

Single-Loop Control and Trajectory Following of a Flapping-Wing Microrobot

Pakpong Chirarattananon, Kevin Y Ma, and Robert J Wood

Abstract—Inspired by the agility of flying insects and the recent development on an insect-scale aerial vehicle, we propose a single-loop adaptive flight control suite designed with an emphasis on the ability to track dynamic trajectories as a step towards the goal of performing acrobatic maneuvers as observed in real insects. Instead of the conventional approach of having cascaded control loops, the proposed controller directly regulates the commanded torques to stabilize the attitude and lateral position in a single loop. The method is verified by performing trajectory following flights with the insect-like robot. The results show that the position errors during trajectory following flights are comparable to those observed from steady hovering flights.

I. INTRODUCTION

Insects are amongst the most diverse groups of animals on the planet. Flying insects are capable of exhibiting complex aerial feats unmatched by other flying animals. The exceptional maneuverability of these flying insects inspires biologists to advance their understanding of flapping-wing aerodynamics and insect flight [1], [2], [3], prompting several efforts to develop insect-scale aerial vehicles [4], [5].

One of the recent successful prototypes from the RoboBees project demonstrated its first unconstrained flight [6]. The 80mg *Micro Aerial Vehicle* (MAV) shown in figure 1 is a result of the culmination of research in meso-scale actuation and manufacturing technology [7], [8]. The flapping-wing robot is able to generate body torques and sufficient lift force [9] satisfying the key requirements for stable flight with the aid of an active flight controller [6].

In an effort to improve the flight performance demonstrated in [6], the lack of comprehensive knowledge of the system and variation caused by imperfect fabrication motivated the development of a suite of adaptive flight controller capable of coping with model uncertainties [10]. This brought about marked improvement in flight performance as evidenced by a reduction in position errors, particularly for hovering flights. Moreover, the fidelity of this flight controller was further demonstrated in vertical takeoff and landing flights.

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.

The authors are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA, and the Wyss institute for Biologically Inspired Engineering, Harvard University, Boston, MA, 02115, USA (email: chirarat@fas.harvard.edu; kevinma@seas.harvard.edu; rjwood@eecs.harvard.edu).

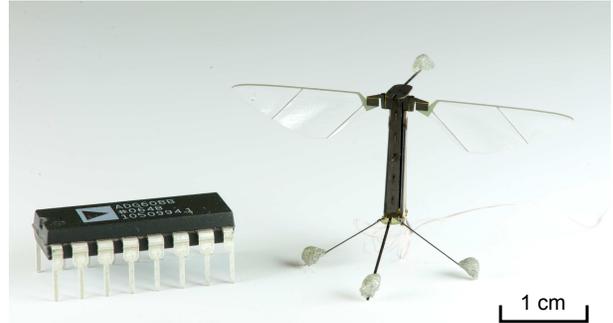


Fig. 1. Photograph of a biologically-inspired flapping-wing robot next to a 16-pin dual in-line package (DIP) integrated circuit for scale.

In flight control of MAVs of similar dynamics—quadrotors, it is common to divide a controller into an inner loop and an outer loop [11], [12], [13]. This also applies to our previous work in [10]. This approach relies on the assumption that the dynamics of the inner loop is significantly faster than that of the outer loop, hence two loops could be arranged in a cascaded configuration. In this case of [10], the inner loop, which control the attitude dynamics, takes the output from the lateral controller in the form of an attitude setpoint as its input. While this was shown to be an effective method to produce steady hovering flights, it is conceivable that the cascaded control architecture may lead to unavoidable delay due to the mentioned assumption, rendering the controller unsuitable for more aggressive flight trajectories.

As a consequence, in this paper we present the development of a flight controller that discard the cascaded structure, integrating the lateral controller and the attitude controller into a single block. Not only does this eliminate the delay in the control loop, but it also effectively improves the position tracking ability by taking into account the third and fourth order derivative of the desired position while generating the control outputs. We verified the capability of the proposed controller in trajectory following flights using smooth trajectories generated from an optimization routine outlined in the later part of the paper.

Notation:

- In equations, bold letters indicate vectors.
- Given an *unknown parameter* α , its estimate is represented by $\hat{\alpha}$. The estimation error $\tilde{\alpha}$ is defined as $\tilde{\alpha} = \hat{\alpha} - \alpha$.
- Otherwise, $\hat{\cdot}$ represents a unit vector.
- High order derivatives are denoted by bracketed superscript, i.e., $\alpha^{(n)} = d^n\alpha/dt^n$.

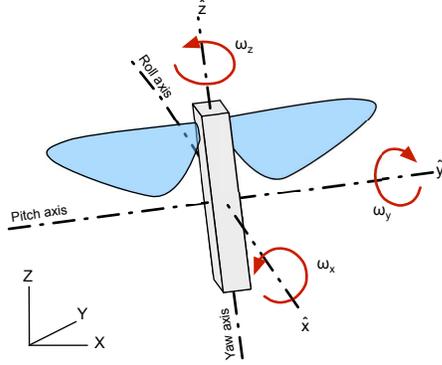


Fig. 2. Definitions of the inertial frame, the body frame, and roll, pitch, and yaw axes.

II. ROBOT DESIGN AND DYNAMIC MODEL

A. Robot design

The robot prototype used in this paper was first presented in [9]. The robot is fabricated in the Harvard Micro-robotics Laboratory using the *Smart Composite Microstructures* (SCM) manufacturing process [14], [9]. The robot in figure 1 weighs 80mg and has a wing span of 3.5cm. It is composed of two bimorph piezoelectric actuators, serving as flight muscles. By applying a voltage across the piezoelectric plates, the actuator acts as a bending cantilever beam. The trajectory of the actuator tip approximates a linear motion. This is then translated into a rotational flapping motion of the wing via a flexure-based four-bar transmission.

The actuator, transmission, and wing compose a mechanical system with properties resembling a second order linear system, roughly equivalent to an ideal forced mass-spring-damper system [15]. Energy storage and dissipation is dominated by the elastic potential energy of the bending actuator, the kinetic energy of the flapping wing and surrounding fluid, and the aerodynamic damping from air drag on the wing.

As a consequence, nominal sinusoidal flapping motion is achieved by using a sinusoidal driving signal. By keeping the frequency constant around the system resonant frequency ($\approx 120\text{Hz}$), the flapping stroke amplitude is maximized. Thrust can be regulated by amplitude modulation. Alteration of wing trajectories allows the robot to generate body torques along its pitch, roll, and yaw axes as defined in figure 2. More details on the generation of body torques and respective flapping schemes can be found in [6], [10].

Using a combination of theoretical approximation and empirical data, we generated a mapping between input signals and four output signals: thrust, roll torque, pitch torque, and yaw torque [6], [10]. This leaves the remaining task of devising a controller to determine the thrust and torques required by the robot to realize pre-planned trajectories.

B. Dynamic model

Ignoring oscillating components, we treat the robot as a rigid body in three dimensional space—the standard approach

taken in the literature [2], [16]. In body attached coordinates, roll, pitch, and yaw axes and their respective angular velocities ω_x , ω_y , and ω_z are defined along \hat{x} , \hat{y} , and \hat{z} directions as illustrated in figure 2.

Given the symmetry of the vehicle, we assume that the moment of inertia matrix J is diagonal and its elements are J_x , J_y , and J_z . The attitude dynamics in the body frame depends on the total torque acting on the robot and is described by the Euler's equation for rigid body dynamics:

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

Let R denote a rotation matrix relating the orientation of a vector in the body frame into the inertial frame, it can also be presented as $R = [\hat{x} \ \hat{y} \ \hat{z}]$. Its derivative can be written as a function of angular velocity and vice versa.

$$\dot{R} = \begin{bmatrix} \omega_z \hat{y} - \omega_y \hat{z} & \omega_x \hat{z} - \omega_z \hat{x} & \omega_y \hat{x} - \omega_x \hat{y} \end{bmatrix},$$

$$\begin{aligned} \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix} &= \begin{bmatrix} \hat{z} \cdot \dot{\hat{y}} & \hat{x} \cdot \dot{\hat{z}} & \hat{y} \cdot \dot{\hat{x}} \end{bmatrix} \\ &= \begin{bmatrix} R_{13}\dot{R}_{12} + R_{23}\dot{R}_{22} + R_{33}\dot{R}_{32} \\ R_{11}\dot{R}_{13} + R_{21}\dot{R}_{23} + R_{31}\dot{R}_{33} \\ R_{12}\dot{R}_{11} + R_{22}\dot{R}_{21} + R_{32}\dot{R}_{31} \end{bmatrix}^T. \end{aligned}$$

The translational dynamics of the robot also depends on the orientation of the robot. In other words, the *normalized* thrust Γ (which has a dimension of acceleration, not force) is nominally aligned with the \hat{z} axis of the robot. It follows that we can write the equation of motion of the robot as

$$m \begin{bmatrix} \ddot{X} & \ddot{Y} & \ddot{Z} \end{bmatrix}^T = m\mathbf{g} + m\Gamma\hat{z}, \quad (2)$$

where m denotes the mass of the robot, \mathbf{g} is a gravity vector, and X , Y , and Z are position of the robot in the inertial frame.

Note that we have not taken into consideration additional aerodynamic effects that arise in free flight. Such effects, including unsteady flow, are difficult to capture using simple models that are suitable for real-time control purposes [3], [17]. As a result, they are regarded as unmodeled dynamics by the controller.

III. CONTROLLER DESIGN

The inherent instability of flapping-wing MAVs [2], [3] requires active flight control. To prevent the robot from crashing, the attitude controller must nominally align the robot's thrust vector against gravity. In [6], we demonstrated that using a flight controller that possessed a large region of attraction over the $SO(3)$ space, the flapping-wing robot achieved stable hovering flights. One distinctive character of the proposed controller is the relaxation of control over the exact yaw orientation as it is dispensable in controlling the heading of the robot, and hence the lateral position of the vehicle.

To attain more precise hovering, we identified several critical unknown parameters that significantly affect the flight performance and re-designed an adaptive flight controller using sliding mode control techniques [10]. In this case, the

lateral position of the robot is regulated by changing the attitude setpoint of the robot, while the altitude is controlled separately. In other words, the lateral controller and the attitude controller operate in a cascaded fashion. The lateral controller determines the attitude setpoint by assuming that the attitude dynamics are considerably faster than the lateral dynamics, and therefore the closed-loop attitude dynamics can be treated as a first order lowpass system. In the mean time, the attitude controller attempts to realize the attitude setpoint and minimize the angular velocity of the robot. The block diagram summarizing key components of this control architecture is shown in figure 3. This markedly improved the accuracy in position and substantially reduced visible oscillations during hovering flights as compared with [6]. Simple lateral maneuvers were also demonstrated, nonetheless, there was significant room for improvement.

Another drawback of cascading the lateral controller and the attitude controller in the way implemented in [10] is that the attitude controller always tries to minimize the rotational rate. Since the rotational rate is related to the third order derivative of the position, it is anticipated that a controller that also determines a suitable angular velocity setpoint would bring about better performance in trajectory following, particularly when more aggressive movements are involved.

However, it is not trivial to evaluate the angular velocity setpoint from the pre-planned trajectory and the current state of the vehicle. The complication arises as we try to retain provable Lyapunov stability while the exact yaw orientation of the robot is not directly controlled. In this paper, we propose a Lyapunov function that comprises of variables made of various derivatives of position error projected on to suitable directions. The outcome is a lateral position controller that directly regulates the desired torques from position error, bypassing the attitude controller and, thus, dropping the assumption regarding the response of the attitude controller. The product is a more versatile controller that is capable of more aggressive trajectory following in addition to only steady hovering.

In this section, we present the derivation of the proposed altitude controller and the lateral position controller based on the sliding mode control method. Although they are presented separately as illustrated in figure 4, they operate in parallel rather than in a cascaded configuration. As a consequence, they can technically be classified as a single control loop. Later, the controller is modified to accommodate an adaptive component and the stability is verified via Lyapunov's direct method.

A. Altitude Control

To begin, we define a position vector (\mathbf{r}) and the desired position vector (\mathbf{r}_d) with respect to the inertial frame:

$$\begin{aligned} \mathbf{r} &= [X \ Y \ Z]^T \\ \mathbf{r}_d &= [X_d \ Y_d \ Z_d]^T. \end{aligned} \quad (3)$$

Given the robot's thrust Γ and the gravity vector \mathbf{g} from equation (2), the translational dynamics of the robot is

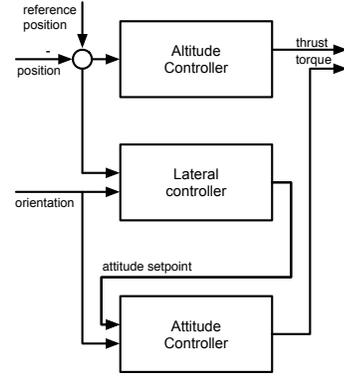


Fig. 3. A block diagram illustrating the structure of the flight controller in [10]. Here the lateral controller computes the attitude setpoint as an input for the attitude controller.

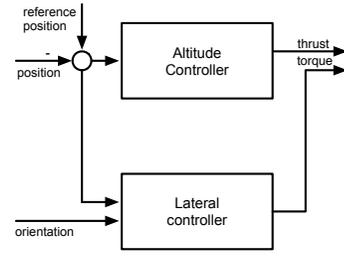


Fig. 4. A simplified block diagram showing the underlying structure of the proposed single-loop controller. The attitude controller is incorporated into the lateral controller which operates in parallel to the altitude controller.

described by

$$\begin{aligned} \ddot{\mathbf{r}} &= \Gamma \hat{\mathbf{z}} + \mathbf{g} \\ &= \Gamma [R_{13} \ R_{23} \ R_{33}]^T - [0 \ 0 \ g]^T. \end{aligned} \quad (4)$$

The altitude dynamics are given by the third row of equation (4).

$$\begin{aligned} \ddot{Z} &= \ddot{\mathbf{r}} \cdot [0 \ 0 \ 1]^T \\ &= \Gamma (\hat{\mathbf{z}} \cdot [0 \ 0 \ 1]^T) - g. \end{aligned} \quad (5)$$

For control purposes, we define a sliding surface variable S_Γ and the variable \ddot{Z}_r as the following:

$$\begin{aligned} S_\Gamma &= (\ddot{Z} - \ddot{Z}_d) + \Lambda_1 (\dot{Z} - \dot{Z}_d) + \Lambda_2 (Z - Z_d) \\ &= \ddot{Z} - \ddot{Z}_r, \end{aligned} \quad (6)$$

where Λ_i 's are positive constants. In this work, we presume that the generated normalized thrust is approximately a lowpass filtered signal of the thrust input T , with γ being a respective filter coefficient such that $\dot{\Gamma} = \gamma(T - \Gamma)$. The time derivative of S_Γ is

$$\begin{aligned} \dot{S}_\Gamma &= \dot{\Gamma} R_{33} + \Gamma \dot{R}_{33} - \frac{d}{dt} \ddot{Z}_r \\ &= \gamma(T - \Gamma) R_{33} + \Gamma (-R_{32} \omega_x + R_{31} \omega_y) - \frac{d}{dt} \ddot{Z}_r \end{aligned} \quad (7)$$

Here we propose a Lyapunov candidate function

$$V_\Gamma = \frac{1}{2} S_\Gamma^2.$$

Subsequently, the following control law

$$T = \Gamma - \gamma^{-1} R_{33}^{-1} \left[\Gamma (-R_{32}\omega_x + R_{31}\omega_y) - \frac{d}{dt} \ddot{Z}_r + K_\Gamma S_\Gamma \right],$$

with a positive constant gain K_Γ and the measured thrust given as $\Gamma = \|\ddot{\mathbf{r}} + \mathbf{g}\|$ render the derivative of the Lyapunov function negative definite

$$\dot{V}_\Gamma = S_\Gamma \dot{S}_\Gamma = -K_\Gamma S_\Gamma^2 \leq 0.$$

According to the invariant set theorem, the system is stable in a Lyapunov sense [18].

B. Lateral position control

Since the angular velocity is related to the third-order derivative of the robot's position, we consider a variable \mathbf{e} , made up of the differences between the robot's position and the setpoint and their derivatives.

$$\begin{aligned} \mathbf{e} &= \left(\mathbf{r}^{(3)} - \mathbf{r}_d^{(3)} \right) + \lambda_1 (\ddot{\mathbf{r}} - \ddot{\mathbf{r}}_d) \\ &\quad + \lambda_2 (\dot{\mathbf{r}} - \dot{\mathbf{r}}_d) + \lambda_3 (\mathbf{r} - \mathbf{r}_d) \\ &= \mathbf{r}^{(3)} - \mathbf{r}_r^{(3)} \end{aligned} \quad (8)$$

Note that for the lateral controller, we assume that thrust is approximately constant, that is $\dot{\Gamma}$ is neglected. This is a reasonable assumption given that, apart from the initial takeoff period, the thrust rarely varies by more than 5%. The third derivative of \mathbf{r} then becomes

$$\mathbf{r}^{(3)} \approx \Gamma \dot{\hat{\mathbf{z}}} = \Gamma (-\omega_x \hat{y} + \omega_y \hat{x}). \quad (9)$$

We propose the following sliding surface \mathbf{S}_τ and the Lyapunov function candidate V_τ :

$$\begin{aligned} \mathbf{S}_\tau &= \left[-\mathbf{e} \cdot \hat{y} / \Gamma \quad \mathbf{e} \cdot \hat{x} / \Gamma \quad \omega_z \right]^T \\ &= \begin{bmatrix} \omega_x + \Gamma^{-1} \left(\mathbf{r}_r^{(3)} \cdot \hat{y} \right) \\ \omega_y - \Gamma^{-1} \left(\mathbf{r}_r^{(3)} \cdot \hat{x} \right) \\ \omega_z \end{bmatrix}, \end{aligned} \quad (10)$$

$$V_\tau = \frac{1}{2} \mathbf{S}_\tau^T \mathbf{J} \mathbf{S}_\tau. \quad (11)$$

Notice that angular velocity terms appear in (10), linking the attitude dynamics to the lateral dynamics. Using equations (1) and (10), we can write the derivative of the sliding surface as

$$\mathbf{J} \dot{\mathbf{S}}_\tau = \boldsymbol{\tau} - (\boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega}) - \Gamma^{-1} \mathbf{J} \frac{d}{dt} \begin{bmatrix} -\left(\mathbf{r}_r^{(3)} \cdot \hat{y} \right) \\ \left(\mathbf{r}_r^{(3)} \cdot \hat{x} \right) \\ 0 \end{bmatrix}.$$

This suggests the commanded body torque

$$\begin{aligned} \boldsymbol{\tau} &= -\Gamma^{-1} \begin{bmatrix} \mathbf{r}_r^{(3)} \cdot \hat{y} \\ -\mathbf{r}_r^{(3)} \cdot \hat{x} \\ 0 \end{bmatrix} \times \mathbf{J} \boldsymbol{\omega} \\ &\quad + \Gamma^{-1} \mathbf{J} \frac{d}{dt} \begin{bmatrix} -\left(\mathbf{r}_r^{(3)} \cdot \hat{y} \right) \\ \left(\mathbf{r}_r^{(3)} \cdot \hat{x} \right) \\ 0 \end{bmatrix} - K_\tau \mathbf{S}_\tau, \end{aligned} \quad (12)$$

so that the time derivative of the proposed Lyapunov function is negative definite and the system is proven stable:

$$\begin{aligned} \dot{V}_\tau &= -\mathbf{S}_\tau^T K_\tau \mathbf{S}_\tau - \mathbf{S}_\tau^T (\mathbf{S}_\tau \times \mathbf{J} \boldsymbol{\omega}) \\ &= -\mathbf{S}_\tau^T K_\tau \mathbf{S}_\tau \leq 0. \end{aligned}$$

Examining the control law in equation (12), it can be seen that the third derivative of \mathbf{r} could be written in terms of angular velocity as in equation (9). Therefore, only the second derivative of \mathbf{r} is required, alongside the body orientation and its first derivative. Furthermore, that fact that $\mathbf{r}_d^{(3)}$ is included in $\mathbf{r}_r^{(3)}$ implies that the controller effectively tracks a setpoint for the angular velocity and its derivative—the property lacking in previous work [10].

C. Adaptive Control

In [10], we identified six unknown parameters that were crucial to accomplish steady hover: the misalignment of the thrust vector from the \hat{z} axis (ϵ_x and ϵ_y), three unknown torque offsets ($\boldsymbol{\tau}_o = [\tau_{ox} \quad \tau_{oy} \quad \tau_{oz}]^T$), and the normalized thrust offset (T_o). In this section, we present how the proposed controller is adapted to take into account the affects of these unknowns. A predictor and an adaptive component are also implemented to ensure that these estimates of this unknowns converge to their true values and stability is still guaranteed in the Lyapunov analysis.

1) *Altitude control law:* For a small deviation of the thrust vector from the presumed robot \hat{z} axis, the thrust takes on small lateral components, resulting in a slight modification to equation (4).

$$\ddot{\mathbf{r}} = \Gamma (\dot{\hat{\mathbf{z}}} + \epsilon_x \hat{x} - \epsilon_y \hat{y}) - \mathbf{g} \quad (13)$$

Similarly, the thrust dynamics is modified to include the unknown offset by substituting T by $T_c - T_o$, $\dot{\Gamma} = \gamma (T_c - T_o - \Gamma)$, where T_c is the commanded thrust input. The derivative of the sliding surface defined in equation (6) becomes

$$\begin{aligned} \dot{S}_\Gamma &= (T_c - T_o - \Gamma) (R_{33} + \epsilon_x R_{31} - \epsilon_y R_{32}) \\ &\quad + \Gamma \epsilon_x (-R_{33}\omega_y + R_{32}\omega_z) \\ &\quad + \Gamma \epsilon_y (-R_{33}\omega_x + R_{31}\omega_z) - \frac{d}{dt} \ddot{Z}_r. \end{aligned}$$

Defining

$$\begin{aligned} \boldsymbol{\mu} &= \Gamma \hat{\epsilon}_x (-R_{33}\omega_y + R_{32}\omega_z) + \Gamma \hat{\epsilon}_y (-R_{33}\omega_x + R_{31}\omega_z) \\ &\quad + \Gamma R_{31}\omega_y - \Gamma R_{32}\omega_x - d\ddot{Z}_r/dt, \end{aligned}$$

the following control signal

$$T_c = \hat{T}_o + \Gamma - \Gamma^{-1} R_{33}^{-1} (1 - \hat{\epsilon}_x R_{31} R_{33}^{-1} + \hat{\epsilon}_y R_{32} R_{33}^{-1}) \times (\mu + K_\Gamma S_\Gamma) \quad (14)$$

makes the derivative of the sliding surface expressible as

$$\begin{aligned} \dot{S}_\Gamma &= \begin{bmatrix} \gamma (R_{33} + R_{31} \hat{\epsilon}_x - R_{32} \hat{\epsilon}_y) \\ \Gamma R_{33} \omega_y - \Gamma R_{32} \omega_z - R_{31} R_{33}^{-1} \mu \\ \Gamma R_{33} \omega_x - \Gamma R_{31} \omega_z - R_{32} R_{33}^{-1} \mu \end{bmatrix}^T \begin{bmatrix} \tilde{T}_o \\ \tilde{\epsilon}_x \\ \tilde{\epsilon}_y \end{bmatrix} \\ &\quad - K_\Gamma S_\Gamma + \gamma \tilde{T}_o (-R_{31} \tilde{\epsilon}_x + R_{32} \tilde{\epsilon}_y) \\ &= Y_\Gamma \tilde{\mathbf{a}} - K_\Gamma S_\Gamma + \gamma \tilde{T}_o (-R_{31} \tilde{\epsilon}_x + R_{32} \tilde{\epsilon}_y), \end{aligned} \quad (15)$$

where we have defined \mathbf{a} as a vector consisting of the three unknown parameters and Y_Γ accordingly. The first two terms in equation (15) are the typical form that usually appears in the derivation of adaptive sliding mode controller [18], [10]. The last term is handled explicitly in the last paragraph of section III-C.4.

2) *Lateral control law:* Including the effect of ϵ_x and ϵ_y , we define an estimate of $\mathbf{r}^{(3)}$ based on the estimates of ϵ_x and ϵ_y :

$$\hat{\mathbf{r}}^{(3)} = \Gamma \left(\dot{\hat{z}} + \hat{\epsilon}_x \dot{\hat{x}} - \hat{\epsilon}_y \dot{\hat{y}} \right). \quad (16)$$

It follows that the estimate of \mathbf{e} from equation (8) can also be written as $\hat{\mathbf{e}} = \hat{\mathbf{r}}^{(3)} - \mathbf{r}_r^{(3)}$, where no change has been made to $\mathbf{r}_r^{(3)}$. As a consequence, the sliding surface of the lateral position controller is re-defined:

$$\begin{aligned} \hat{S}_\tau &= \begin{bmatrix} -\hat{\mathbf{e}} \cdot \hat{\mathbf{y}} / \Gamma & \hat{\mathbf{e}} \cdot \hat{\mathbf{x}} / \Gamma & \omega_z \end{bmatrix} \\ &= \begin{bmatrix} \omega_x - \hat{\epsilon}_x \omega_z + \Gamma^{-1} \left(\mathbf{r}_r^{(3)} \cdot \hat{\mathbf{y}} \right) \\ \omega_y + \hat{\epsilon}_y \omega_z - \Gamma^{-1} \left(\mathbf{r}_r^{(3)} \cdot \hat{\mathbf{x}} \right) \\ \omega_z \end{bmatrix}. \end{aligned} \quad (17)$$

Using the definition in equation(8), we define

$$\begin{aligned} \dot{\hat{\mathbf{r}}}_r^{(3)} &= \mathbf{r}_d^{(4)} - \lambda \left(\hat{\mathbf{r}}^{(3)} - \mathbf{r}_d^{(3)} \right) - \lambda_2 (\ddot{\mathbf{r}} - \ddot{\mathbf{r}}_d) - \lambda_3 (\dot{\mathbf{r}} - \dot{\mathbf{r}}_d) \\ &= \dot{\mathbf{r}}_r^{(3)} + \lambda_1 \left(\hat{\mathbf{r}}^{(3)} - \mathbf{r}^{(3)} \right), \end{aligned} \quad (18)$$

such that equations (16) and(18) give

$$\begin{aligned} \dot{\hat{\mathbf{r}}}_r^{(3)} \cdot \hat{\mathbf{y}} &= \dot{\hat{\mathbf{r}}}_r^{(3)} \cdot \hat{\mathbf{y}} - \lambda_1 \Gamma \omega_z \tilde{\epsilon}_x \\ \dot{\hat{\mathbf{r}}}_r^{(3)} \cdot \hat{\mathbf{x}} &= \dot{\hat{\mathbf{r}}}_r^{(3)} \cdot \hat{\mathbf{x}} - \lambda_1 \Gamma \omega_z \tilde{\epsilon}_y. \end{aligned} \quad (19)$$

Expressing the body torque as the commanded torque τ_c and the unknown offset τ_o , $\tau = \tau_c + \tau_o$, we propose the following control law:

$$\begin{aligned} \tau_c &= \hat{\tau}_o - \begin{bmatrix} \Gamma^{-1} \mathbf{r}_r^{(3)} \cdot \hat{\mathbf{y}} - \omega_z \hat{\epsilon}_x \\ -\Gamma^{-1} \mathbf{r}_r^{(3)} \cdot \hat{\mathbf{x}} + \omega_z \hat{\epsilon}_y \\ 0 \end{bmatrix} \times J \omega \\ &\quad + \Gamma^{-1} J \begin{bmatrix} -\left(\mathbf{r}_r^{(3)} \cdot \dot{\hat{\mathbf{y}}} + \dot{\hat{\mathbf{r}}}_r^{(3)} \cdot \hat{\mathbf{y}} \right) \\ \left(\mathbf{r}_r^{(3)} \cdot \dot{\hat{\mathbf{x}}} + \dot{\hat{\mathbf{r}}}_r^{(3)} \cdot \hat{\mathbf{x}} \right) \\ 0 \end{bmatrix} \\ &\quad + \begin{bmatrix} \hat{\epsilon}_x \dot{\omega}_z \\ -\hat{\epsilon}_y \dot{\omega}_z \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\hat{\epsilon}}_x \omega_z \\ -\dot{\hat{\epsilon}}_y \omega_z \\ 0 \end{bmatrix} - K_\tau \hat{S}_\tau. \end{aligned}$$

It can be shown that using the proposed control law and equations (11), (17), and (19) the time derivative of the re-defined sliding surface is

$$J \dot{\hat{S}}_\tau = -K_\tau \hat{S}_\tau + (\tilde{\tau}_o + Y_\tau \tilde{\mathbf{a}}), \quad (20)$$

where

$$Y_\Gamma = \begin{bmatrix} 0 & \lambda_1 J_x \omega_z & 0 \\ 0 & 0 & -\lambda_1 J_y \omega_z \\ 0 & 0 & 0 \end{bmatrix}.$$

Observe that this proposed commanded torque τ_c only contains measurable variables and the adaptive parameters ($\dot{\hat{\epsilon}}_x$ and $\dot{\hat{\epsilon}}_y$) that will be given in section III-C.4.

3) *Predictor:* Prior to presenting the adaptive algorithm, we first design a predictor. The idea is that some parameter errors are reflected in prediction errors. This information could be used in conjunction with the regular tracking error to estimate the unknown parameters. This strategy is generally known as composite adaptation [18]. In our case, the predictor also has a vital role in the stability property of the Lyapunov function candidate shown in the next section.

First, let s be a Laplace variable, we define a first-order lowpass filter function $f_\gamma(\cdot) = \gamma(s + \gamma)^{-1}$. The generated thrust, therefore, can be written in the form

$$\Gamma = f_\gamma(T_c - T_o) = T_f - T_o,$$

where $\Gamma_f = f_\gamma(T_c)$. The translational dynamics of the robot in equation (13) then becomes

$$\begin{aligned} \ddot{\mathbf{r}} &= \mathbf{g} + (T_f - T_o) (\hat{z} + \epsilon_x \hat{x} - \epsilon_y \hat{y}) \\ \ddot{\mathbf{r}} - T_f \hat{z} - \mathbf{g} &= \epsilon_x T_f \hat{x} - \epsilon_y T_f \hat{y} - T_o \hat{z} + \mathcal{O}(\epsilon T_o) \end{aligned} \quad (21)$$

By neglecting the second-order effects, we can apply the lowpass filter throughout twice and express the quantity on the right hand side of equation (21) in vector form as

$$\begin{aligned} f_\gamma^2(\ddot{\mathbf{r}} - T_f \hat{z} - \mathbf{g}) &\approx \begin{bmatrix} -f_\gamma^2(\hat{z}^T) \\ f_\gamma^2(T_f \hat{x}^T) \\ -f_\gamma^2(T_f \hat{y}^T) \end{bmatrix}^T \mathbf{a} \\ &= W \mathbf{a}. \end{aligned} \quad (22)$$

At this point, we can substitute the vector \mathbf{a} by its estimate $\hat{\mathbf{a}}$ and the estimation error $\tilde{\mathbf{a}}$ and rearrange the terms so that all measurable and known quantities are on the left hand side of the equation and call it ε ,

$$\begin{aligned} f_\gamma^2(\ddot{\mathbf{r}} - T_f \hat{z} - \mathbf{g}) + W \hat{\mathbf{a}} &= W \tilde{\mathbf{a}} \\ \varepsilon &= W \tilde{\mathbf{a}}. \end{aligned} \quad (23)$$

Note that the matrix W is also measurable in real time.

4) *Adaptive implementation:* Here we propose a single Lyapunov function candidate for control of the flapping-wing MAV:

$$V = \frac{1}{2} S_\Gamma^2 + \frac{1}{2} \hat{S}_\tau J \hat{S}_\tau + \frac{1}{2} T \tilde{\mathbf{a}}^T \Upsilon^{-1} \tilde{\mathbf{a}} + \frac{1}{2} \tilde{\tau}_o^T \Psi^{-1} \tilde{\tau}_o. \quad (24)$$

The first two terms in equation (24) correspond to altitude control and lateral position control, while the latter two terms are penalty terms for errors in the estimation of unknown

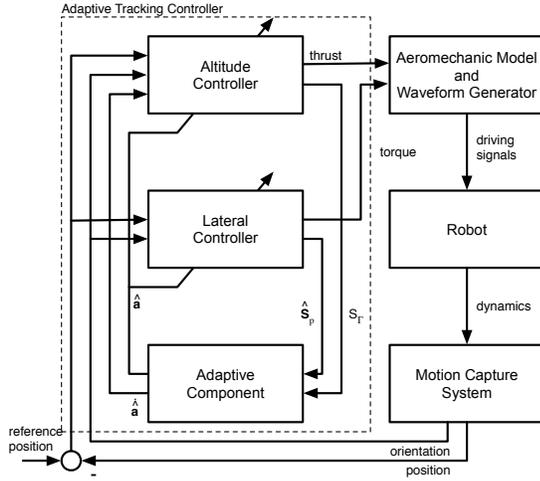


Fig. 5. A block diagram showing the details of the adaptive controller and complete feedback loop.

parameters. Υ and Ψ are positive diagonal matrices acting as adaptive gains.

Using the control laws presented in the preceding sections, the derivative of the Lyapunov function candidate is obtained by substituting in the results from equation (15) and (20),

$$\begin{aligned} \dot{V} = & -K_\Gamma S_\Gamma^2 - \hat{S}_\tau^T K_\tau \hat{S}_\tau + S_\Gamma Y_\Gamma \tilde{\mathbf{a}} + \hat{S}_\tau^T (\tilde{\tau}_o + Y_\tau \tilde{\mathbf{a}}) \\ & + \gamma \tilde{T}_o (-R_{31} \tilde{\epsilon}_x + R_{32} \tilde{\epsilon}_y) \\ & + \dot{\hat{\mathbf{a}}}^T \Upsilon^{-1} \tilde{\mathbf{a}} + \dot{\hat{\tau}}_o^T \Psi^{-1} \tilde{\tau}_o. \end{aligned} \quad (25)$$

Hence, we obtain the adaptive law for the unknown torque offset,

$$\dot{\hat{\tau}}_o = -\Psi \hat{S}_\tau.$$

For the estimation of \mathbf{a} , we propose the following adaptive algorithm:

$$\begin{aligned} \dot{\hat{\mathbf{a}}} = & -\Upsilon \left(Y_\Gamma S_\Gamma + Y_\tau \hat{S}_\tau \right) \\ & -\Upsilon (\Delta + \Sigma) (W^T W)^{-1} W^T \epsilon, \end{aligned} \quad (26)$$

where Δ is a positive diagonal matrix and Σ is a matrix with zero diagonal elements:

$$\Sigma = \frac{\gamma}{2} \begin{bmatrix} 0 & -R_{31} & 0 \\ -R_{31} & 0 & R_{32} \\ 0 & R_{32} & 0 \end{bmatrix}. \quad (27)$$

This incorporation of Σ enables us to cancel out terms that are the product of two parameter errors. Invertibility of $W^T W$ depends on the rank condition of W . From equation (22), it can be seen that W always has full rank before filtered. This might no longer be true after filtered. However, physically W is unlikely to be ill-conditioned. In practice $W^T W$ was always found to be invertible. Lastly, substitution of equations (23), (26), and (27) into (25) yields

$$\dot{V} = -K_\Gamma S_\Gamma^2 - \hat{S}_\tau^T K_\tau \hat{S}_\tau - \tilde{\mathbf{a}}^T \Delta \tilde{\mathbf{a}} \leq 0, \quad (28)$$

that is, the derivative of the Lyapunov function candidate is negative definite. To finalize the stability proof, the invariant

set theorem is applied. The value of V keeps diminishing as long as S_Γ , \hat{S}_τ , and $\tilde{\mathbf{a}}$ are not all zeros. The fact that \hat{S}_τ approaches zero does not immediately imply that the lateral dynamics would be stabilized since when \hat{S}_τ was defined in equation (17), it includes $\hat{\epsilon}_x$ and $\hat{\epsilon}_y$ rather than their true values. It is the inclusion of information from the predictor that results in the last term of equation (28) which ensures that the parameter estimates converge to their true values, and hence \hat{S}_τ eventually approaches S_τ and lateral stability is satisfied along with altitude stability.

IV. TRAJECTORY GENERATION

In smooth trajectory planning of robot manipulators, it is common to minimize the average squared jerk along the trajectory considered. This is in accordance with findings in psychophysical experiments on human arm movements [19]. In path planning of non-holonomic MAVs similar to our robot or quadrotors, minimum squared snap is preferred as the fourth order derivative in position can be related to torque. Hence, the optimization becomes the problem of minimizing some function of effort or torque inputs [16].

In this paper, we devised an algorithm similar to those found in [16], [20]. The optimization routine computes a smooth polynomial trajectory from specified waypoints by minimizing feedforward torque inputs while keeping the first four order derivatives of the position at the starting and ending points zero. The derivatives of the position at intermediate points are left unconstrained. The generated trajectories appear similar to those generated by minimizing the average squared snap along the trajectories.

V. EXPERIMENTS

A. Experimental Setup

Without onboard sensors and control, flight control experiments are carried out in a flight arena equipped with eight motion capture *VICON* cameras. These cameras provide real-time position and orientation feedback by tracking retroreflective markers attached to the robot at the rate of 500Hz, covering a tracking volume of $0.3 \times 0.3 \times 0.3$ m.

Computation is performed by external computers running an xPC target (*MathWorks*) environment. The control algorithm is implemented at the rate of 10kHz for both input sampling and output signal generation. Drive signals are delivered to the robot via a bundle of four 51-gauge copper wires after passing through a digital-to-analog converter and a high voltage amplifier.

The closed-loop latency of the experimental setup was found to be 12ms, sufficiently small to experimentally achieve stable flights [6], [10]. The effect of a wire tether is not taken into consideration due to its unpredictable nature. However, simple calculations suggest its contribution should not affect the flight dynamics significantly.

Without direct measurements, velocity, acceleration, and angular velocity are not available. In this paper, they are estimated from the use of filtered derivatives, resulting in a slight inevitable delay to these measurements.

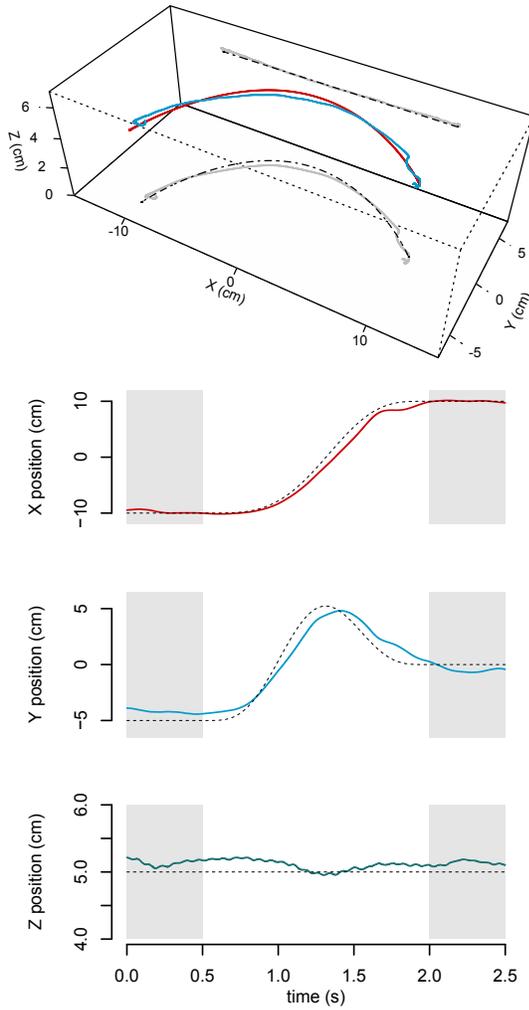


Fig. 6. (top) A 3D reconstruction of the trajectory from a 1.5-second trajectory following flight. The red line indicates a desired path and the blue line is the actual trajectory. Grey lines and dashed lines are projections of the colored lines. (bottom) Position-vs-time plots of the reference trajectory and flight path from the top figure along three axes.

Prior to performing unconstrained flight experiments, the robot prototype underwent a characterization process. This began with visual inspection of the flapping amplitude at various frequencies to identify a suitable operating frequency, where flapping trajectories of both wings are large and most symmetrical. Next, several open-loop takeoff flights—or *trimming* flights—were carried out in the flight arena. This process allowed us to preliminarily identify inherent torque offsets. These torque offsets vary considerably from robot to robot as a result of manufacturing imperfections. Information obtained from these steps is sufficient to perform closed-loop experiments.

B. Hovering Flight

To verify that the proposed controller is capable of stabilizing the robot, we first command the robot to hover at a stationary setpoint. After a several 5-second flights, the adaptive components tuned the parameter estimates sufficiently close

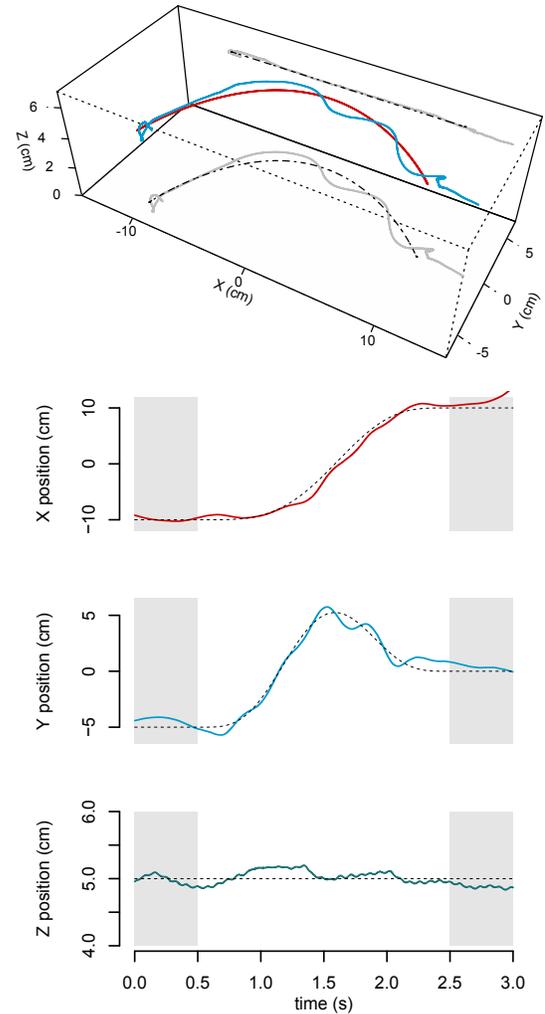


Fig. 7. (top) A 3D reconstruction of the trajectory from a 2.0-second trajectory following flight. The red line indicates a desired path and the blue line is the actual trajectory. Grey lines and dashed lines are projections of the colored lines. (bottom) Position-vs-time plots of the reference trajectory and flight path from the top figure along three axes.

to their true values. In the absence of mechanical fatigue, the robot regularly stayed close to the setpoint with a position error smaller than one body length. The *Root Mean Square* (RMS) errors in position for example hovering flights can be found in Table I. This confirms that the performance of the proposed controller is comparable to the previous adaptive controller in [10]. This is consistent with hovering flights which the setpoint is stationary. Theoretically, the robot is able to stay aloft indefinitely without crashing. In practice, we attempt to minimize the total operating time to prevent mechanical fatigue.

C. Trajectory Following

To demonstrate the tracking ability of the proposed single-loop controller, we demonstrate our flapping-wing robot performing smooth trajectories generated by the algorithm described in section IV. The aim here is to inspect the robot following similar trajectories at various speed.

TABLE I
COMPARISON OF THE RMS POSITION ERRORS FROM HOVERING
FLIGHTS AND TRAJECTORY FOLLOWING FLIGHTS.

Trajectory	RMS errors (cm)		
	X	Y	Z
hovering flight 1	0.72	0.70	0.15
hovering flight 2	1.25	0.57	0.12
1.0-second trajectory	1.24	1.01	0.16
1.5-second trajectory	0.84	0.82	0.13
2.0-second trajectory	0.93	0.57	0.10

The trajectories in the following experiment were generated from three setpoints. The robot was set to initially hover at the starting position, navigate to the middle point at specified times, and come to stop at the final waypoint in 1.0s, 1.5s and 2.0s. First to fourth order derivatives of the position at the starting point and ending point were set to zero as detailed in the previous section.

Figures 6 and 7 show the reference trajectories and recorded paths of the robot flying through the waypoints in 1.5s and 2.0s respectively. The RMS position errors for these flights, along with the 1.0-second flight, are also listed in Table I. The errors listed are the average from the maneuver interval and 0.5s before and after. It can be seen that in these cases, the robot was able to follow the predefined path with no evident increase in position errors compared with hovering flights. The exception is observed in the 1.0-second flight, of which the RMS errors are slightly larger than the rest.

VI. DISCUSSION AND FUTURE WORK

Towards the goal of aggressive maneuvers such as perching or acrobatic movements as demonstrated by other MAVs [11], [21], [22], we developed a single-loop controller that enables an insect-scale flapping-wing robot to follow dynamic trajectories with small errors. The single-loop design eliminates a few assumptions required in the previous controller [10]. We demonstrated that the approach brought about improved trajectory following while retaining an adaptive ability without compromising stability.

However, it is noticeable that position errors rose as flight time was reduced. This is unsurprising as additional aerodynamic effects were not incorporated into the dynamic model due to the lack of simple and accurate model. It is conceivable that an improved dynamic model could contribute towards the goal of achieving more aggressive movements. Alternatively, iterative learning techniques as illustrated in [11], [22] should also allow the robot to iteratively adapt the model based on information obtained from previous flights and eventually succeed in realizing pre-calculated trajectories with relatively small errors.

REFERENCES

[1] S. P. Sane and M. H. Dickinson, "The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight," *Journal of Experimental Biology*, vol. 205, no. 8, pp. 1087–1096, 2002.

[2] C. T. Orlowski and A. R. Girard, "Dynamics, stability, and control analyses of flapping wing micro-air vehicles," *Progress in Aerospace Sciences*, vol. 51, pp. 18–30, 2012.

[3] L. Ristroph, A. J. Bergou, G. J. Berman, J. Guckenheimer, Z. J. Wang, and I. Cohen, "Dynamics, control, and stabilization of turning flight in fruit flies," in *Natural locomotion in fluids and on surfaces*. Springer, 2012, pp. 83–99.

[4] D. Lentink, S. R. Jongerius, and N. L. Bradshaw, "The scalable design of flapping micro-air vehicles inspired by insect flight," in *Flying Insects and Robots*. Springer, 2010, pp. 185–205.

[5] T. N. Pornsin-Sirirak, Y.-C. Tai, C.-M. Ho, and M. Keennon, "Microbat: A palm-sized electrically powered ornithopter," in *Proceedings of NASA/JPL Workshop on Biomimetic Robotics*, 2001, pp. 14–17.

[6] 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, 2013.

[7] P. S. Sreetharan, J. P. Whitney, M. D. Strauss, and R. J. Wood, "Monolithic fabrication of millimeter-scale machines," *Journal of Micromechanics and Microengineering*, vol. 22, no. 5, p. 055027, 2012.

[8] R. Wood, E. Steltz, and R. Fearing, "Optimal energy density piezoelectric bending actuators," *Sensors and Actuators A: Physical*, vol. 119, no. 2, pp. 476–488, 2005.

[9] K. Y. Ma, S. M. Felton, and R. J. Wood, "Design, fabrication, and modeling of the split actuator microrobotic bee," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1133–1140.

[10] P. Chirarattananon, K. Y. Ma, and R. J. Wood, "Adaptive control for takeoff, hovering, and landing of a robotic fly," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, to appear.

[11] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[12] M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Inversion based direct position control and trajectory following for micro aerial vehicles," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2933–2939.

[13] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. of the AIAA Guidance, Navigation, and Control Conference*, vol. 2, 2007.

[14] J. Whitney, P. Sreetharan, K. Ma, and R. Wood, "Pop-up book MEMS," *Journal of Micromechanics and Microengineering*, vol. 21, no. 11, p. 115021, 2011.

[15] B. M. Finio, N. O. Pérez-Arancibia, and R. J. Wood, "System identification and linear time-invariant modeling of an insect-sized flapping-wing micro air vehicle," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1107–1114.

[16] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[17] W. Shyy, H. Aono, S. K. Chimakurthi, P. Trizila, C.-K. Kang, C. E. Cesnik, and H. Liu, "Recent progress in flapping wing aerodynamics and aeroelasticity," *Progress in Aerospace Sciences*, vol. 46, no. 7, pp. 284–327, 2010.

[18] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall New Jersey, 1991, vol. 199, no. 1.

[19] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[20] M. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "Path planning for motion dependent state estimation on micro aerial vehicles," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

[21] A. L. Desbiens, A. T. Asbeck, and M. R. Cutkosky, "Landing, perching and taking off from vertical surfaces," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 355–370, 2011.

[22] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadcopter multi-flips," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1642–1648.