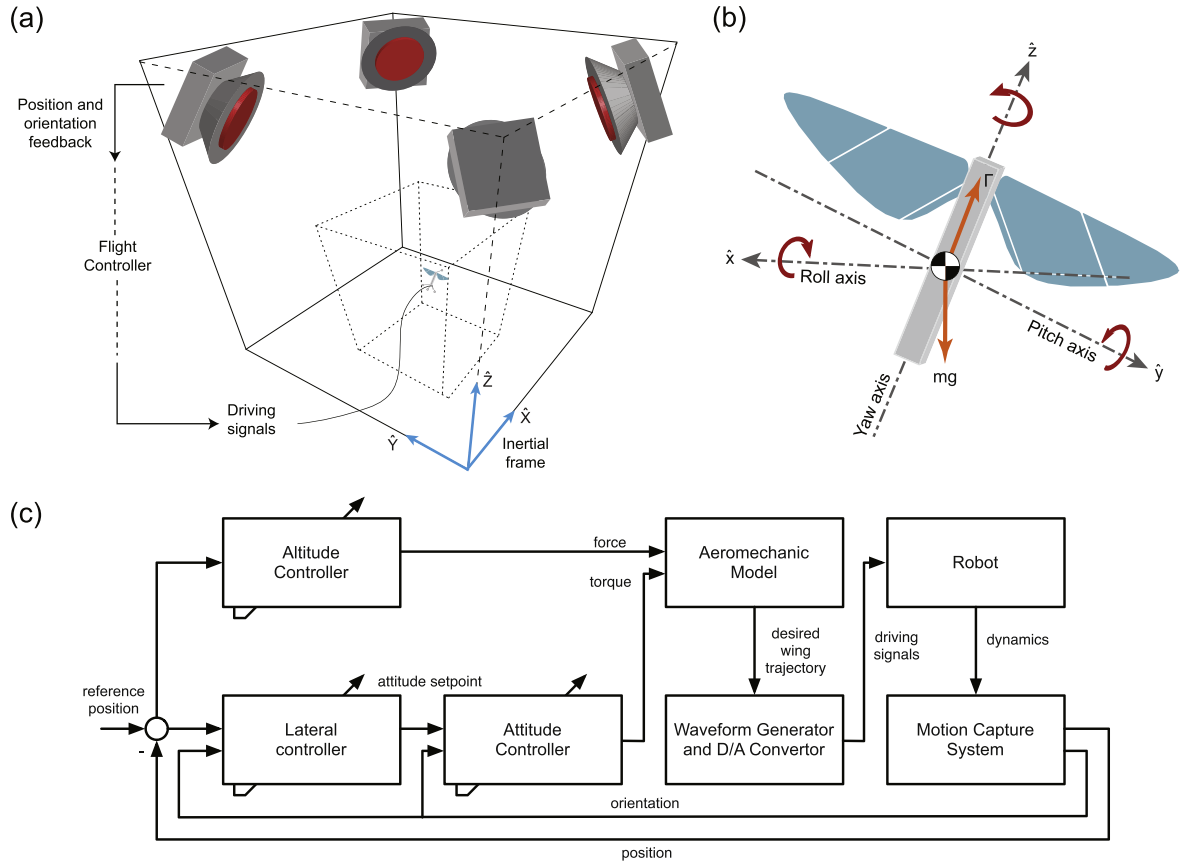


Figure 2. (a) Schematic drawing of the flight arena. (b) Definitions of the body frame and roll, pitch, and yaw axes. (c) Block diagram explaining the experimental setup and the control strategy. The cascaded lateral controller and attitude controller are operated in parallel with the altitude controller.

the attitude dynamics can be described by the Euler equation

$$\mathbf{J}\dot{\omega} = \sum \tau - (\omega \times \mathbf{J}\omega), \quad (1)$$

where $\sum \tau_i$ is the total torque acting on the vehicle. In this work, we only account for the torques generated by the flapping wings as a result of the control command. This is in order to avoid complexity in modeling and reduce computational power required for active control purposes, additional damping aerodynamic effects that arise in free flight [7, 24, 25], which should be insignificant while the robot is stationary during hovering, are regarded as disturbance.

Given an orientation feedback, one can construct a rotation matrix (R) relating the body frame to the inertial frame. It follows that the angular velocity can be written as a function of the body axes and its time derivative, or the rotation matrix and its time derivative as

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \hat{z} \cdot \dot{\hat{y}} \\ \hat{x} \cdot \dot{\hat{z}} \\ \hat{y} \cdot \dot{\hat{x}} \end{bmatrix}. \quad (2)$$

The lateral dynamics of the robot near hovering, when the robot is generating thrust (Γ) equal to its own weight, can be simplified to a two-dimensional second-order system. Assuming the vehicle is only slightly deviated from a vertical orientation, the lateral force generated by the robot is

approximately proportional to the deviation of the robot's \hat{z} axis from the vertical. Therefore, lateral forces in the dynamics can be expressed in terms of the rotation matrix as

$$m \frac{d^2}{dt^2} \begin{bmatrix} X \\ Y \end{bmatrix} = \Gamma \begin{bmatrix} R_{13} \\ R_{23} \end{bmatrix} = mg \begin{bmatrix} R_{13} \\ R_{23} \end{bmatrix}, \quad (3)$$

where m denotes the mass of the robot, g is the gravitational constant, and X, Y are lateral position in the inertial frame.

2.3. System model

In [4, 18], it was shown that the thrust produced by the robotic insect is approximately a linear function of the actuator voltage. The robot was capable of producing thrust larger than 1.3 mN, or more than 1.5 times its own weight. Body torques on the order of one μNm can be achieved by using the three different flapping schemes illustrated in figures 1(c)–(e). Roll torque is generated by differentially changing the stroke amplitude on two wings. Pitch torque is generated by shifting the location of the mid-stroke planes forward or backward. Adding a second harmonic signal into the flapping trajectory results in a difference in stroke velocity on forward and backward strokes. This influences drag forces and alters the wing angle of attack. When applied to both wings in the opposite direction, yaw torque is generated. These torque generation modes are greatly simplified in

comparison to observed insect wing kinematics [21]. Nevertheless, they are often chosen for simulations and dynamically-scaled robot experiments [24, 26, 27].

For control purposes, first the model of the system—a map or a transfer function between input signals and the resultant thrust or torques—has to be identified. The key challenge in obtaining such a model is the lack of a viable multi-axis force/torque sensor. In [4], a custom dual-axis force-torque capacitive sensor similar to the design in [28] was used to measure a single axis of torque and a single force perpendicular to the torque axis. This sensor, therefore, cannot determine the coupling between torques along different axes. Furthermore, despite over a decade of progress in micromanufacturing, there still exists considerable variation between robots. Additionally, the process of mounting the robot on the sensor is challenging and possibly destructive, making it impractical to characterize all robots prior to the flight experiments.

As a consequence, a more theoretical approach is taken. Quasi-steady analyses are often employed to capture the aerodynamic forces in insect flight [21, 29]. Taking a similar approach, in [13], the blade-element method was used to provide estimates of the aerodynamic forces and moments. This study is particularly suitable for our robot as it was carried out using wings operating at the same scale and Reynolds number as our robot. Moreover, passive wing hinges also present on the robot in this work were also used in the experiments and modeling in [13].

The model in [13] is used to compute the estimates of the resultant thrust and body torques using the flapping schemes shown in figures 1(c)–(e). Based on this, we constructed a theoretical approximation of time-averaged thrust and torques as a function of wing trajectory. The findings suggests that, for constant frequency and over a small range of inputs, aerodynamic thrust can be approximated as a linear function of the flapping amplitude, irrespective of other torque input parameters. This is consistent with the empirical evidence in [18]. Similarly, torques about the body axes are approximately linear functions of their respective input parameters as previously reported in [4]. The model further predicts minimal coupling between the three torque generation modes (in agreement with findings and assumptions in related work [24, 26]), and suggests that the torques are also dependent on the flapping amplitude. These can be summarized into a set of equations, as previously presented in [1], as followings:

$$\begin{aligned} \Gamma &= a_t \Phi_0 - b_t \\ \tau_r &= (a_r \Phi_0 - b_r) \delta_r \\ \tau_p &= (a_p \Phi_0 - b_p) \delta_p \\ \tau_y &= (a_y \Phi_0 - b_y) \eta, \end{aligned} \quad (4)$$

where Γ denotes the thrust, τ_i 's represent roll, pitch, and yaw torques, Φ_0 is the flapping amplitude, δ_r is the differential stroke angle, δ_p is the shift in mean stroke angle, η is a relative proportion of a second-harmonic signal used for generating imbalanced drag forces, and the a_i and b_i terms are constants resulting from the linearization.

Once we have obtained the required wing trajectory for the desired thrust and torques, the corresponding actuator drive signals are calculated by approximating the actuator-transmission-wing system as a second order linear system [14]. To elaborate, the piezoelectric actuator also acts as a spring-like component to store elastic energy while driving the wing inertia, inducing an oscillatory behavior akin to the coupled muscle–thorax–wing system in flies [19]. As a consequence, a shift in the mean stroke angle, for example, translates to a constant offset in the drive signal. The model enables us to calculate the voltage amplitudes and offsets required to generate thrust to stay aloft and torques for control. Based on the predictions, one could also ensure that the total voltage required does not exceed the maximum actuator voltage.

3. Controller design

Driven by the lack of both empirical measurements and an accurately identified model of the robot as stated in section 2.3, we employed an adaptive controller in order to estimate unknown parameters. The overall flight controller, initially presented in [15], is comprised of three sub-controllers: a lateral controller, an attitude controller, and an altitude controller. In comparison with with Dipteran insects, in our robot the lateral controller has slower dynamics and can be associated with the optomotor control system in insects, whereas the attitude controller holds a role similar to the mechanosensory system for rapid feedback. Here the lateral controller takes position feedback from a motion capture system and determines the desired orientation of the robot in order to maneuver the robot to a position setpoint. This desired orientation serves as the setpoint for the attitude controller that evaluates the required torques from the vehicle to achieve the desired attitude. In parallel, the altitude controller computes the suitable thrust force to maintain the robot at the desired height based on the position feedback. The block diagram representing these controllers is presented in figure 2(c). These controllers are considerably different from those in [1] as they employ the use of sliding mode control techniques [30] for adaptive purposes. Moreover, higher order models of lateral and altitude dynamics are implemented to reduce the oscillating behaviors seen in the results from [1].

3.1. Adaptive attitude controller

A consensus drawn from several stability studies indicates that, similar to insect flight, flapping-wing MAVs in hover are unstable without active control [6, 7]. Together with uncertainties due to an incomplete model of the vehicle and the requirement to vary the attitude setpoint for lateral maneuvers, it is necessary to design a robust controller that allows for significant excursions from the hovering state. As opposed to traditional linear controllers based on a linearization about hover, we employ Lyapunov's direct method to design a controller with a large domain of attraction. The attitude controller employed here is distinct from the one that

demonstrated the first successful flights in [1] as it enables better tracking and adaptive ability for uncertain parameter estimates.

The goal of the attitude controller is to align the robot \hat{z} axis with the desired attitude vector \hat{z}_d , without specifically control the \hat{x} and \hat{y} axes. Such a strategy allows the robot to maneuver in the desired direction while relaxing control over exact yaw orientation. In other words, the robot has no preference to roll or pitch, but a combination of them would be chosen so that the body \hat{z} axis aligns with the desired attitude vector \hat{z}_d with minimum effort.

Based on a sliding control approach [30], we begin by defining a sliding surface composed of an angular velocity vector ω and the attitude error \mathbf{e} ,

$$\mathbf{s}_a = \omega + \Lambda \mathbf{e}, \quad (5)$$

where Λ is a positive diagonal gain matrix. The attitude error \mathbf{e} is selected to correspond to the amount of the deviation of \hat{z} from \hat{z}_d ,

$$\begin{aligned} \mathbf{e} &= [\hat{y} \cdot \hat{z}_d \quad -\hat{x} \cdot \hat{z}_d \quad 0]^T \\ &= \begin{bmatrix} R_{12} & R_{22} & R_{32} \\ -R_{11} & -R_{21} & -R_{31} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{z}_{d1} \\ \hat{z}_{d2} \\ \hat{z}_{d3} \end{bmatrix}. \end{aligned} \quad (6)$$

Note that the third element of the attitude error vector is zero, consistent with the decision not to control the exact yaw orientation. The composite variable \mathbf{s}_a is zero when \hat{z} aligns with \hat{z}_d and the robot has no angular velocity, hence the controller would only command a yaw torque to neutralize the yaw rate and disregard the yaw orientation. Let $\hat{\alpha}$ be a vector containing the estimates of unknown parameters and $\tilde{\alpha}$ be the estimation error defined as $\tilde{\alpha} = \hat{\alpha} - \alpha$, we propose the following Lyapunov function candidate

$$V_a = \frac{1}{2} \mathbf{s}_a^T \mathbf{J} \mathbf{s}_a + \frac{1}{2} \tilde{\alpha}^T \Gamma^{-1} \tilde{\alpha}, \quad (7)$$

here Γ is a positive diagonal adaptive gain matrix. Assuming that the robot also produces some unknown constant torques $-\tau_o$ in addition to the commanded torque τ_c by the controller, equation (1) can be rewritten as

$$\mathbf{J} \dot{\omega} = \tau_c - \tau_o - (\omega \times \mathbf{J} \omega). \quad (8)$$

As a result, from equations (5), (7) and (8), the time derivative of the Lyapunov function is given by

$$\dot{V}_a = \mathbf{s}_a^T (\tau_c - \tau_o - (\omega \times \mathbf{J} \omega) + \mathbf{J} \Lambda \dot{\mathbf{e}}) + \tilde{\alpha}^T \Gamma^{-1} \dot{\tilde{\alpha}}. \quad (9)$$

Defining $\hat{\mathbf{J}}$ as the estimate of the inertia matrix, we propose the control law

$$\tau_c = -K_a \mathbf{s}_a + \hat{\tau}_o - (\Lambda \mathbf{e} \times \hat{\mathbf{J}} \omega) - \hat{\mathbf{J}} \Lambda \dot{\mathbf{e}} \quad (10)$$

$$= -K_a \mathbf{s}_a + Y \hat{\alpha}, \quad (11)$$

where K is a positive diagonal gain matrix, $\hat{\tau}_o$ is an estimate of the unknown offset torque τ_o , and the matrix Y and the parameter estimate vector $\hat{\alpha}$ are

$$Y = \begin{bmatrix} -\Lambda \dot{\mathbf{e}}_1 & \Lambda \dot{\mathbf{e}}_3 \omega_y & -\Lambda \dot{\mathbf{e}}_2 \omega_z \\ -\Lambda \dot{\mathbf{e}}_3 \omega_x & -\Lambda \dot{\mathbf{e}}_2 & \Lambda \dot{\mathbf{e}}_1 \omega_z \\ \Lambda \dot{\mathbf{e}}_2 \omega_x & -\Lambda \dot{\mathbf{e}}_1 \omega_y & -\Lambda \dot{\mathbf{e}}_3 \end{bmatrix} I_{3 \times 3}$$

$$\hat{\alpha} = [\hat{\mathbf{J}}_{xx} \quad \hat{\mathbf{J}}_{yy} \quad \hat{\mathbf{J}}_{zz} \quad \hat{\tau}_{o1} \quad \hat{\tau}_{o2} \quad \hat{\tau}_{o3}]^T. \quad (12)$$

Equation (9) then becomes

$$\dot{V}_a = -\mathbf{s}_a^T K_a \mathbf{s}_a + \mathbf{s}_a^T Y \hat{\alpha} + \tilde{\alpha}^T \Gamma^{-1} \dot{\tilde{\alpha}}. \quad (13)$$

This suggests the adaptive law

$$\dot{\hat{\alpha}} = -\Gamma Y^T \mathbf{s}_a, \quad (14)$$

which renders the time derivative of the Lyapunov function to be negative definite,

$$\dot{V}_a = -\mathbf{s}_a^T K_a \mathbf{s}_a \leq 0. \quad (15)$$

According to the invariant set theorem, the system is theoretically almost globally asymptotically stable. That is, the composite variable and the estimation errors converge to zero. The exception occurs when the \hat{z} axis points in the opposite direction to the desired attitude vector. Additionally, notice that no particular representation of rotation is used, hence no care needs to be taken to avoid a singularity or any ambiguity in the choice of representation.

The presented attitude controller has a few benefits over the controller employed in [1]. For instance, it has better tracking ability, and the adaptive part takes into consideration the torque offset errors and uncertainty in the estimate of the inertia and makes corrections based on the feedback.

3.2. Adaptive lateral controller

The lateral controller is designed based on the dynamics described in equation (3). This controller relies on position feedback to compute the desired attitude vector that becomes a setpoint for the attitude controller. An adaptive part is incorporated in order to account for misalignment between the presumed thrust vector and the true orientation of the thrust vector. Moreover, the lateral controller assumes that the response of the attitude controller can be described by a first order differential equation as shown in equation (16)—this consideration was not present in earlier work [1].

$$\frac{d}{dt} \begin{bmatrix} R_{13} \\ R_{23} \end{bmatrix} = \gamma \left(\begin{bmatrix} \hat{z}_{d1} \\ \hat{z}_{d2} \end{bmatrix} - \begin{bmatrix} R_{13} \\ R_{23} \end{bmatrix} \right). \quad (16)$$

Here γ^{-1} is an approximate time constant of the closed-loop attitude dynamics.

The lateral dynamics captured by equation (3) is, in fact, obtained from the assumption that the thrust vector lies perfectly along the body \hat{z} axis. In other words, it assumes that the thrust vector in the inertial frame is given as

$$\begin{aligned} \frac{\Gamma}{|\Gamma|} &= \hat{z} \\ &= R [0 \quad 0 \quad 1]^T. \end{aligned} \quad (17)$$

To compensate for a possible misalignment, we introduce another rotation matrix R_e representing a small rotation that maps the body \hat{z} to the true direction of the thrust vector:

$$\frac{\Gamma}{|\Gamma|} = RR_e [0 \ 0 \ 1]^T. \quad (18)$$

For small constant rotations, R_e can be parametrized by three parameters $|\epsilon_x|, |\epsilon_y|, |\epsilon_z| \ll 1$, such that

$$R_e = \begin{bmatrix} 1 & -\epsilon_z & \epsilon_y \\ \epsilon_z & 1 & -\epsilon_x \\ -\epsilon_y & \epsilon_x & 1 \end{bmatrix}. \quad (19)$$

Therefore, the complete model of the lateral dynamics is obtained by substituting equations (16), (18) and (19) into (3).

$$\begin{aligned} & \gamma^{-1} m \frac{d^3}{dt^3} \begin{bmatrix} X \\ Y \end{bmatrix} + m \frac{d^2}{dt^2} \begin{bmatrix} X \\ Y \end{bmatrix} \\ & = mg \begin{bmatrix} \hat{z}_{d1} \\ \hat{z}_{d2} \end{bmatrix} + \epsilon_x \begin{bmatrix} -R_{12} \\ -R_{22} \end{bmatrix} + \epsilon_y \begin{bmatrix} R_{11} \\ R_{21} \end{bmatrix}. \end{aligned} \quad (20)$$

Subsequently, the controller can be designed based on a similar idea as used for the attitude controller. The sliding surface s_l , and the Lyapunov function candidate V_l are defined as

$$\begin{aligned} s_l &= \left(\frac{d^2}{dt^2} + 2\lambda \frac{d}{dt} + \lambda^2 \right) \begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix} \\ V_l &= \frac{1}{2} \gamma^{-1} s_l^T s_l + \tilde{\beta}^T \Pi^{-1} \tilde{\beta}, \end{aligned} \quad (21)$$

where \tilde{X} and \tilde{Y} are position errors (the difference between the current position and the position setpoint), $\tilde{\beta}$ is a vector containing the estimation errors of γ^{-1} , ϵ_x , and ϵ_y , and Π is an adaptive gain. Given a positive diagonal controller gain matrix K_l , it can be proven that the following control law

$$g \begin{bmatrix} \hat{z}_{d1} \\ \hat{z}_{d2} \end{bmatrix} = -K_l s_l + \frac{d^2}{dt^2} \begin{bmatrix} X \\ Y \end{bmatrix} + Y \hat{\beta}, \quad (22)$$

with the term $Y \hat{\beta}$ written as

$$Y \hat{\beta} = \begin{bmatrix} \frac{d^3}{dt^3} \begin{bmatrix} X \\ Y \end{bmatrix} - \frac{d}{dt} s_l & R_{12}/m & -R_{11}/m \\ & R_{22}/m & -R_{21}/m \end{bmatrix} \begin{bmatrix} \hat{\gamma}^{-1} \\ \hat{\epsilon}_x \\ \hat{\epsilon}_y \end{bmatrix}, \quad (23)$$

and the adaptive law

$$\dot{\hat{\beta}} = -\Pi Y^T s_l, \quad (24)$$

make the time derivative of the proposed Lyapunov function candidate negative definite

$$\dot{V}_l = -s_l^T K_l s_l \leq 0. \quad (25)$$

Again, the invariant set theorem can be applied to ensure the stability of the system. In the case of hovering, the position setpoint is constant. In more general cases, the controller also possesses the ability to track time-varying setpoints as the first

and second derivative of the setpoint are incorporated into the composite variable s_l .

3.3. Adaptive altitude controller

The altitude controller has a structure similar to the lateral controller in the preceding section, but with only one dimensional dynamics and a feedforward term to account for gravity. The input to the altitude dynamics is, however, the commanded thrust. The adaptive part is responsible for estimating the thrust offset and a time constant similar to γ in the lateral controller.

The main assumption on the altitude controller is that the robot orientation is always upright, thus the generated thrust is always aligned with the vertical axis. The primary reason for this assumption is to preserve the limited control authority for the more critical attitude controller. To illustrate, a tilted robot may lose altitude due to a reduction in thrust along the inertial vertical axis. Instead of producing more thrust to compensate, we prioritize control authority to the attitude controller to bring the robot upright and reorient the thrust to the vertical axis.

4. Unconstrained flight experiments

4.1. Experimental setup

Flight control experiments are performed in a flight arena equipped with eight motion capture *VICON* cameras, providing a tracking volume of $0.3 \times 0.3 \times 0.3$ m. The system provides position and orientation feedback by tracking the position of four retroreflective markers at a rate of 500 Hz. Orientation feedback is given in the form of Euler angles that can immediately be converted into a rotation matrix. Computation for control is carried out on external computers using an xPC Target system (*Mathworks*), which operates at 10 kHz for both input sampling and output signal generation. Power is delivered to the robot via a 0.6 m long bundle of four 51-gauge copper wires. The latency of the complete experimental setup was found to be approximately 12 ms—less than two wing beats, comparable to the measured neural delay time of fruit flies [7].

The lack of direct velocity and angular velocity measurement requires us to estimate both velocities via the use of filtered derivatives. The approach allows some attenuation of high frequency disturbances, but the estimates suffer from delays introduced by filter phase shifts. This delay, however, was found to be sufficiently small and did not prevent us from achieving stable hover in the experiments.

4.2. Open-loop trimming

Initially, the vehicle is mounted on a static setup and a high-speed video camera is used to measure the flapping amplitude of the robot at various frequencies to determine the resonant frequency of the system and to characterize a suitable operating point where asymmetry between the two wings is

minimized. Once the operating frequency is chosen, trimming flights are executed in the flight arena in attempt to determine the configuration where the net torque produced by the robot is close to zero.

Open-loop trimming is carried out by commanding the robot to produce constant thrust and torques. Visual feedback and state feedback are used to determine the amount of undesired bias torques. The process is repeated with a new set of offset torques in an attempt to minimize the observed bias torque. Due to the inherent instability of the robot, a successful open-loop flight usually crashes in less than 0.4 s. This emphasizes the need of active control for the vehicle. An automatic switch-off routine is also implemented to cut off the power when the robot deviates more than 60° from vertical to prevent damages from crashing. An example of a well-trimmed open-loop flight, where the robot ascended to more than 4 cm in altitude before crashing, is displayed in figure 3(a).

4.3. Attitude-controlled flight

To evaluate the performance of the attitude controller, we first carried out closed-loop flights without the lateral controller by setting the attitude setpoint in equation (6) to be the upright orientation, bypassing the lateral controller block in figure 2(c). Theoretically, once the attitude and altitude are controlled, the robot should be able to stay aloft for an indefinite period.

Here, information obtained from open-loop flights serve as initial estimates of torque biases for closed-loop flights. It was found that the attitude controller was able to keep the robot in the upright orientation as captured in figure 3(b). However, in the absence of the lateral controller, the robot generally traverses over 20 cm in 1–2 s and immediately crashes once it is outside the tracking volume of the motion capture system. The results emphasize the need for a lateral controller to keep the robot flying in the control volume for an extended period.

4.4. Hovering flight

To begin with, hovering flights are performed using initial estimates of torque biases obtained from open-loop flights. At the beginning of each flight, only the attitude controller and the altitude controller are active. The robot usually takes less than one second to reach the altitude setpoint. The lateral controller is initiated 0.2 s into the flight, however, it is not fully activated until $t = 0.4$ s. Similarly, the adaptive parts are activated 0.8 s into the flight, but are not in full operation until 1.0 s.

Oftentimes, the parameter estimates derived from open-loop trimming flights are sufficiently accurate for the robot to stay aloft for a few seconds while the adaptive parts enhance the performance of the controller by adjusting these estimates. Nevertheless, parameter estimates learned at the end of each flight are incorporated into the controller as new estimates. These include the torque offset (τ_e) in the attitude controller, orientation misalignment in the lateral controller (ϵ_x, ϵ_y), and the thrust offset in the altitude controller.

In the absence of mechanical fatigue, after several 5–10 s tuning flights, estimated parameters tend to converge to constant values. At this point, the robot is typically able to maintain its altitude setpoint within a few millimeters, while the lateral precision is on the order of one to two centimeters. It is likely that local air currents or tension from the power wires is the cause of disturbances.

Figures 3(c)–(e) demonstrates an example of a typical hovering flight after parameter convergence. In this case the flight lasts seven seconds, after which the power is cut off. In this sequence, the robot maintained an altitude of 6.0 cm above the ground while it translated laterally around the setpoint. More examples of hovering flights can also be found in the supplementary video (available at stacks.iop.org/BB/9/025004/mmedia).

Regarding the contribution from the wire tether, it is expected that a hanging 10 cm tether would weigh less than 5 milligrams. When hanging 1–2 cm from the robot's body, it could exert a torque in the order of 10^{-8} Nm to the body: this is considerably below the robot's ability to generate up to one μ Nm torque [1, 4]. Nevertheless, occasionally, it could be identified from the video that a wire tether was stuck to the fabric on the floor of the arena, causing the robot to struggle to tilt or translate laterally.

4.5. Lateral maneuver

To demonstrate tracking ability for both the attitude controller and lateral controller and verify the efficiency of accomplishing lateral maneuvers by tilting the body, we illustrate that simple lateral maneuvers along a pre-generated trajectory can be achieved. In this part, a sinusoidal trajectory with the amplitude of 6.0 cm and the period of 3.0 s is chosen. The robot was configured to follow this trajectory for two cycles while retaining its altitude.

Figures 4(a)–(b) shows the result of an example of a lateral flight maneuver. It reveals that throughout 8.0 s of the flight, the robotic insect could maintain its altitude within a few millimeters from the setpoint. The lateral position, however, had a tendency to lag behind the reference trajectory by approximately 0.2 s and occasionally overshoot the target. This is in accord with the plot of the robot's orientations in figure 4(b), in which the robot often failed to catch up with the desired orientation when turning.

4.6. Vertical takeoff and landing

In order to avoid violent crashes and simultaneously demonstrate precise maneuvers, here we show a controlled takeoff and landing flight of a robotic insect, similar to the maneuver reported in [15]. At the time of landing, the translational and angular velocities must be relatively small, otherwise the momentum would cause the robot to crash. Moreover, when the robot approaches the ground, downwash from the flapping wings may introduce disturbances in the form of ground effects as seen in larger flying vehicles, and

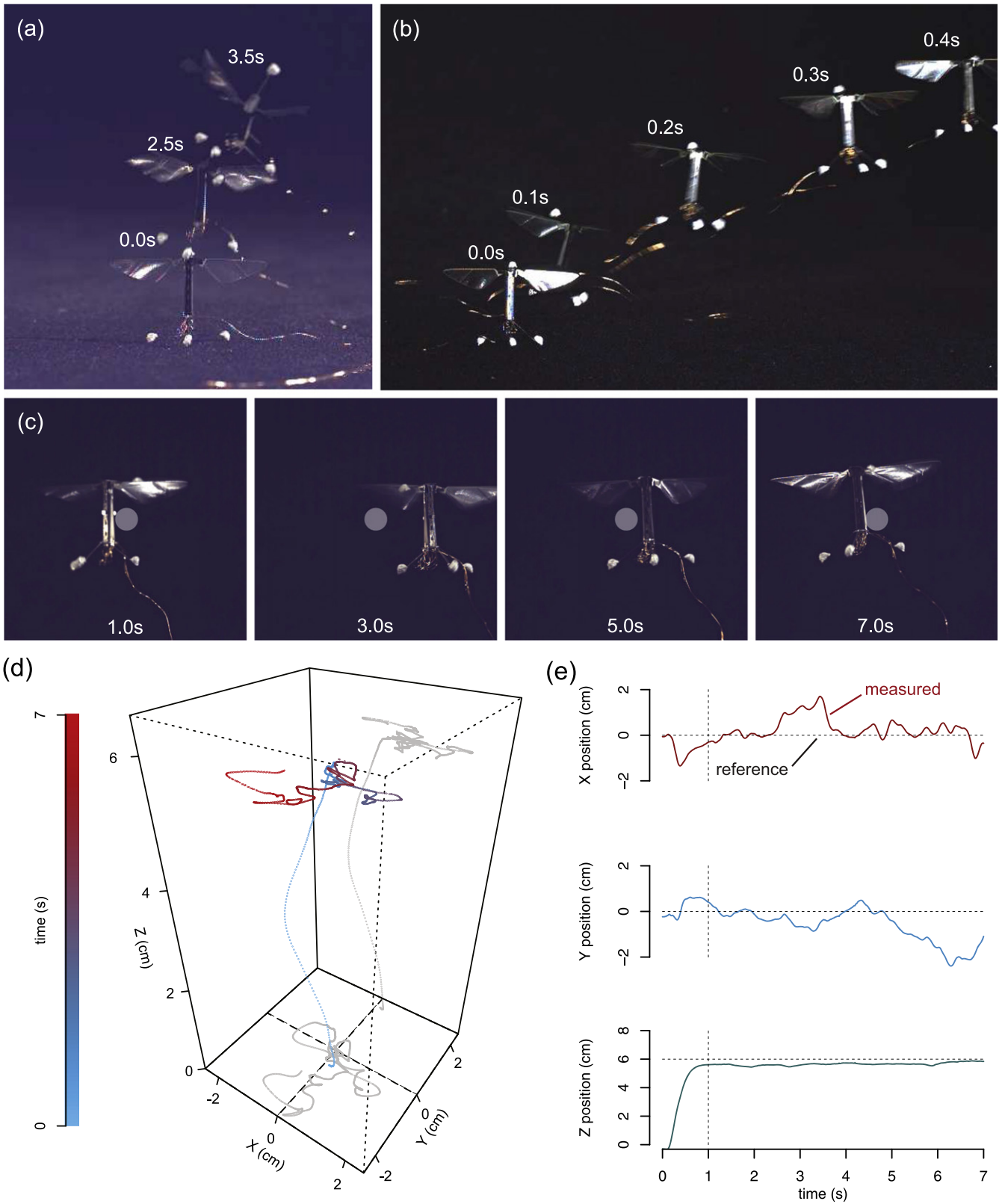


Figure 3. (a) Composite image showing an example of a trimming flight. (b) An composite image of a robotic insect taking of in a attitude-controlled flight. (c) Image sequence illustrating the robot hovering around the setpoint at various times (the grey dots indicate the approximate desired location). (d) A 3D reconstruction of the robot's trajectory during a 7 s hovering flight. (e) A position versus time plots of the robot location in the same flight as in (c) and (d).

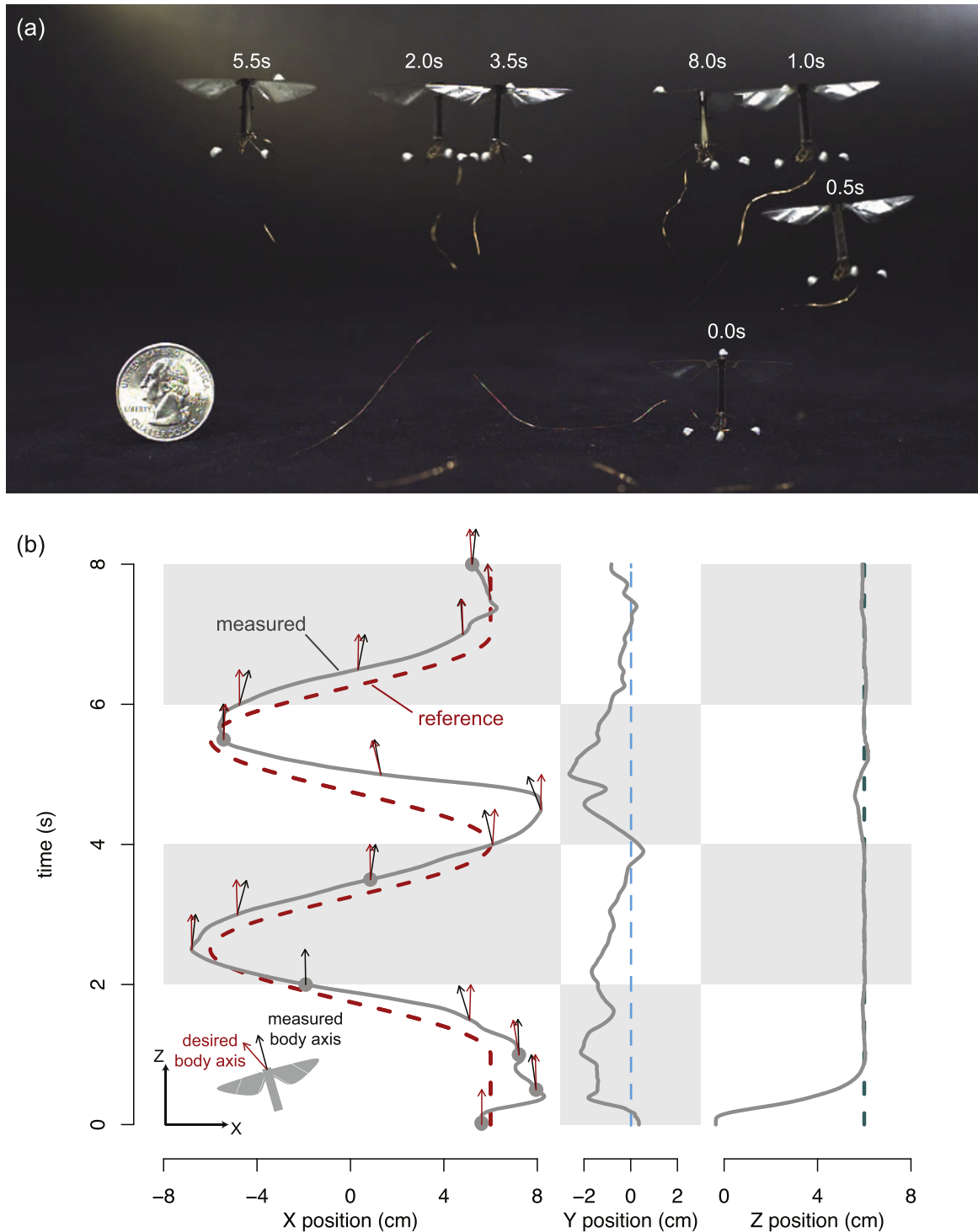


Figure 4. (a) A composite image showing the robot in various locations and times while performing a lateral maneuver. (b) A plot of the robot's position and desired location in a lateral flight. The orientations of the robot's body axis are projected onto the X – Z plane.

destabilize the robot. Here, we illustrate successful landing flights via the use of a simple control strategy with the aid of mechanical landing gear.

The landing gear is designed with two goals: to widen the base of the robot and to absorb the impact of landing. Carbon fiber extensions are attached to the existing structure through a viscoelastic urethane spacer (Sorbothane). A photograph of a robot with the additional landing gear is shown in figure 5(a).

Landing is achieved by slowly reducing the altitude setpoint. To ensure that the robot remains in the nominal upright orientation and stays close to the lateral setpoint, the change in altitude setpoint is suspended when the vehicle is in an unstable state, defined as the l_2 -norm of the composite variable s_a or s_l being larger than the chosen thresholds. Once the robot is below a certain height (≈ 8 mm) above the ground, the driving signals are ramped down, leaving the landing gear to absorb the impact from falling.

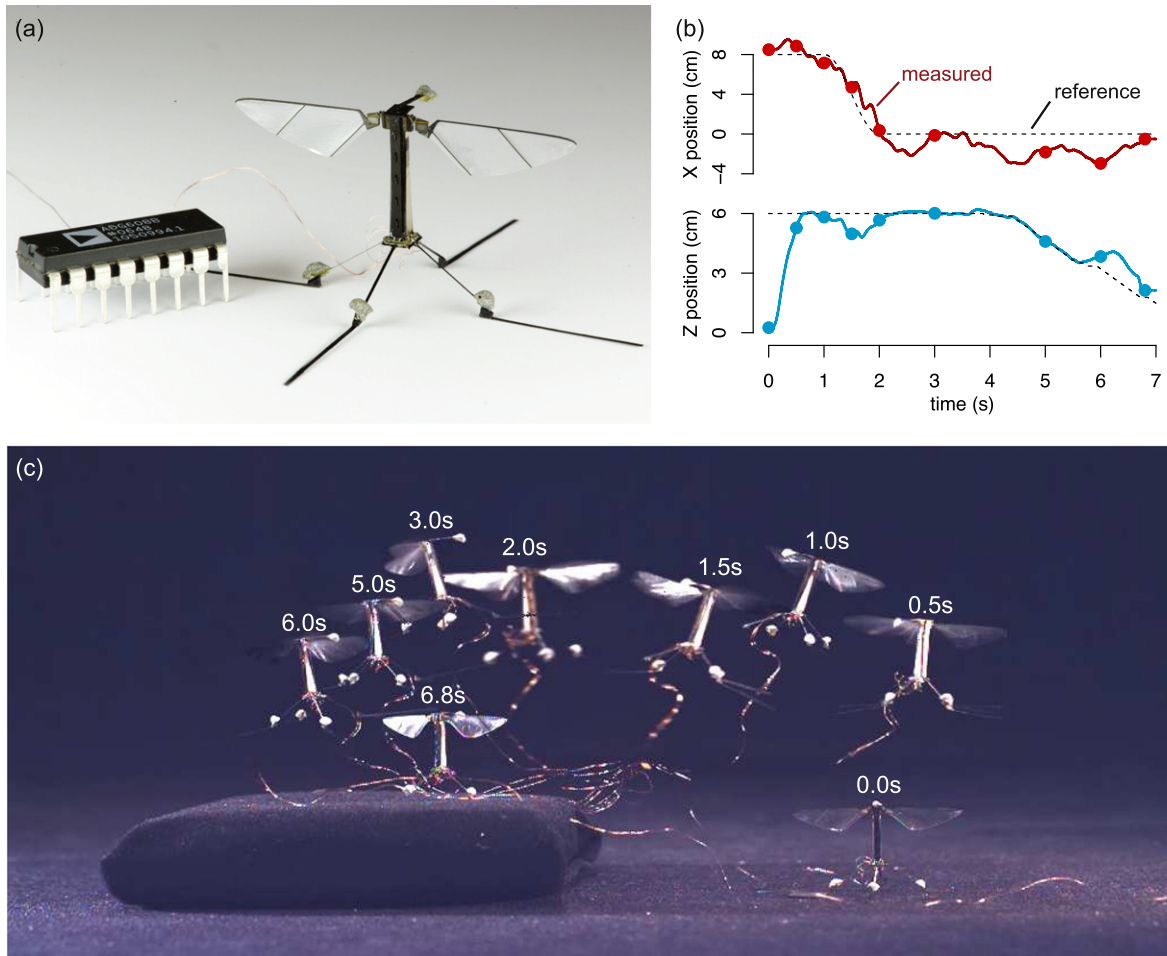


Figure 5. (a) Photograph of the robotic insect with extended landing gears next to a 16-pin dual in-line package (DIP) integrated circuit for scale. The landing mechanism was first shown in [15]. (b) Trajectory plot for an example landing flight. (c) An image illustrating the robot traversing laterally before landing on an elevated platform.

An example trajectory of a successful vertical takeoff and landing flight of the robotic insect is displayed in figures 5(b)–(c). In this case, the robot took off towards the altitude setpoint at 6 cm and started the landing process just after 1.0 s. The nominal landing speed was set at 1.5 cm s^{-1} . According to the plot, the robot followed the altitude setpoint closely. Nevertheless, just after $t = 5.5 \text{ s}$, it can be seen that the landing was briefly suspended as the vehicle drifted away from the lateral setpoint beyond the tolerance. Eventually, the robot reached the pre-defined landing altitude and the power was ramped down after six seconds. Video footage of a few landing flights can also be found in the supplementary video available at stacks.iop.org/BB/9/025004/mmedia.

5. Conclusion and discussion

We have presented a comprehensive flight controller designed for a bio-inspired flapping-wing microrobot. Driven by modeling uncertainty and the nonlinear nature of the system, Lyapunov's direct method was employed to guarantee the stability of the proposed adaptive controllers. We have

demonstrated that, using a simplified set of wing kinematics, as opposed to more sophisticated wing kinematics observed in real insects, the proposed controller is sufficient to realize hovering flights and simple maneuvers.

5.1. Comparison to the non-adaptive controller

To compare that the proposed controller to its non-adaptive counterpart from [1], we consider 25 flight trajectories from the non-adaptive controller and 28 flight trajectories obtained from the adaptive controller. Only portions of stable flights after taking off are selected and consolidated into a non-adaptive dataset and an adaptive dataset. The non-adaptive dataset consists of 125 s (15 000 wingbeats) of flying time while the adaptive dataset is 115 s (13 800 wingbeats) long. These trajectories are plotted along the X , Y , and Z axes relative to the setpoints in figure 6(a). They are overlaid by boxplots representing the mean positions and the standard deviations. It can be seen that the proposed adaptive controller markedly reduces standard errors by over 50%. The improvement is most pronounced along the \hat{Z} direction, thanks to the adaptive altitude controller.

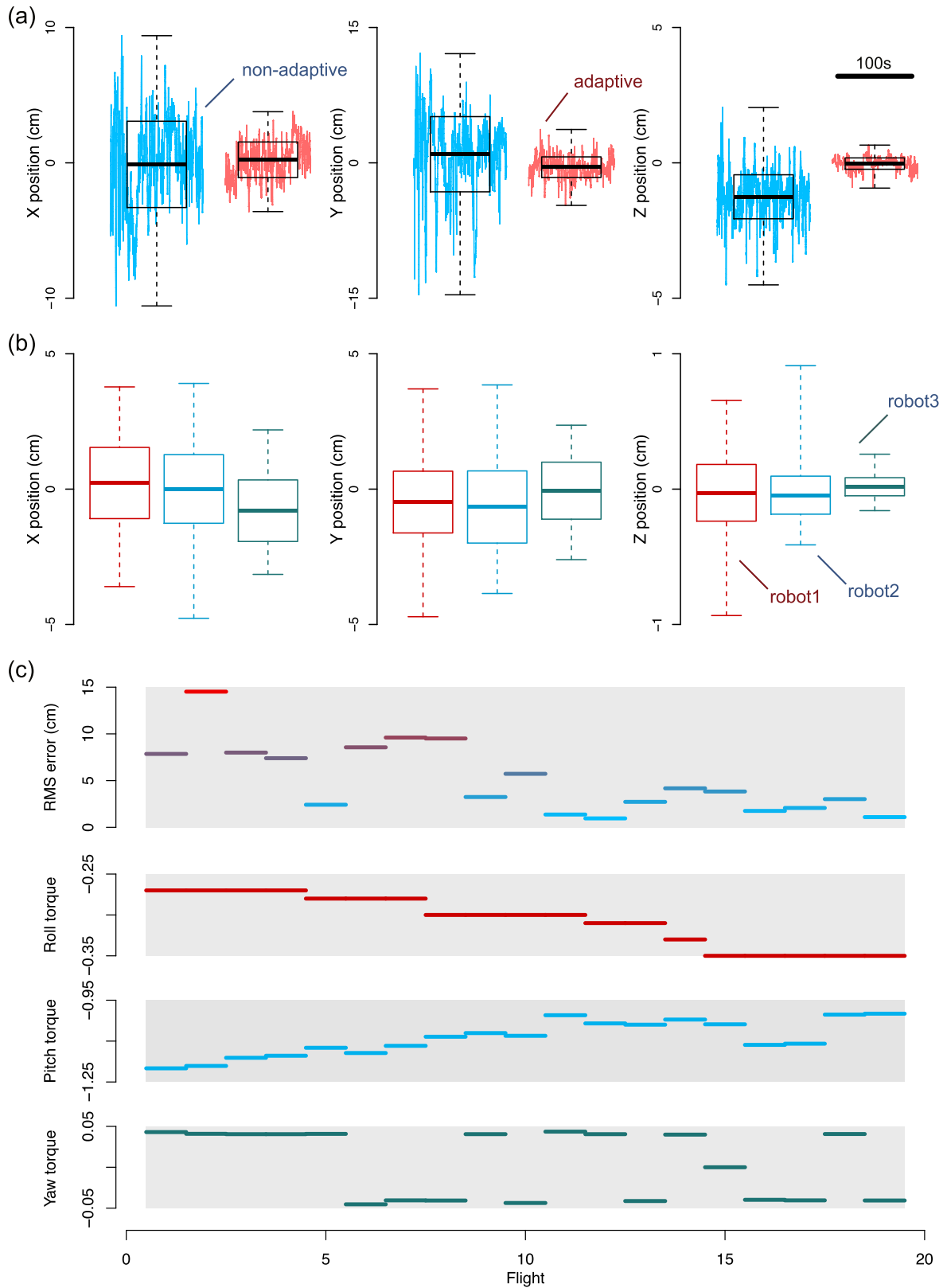


Figure 6. (a) Positions of the robot with respect to the setpoints from 25 non-adaptive flights (blue) and 28 adaptive flights (red) overlaid by boxplots showing the mean positions and the standard deviations. (b) Boxplots showing the averages and the standard deviations in positions of three robot from 28, 9 and 7 flights. (c) Data from 19 consecutive flights after the open-loop trimming for steady hovering. The top plot shows the RMS errors of the lateral position. The bottom three plots show the estimated torque offsets at the beginning of each flight.

Furthermore, to validate that the proposed adaptive controller is capable of coping with possible variations between robots, we carried out hovering flight experiments on two additional robots of the same design. On the second robot, data from nine stable flights, totaling 44.5 s (5 340 wingbeats) of flight time was captured. On the third robot, seven stable flights were obtained, with total flight time of 45.5 s (5 460 wingbeats). Figure 6(b) shows boxplots illustrating the mean positions and the standard deviations of stable hovering flights performed by the three robots using the adaptive controller. The plots verify that the magnitudes of the means and the standard deviations are similar and consistent between the robots. Additionally, we also found that in these robots, the torque offsets converged to markedly different values. For instance, the pitch torque offsets of the second and the third robot were found to be ≈ -0.45 units and ≈ -1.50 units respectively, while the pitch torque offset of the first robot (as demonstrated in 6(c)) was ≈ -1.00 units. Note that the torque values here are given in a normalized unit as the true values are not measured. Nevertheless, one unit is estimated to be in the order of one μNm . The results suggest that the adaptive component of the controller is able to overcome variations due to fabrication imperfections between robots.

Computationally, the adaptive laws of the proposed controller have a simple structure comparable to the integral component of a PID controller. The parameter estimates are given by the integral of terms without any matrix inversion or computationally expensive operations (equations (10) and (24) resulting in no significant computing power requirement relative to a standard PID controller.

5.2. Adaptability

Previous work on control for flapping-wing MAVs has typically been carried out through modeling and simulations [6]. As a result, less emphasis was given to uncertainties in state estimation. Some previous research incorporated integral parts into linear controllers to eliminate steady state errors [26, 31]. It is anticipated that this strategy may work equally well near hovering conditions. Due to the inherent non-linearity of the attitude dynamics, the proposed adaptive controller has the potential to surpass linear controllers at other operating points, when the linearization becomes less accurate. It follows that the adaptive controller concept might resemble insect flight control systems more closely than a PID controller since flying insects often perform saccadic movements and highly aggressive maneuvers, experiencing conditions that are remarkably dissimilar to the hovering state.

Regarding the adaptive ability, one consideration is the extent of modeling uncertainties that the adaptive controller can compensate for. In practice, it was found that it was always necessary to perform open-loop trimming flights to obtain the approximate nominal torque offsets (τ_0). This serves as a starting point for a robot to fly short closed-loop flights prior to crashing. Subsequent flights were executed with the adaptive components and the latest parameter estimates were rounded and used as new starting points for later

flights. Figure 6(c) shows the experimental data from 19 consecutive flights after open-loop trimming. By continuously updating the parameter estimates (in this case the torque offset estimates at the beginning of each flight are shown), the root mean square of the lateral position error decreases from more than 10 cm to less than 2 cm. Similarly, the torque offset estimates seem to eventually converge. The adaptive component is capable of correcting torque offset estimates that are up to 0.2 units, whereas the inherent offset could be as much as one unit. To put these numbers into perspective, for a stable hovering flight, the commanded torques from the controller are generally bounded within ± 0.3 units.

In practice, we occasionally observed unanticipated (both gradual and sudden) changes in the torque offset values caused by mechanical changes in the robot (e.g. wing damage or wing hinge fatigue). Due to the small scale, it is often not possible to identify the source of such changes immediately by inspection. To illustrate, a small crack in wing hinge is usually not observable until it has propagated substantially. While the adaptive component of the controller could usually deal with small and gradual changes as the damage propagates, it is difficult to quantify the damage in terms of the amount of imbalanced torque it causes as we are unable to identify the beginning of the failure. In a hypothetical event of slight damage on the one wing, we consider a simulated scenario involving a reduction of lift from one wing. Typically one wing generates ≈ 40 mg of lift to support the weight of the robot. Assuming the center of pressure is approximately one centimeter from the center of mass, each wing produces about $4 \mu\text{Nm}$ of roll torque. The results shown in figure 6(c) suggests that the adaptive controller would be able to cope with $\approx 0.20 \mu\text{Nm}$ of imbalanced roll torque, which may arise from a sudden 5% reduction in lift from one wing. In a scenario where the damage is gradual as often observed in practice, we expect that the adaptive component would be able to deal with more acute damage.

In our adaptive controller, the adaptive gain can be chosen. In experiments, we have seen that the torque offset estimates could be adapted as fast as 2 unit s^{-1} without losing stability. This is, in many circumstances, sufficiently fast for the robot to adapt for changes that occur during flight (e.g. changes in torque offsets, loss of lift, and the presence of gradual fatigue at the wing hinge). In insects, similar learning behavior has been observed [32, 33]. Insects were found to adapt their flight behaviors to cope with wing damage with noticeable effects on flight performance. The results in this work are in agreement with these biological observations, suggesting the presence of an inherent adaptive or learning ability in the insect flight control systems.

5.3. Robustness analysis

Here we offer a simplified analysis to quantify the effect of unaccounted torque offsets on the flight performance. This could also be seen as a theoretical approach to bound the position errors from possible unmodeled dynamics.

To begin with, we exclude the adaptive part of the controller from the analysis, or regard the adaptive gains as zero. For the attitude controller, equations (10) and (15) reveal that the time derivative of the Lyapunov function candidate is no longer always negative definite when $\tilde{\tau}_o > K_a s_a$. Suppose the robot possesses an inherent torque offset of 0.1 units about the pitch or roll axes, this would translate to a consistent attitude error of 6.5° for the controller gains used in the experiments. Treating that as a misalignment of body axes for the lateral controller, the resultant error in lateral position would be 3 cm. Since this does not take into consideration other unmodeled effects (e.g. disturbances or a delay in the control loop), it is reasonable to anticipate the error to be larger in practice. This is approximately in accordance with the data in figure 6(b), where the lateral error drops from ≈ 10 cm to < 2 cm when the torque offset estimates were altered by ≈ 0.2 units.

At the scale of our robot, to obtain millimeter-scale spatial accuracy while hovering on par with insects, the adaptive component must correct for torque offsets with a resolution of approximately $0.01 \mu\text{Nm}$ resolution. This suggests that some insects are capable of recognizing extremely small steady state errors and finely tuning their wing kinematics in response. Since the adaptive process involves integration and does not require rapid feedback, it is likely that insects handle such operation in the primary sensory-motor systems, rather than using the specialized low-latency sensory-based equilibrium reflexes typically used for attitude stabilization [19, 21].

Acknowledgments

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

References

- [1] Ma K Y, Chirarattananon P, Fuller S B and Wood R J 2013 Controlled flight of a biologically inspired insect-scale robot *Science* **340** 603–7
- [2] Wood R J, Steltz E and Fearing R S 2005 Optimal energy density piezoelectric bending actuators *Sensors Actuators A* **119** 476–88
- [3] Sreetharan P S, Whitney J P, Strauss M D and Wood R J 2012 Monolithic fabrication of millimeter-scale machines *J. Micromech. Microeng.* **22** 055027
- [4] Ma K Y, Felton S M and Wood R J 2012 Design, fabrication and modeling of the split actuator microrobotic bee *IROS 2012: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 1133–40
- [5] Finio B M and Wood R J 2012 Open-loop roll, pitch and yaw torques for a robotic bee *IROS 2012: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 113–9
- [6] Orłowski C T and Girard A R 2012 Dynamics, stability and control analyses of flapping wing micro-air vehicles *Prog. Aerosp. Sci.* **51** 18–30
- [7] Ristroph L, Ristroph G, Morozova S, Bergou A J, Chang S, Guckenheimer J, Wang Z J and Cohen I 2013 Active and passive stabilization of body pitch in insect flight *J. R. Soc. Interface* **10** 85
- [8] de Croon G, de Clercq K, Ruijsink R, Remes B and de Wagter C 2009 Design, aerodynamics and vision-based control of the delfly *Int. J. Micro Air Vehicles* **1** 71–97
- [9] Keennon M, Klingebiel K, Won H and Andriukov A 2012 Development of the nano hummingbird: a tailless flapping wing micro air vehicle *Proc. 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* pp 1–24
- [10] Mellinger D, Michael N and Kumar V 2012 Trajectory generation and control for precise aggressive maneuvers with quadrotors *Int. J. Robot. Res.* **31** 664–74
- [11] Muller M, Lupashin S and D'Andrea R 2011 Quadcopter ball juggling *IROS 2011: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 5113–20
- [12] Kendoul F 2012 Survey of advances in guidance, navigation and control of unmanned rotorcraft systems *J. Field Robot.* **29** 315–78
- [13] Whitney J and Wood R 2010 Aeromechanics of passive rotation in flapping flight *J. Fluid Mech.* **660** 197–220
- [14] Finio B M, Pérez-Arancibia N O and Wood R J 2011 System identification and linear time-invariant modeling of an insect-sized flapping-wing micro air vehicle *IROS 2011: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 1107–14
- [15] Chirarattananon P, Ma K Y and Wood R J 2013 Adaptive control for takeoff, hovering and landing of a robotic fly *IROS 2013: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*
- [16] Xu R and Ozguner U 2006 Sliding mode control of a quadrotor helicopter *Proc. 45th IEEE Conference on Decision and Control (2006)* pp 4957–62
- [17] Lee D, Kim H J and Sastry S 2009 Feedback linearization versus adaptive sliding mode control for a quadrotor helicopter *Int. J. Control, Autom. Syst.* **7** 419–28
- [18] Pérez-Arancibia N O, Ma K Y, Galloway K C, Greenberg J D and Wood R J 2011 First controlled vertical flight of a biologically inspired microrobot *Bioinspir. Biomim.* **6** 036009
- [19] Dudley R 2002 *The Biomechanics of Insect Flight: Form, Function, Evolution* (Princeton, NJ: Princeton University Press)
- [20] Finio B M and Wood R J 2010 Distributed power and control actuation in the thoracic mechanics of a robotic insect *Bioinspir. Biomim.* **5** 045006
- [21] Taylor G K 2001 Mechanics and aerodynamics of insect flight control *Biol. Rev.* **76** 449–71
- [22] Zanker J 1988 On the mechanism of speed and altitude control in *Drosophila melanogaster* *Physiol. Entomology* **13** 351–61
- [23] Götz K G and Wehrhahn C 1984 Optomotor control of the force of flight in *drosophila* and *musca* *Biol. Cybern.* **51** 129–34
- [24] Dickson W B, Polidoro P, Tanner M M and Dickinson M H 2010 A linear systems analysis of the yaw dynamics of a dynamically scaled insect model *J. Exp. Biol.* **213** 3047–61
- [25] Sun M, Wang J and Xiong Y 2007 Dynamic flight stability of hovering insects *Acta Mech. Sin.* **23** 231–46
- [26] Dickson W B, Straw A D, Poelma C and Dickinson M H 2006 An integrative model of insect flight control *Proc. 44th AIAA Aerospace Sciences Meeting and Exhibit* (doi:10.2514/6.2006-34)
- [27] Oppenheimer M W, Doman D B and Sigthorsson D O 2010 Dynamics and control of a biomimetic vehicle using biased

- wingbeat forcing functions: Part I. Aerodynamic model *Proc. 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Exposition (Orlando, FL, Jan. 2010)*
- [28] Finio B M, Galloway K C and Wood R J 2011 An ultra-high precision, high bandwidth torque sensor for microrobotics applications *IROS 2011: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 31–38
- [29] Bos F M, Lentink D, van Oudheusden B and Bijl H 2008 Influence of wing kinematics on aerodynamic performance in hovering insect flight *J. Fluid Mech.* **594** 341–68
- [30] Slotine J-J E, Li W *et al* 1991 *Applied Nonlinear Control* vol 199 (Upper Saddle River, NJ: Prentice Hall)
- [31] Geder J, Ramamurti R, Sandberg W and Flynn A 2012 Modeling and control design for a flapping-wing nano air vehicle *Proc. Conf. AIAA Guidance Navigation and Control (Toronto, Aug. 2010)*
- [32] Dukas R and Dukas L 2011 Coping with nonrepairable body damage: effects of wing damage on foraging performance in bees *Animal Behav.* **81** 635–8
- [33] Combes S, Crall J and Mukherjee S 2010 Dynamics of animal movement in an ecological context: dragonfly wing damage reduces flight performance and predation success *Biol. Lett.* **6** 426–9