Advancing Actuation and Control in Flapping-Wing Robots

Daksh Dhingra

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:

Sawyer B. Fuller, Chair

I.Y. Shen

Krithika Manohar

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Advancing Actuation and Control in
Flapping-Wing Robots

Daksh Dhingra

Chair of the Supervisory Committee:
Sawyer B. Fuller
Mechanical Engineering Department

Interest in flying insect robots (FIR) can be attributed to the potential benefits of such platforms. From search and rescue operations, farming automation, and military applications to space exploration these robots have ample use cases. These robots draw inspiration from their natural counterparts. The main goal of this thesis is to develop FIRs with improved agility, control, and maneuverability. Key contributions include automated trimming, precise voltage-to-torque mapping, novel drag-based longitudinal actuation, first principle model identification, model-based control, and discrepancy modeling of system dynamics. These advancements aim to revolutionize micro-aerial robotics and eventually enable applications in search and rescue, environmental exploration, and disaster response.

FIRs are hand-assembled, leading to manufacturing inconsistencies. This results in two major issues, (1) initial rapid rotation and tumbling instead of straight takeoff due to inherent manufacturing torques, and (2) unknown mapping between input voltage and generated torque, causing inconsistent control responses. Correcting for inherent torques, typically around 1μNm, is challenging due to the lack of precise sensors. Current methods are error-prone and time-consuming. This research introduces a torque measurement device using feedback from motion capture cameras, automating the correction process without user intervention or damaging airborne flights. Validation on two robots showed improved takeoff stability in open-loop flights.

Manufacturing inconsistencies also contributes to varied voltage-to-torque mapping between

different robots. This complicates building a generalized controller for tasks like hovering. Using an updated torque measurement device, this research measured torque via static angular deflections. The obtained mapping was validated and led to successful vertical takeoffs in open-loop flights.

This dissertation also addresses the problem of insufficient pitch torques generated with flapping-wing robots. Flying robots, including quadrotors, tilt their thrust vector forward or backward to achieve longitudinal motion. Flying insect robots relies on piezo-actuated flapping wings because of unfavorable downward size scaling in motor-driven propellers. The pitch torque for these robots is generated by moving the mid-stroke position of the wings, essentially by flapping the wings in the front or back of the body. This tilts their thrust vector to achieve longitudinal motion. Piezoelectric actuators, however have a limited bending capacity on the piezo layer limiting the maximum forward and backward movement of flapping wings. This, in turn, limits the extent to which the robot can flap its wing to the front or back of the body. This research proposes to use a paddling-type motion of the wings in combination with pitch torque to achieve higher agility in the longitudinal motion. In the paddling motion, the wings are flapped faster in the forward stroke than in the backward stroke to generate net backward drag force or vice versa. For the first time, the longitudinal movement from a stable hovering position is shown in free flight solely using paddling-type motion.

This dissertation also addresses modeling and optimal control of flapping-wing robots. Flying insect robots require a fast, responsive control system for sophisticated maneuvers. Traditional PID-type feedback control, with manually tuned gains, is insufficient. This research implemented the first optimal control demonstration on an FIR using a first principle model, enabling computationally efficient onboard control. Using an LQR, stable hovering and trajectory tracking were demonstrated on a 150 mg FIR, achieving translational velocities up to 25 cm/s.

This research was further continued to enable high-speed maneuvers on these robots. As high speed maneuvering is critical for efficient exploration but are often not possible on these robots due to complicated unsteady flapping wing aerodynamics, manufacturing uncertainties, and rapid wear. This research used discrepancy modeling via sparse identification of non linear dynamics (SINDy) to learn stroke-averaged translational models from free-flight data. Besides drag coefficients, we found that 20% of the discrepancy in the longitudinal direction is due to input pitch torque, and

30% of the discrepancy in the lateral direction is due to input roll torque. After compensating these discrepancies a model was developed, leading to a 49% increase in hovering precision and a 21% increase in trajectory tracking precision, with speeds up to 26.5 cm/s. This work can enable high-speed, time-optimal maneuvers for insect-sized flapping-wing robots.

The research presented in this thesis has the potential to make flying insect robots (FIRs) more practical for real-world use cases. Key advancements include enhanced maneuverability through a new actuation capability and improved model identification enabling fast trajectory tracking potentially leading to more efficient flights. Dynamic modeling and implementation of optimal control opens avenues for optimization under actuator constraints in different environmental conditions. Additionally, the development of a drag model and the establishment of a torque-to-translation speed relationship facilitate high-speed, precise control, making these robots suitable for navigating confined spaces in future.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

ACKNOWLEDGMENTS

The journey to earning a Ph.D. is often compared to a marathon rather than a 100 m. Throughout this long journey, many individuals have supported me, and without their help, neither this dissertation nor I would have reached this point.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Sawyer Fuller, for his invaluable guidance and mentorship throughout my Ph.D., which has aided my growth both as a researcher and as an individual. I would also like to thank my committee members, Dr. Steve Shen, Dr. Krithika Manohar, and Dr. Behçet Açıkmeşe. Working with Professor Shen and Professor Garbini as a teaching assistant made me a better instructor at the graduate level.

Additionally, I extend my special thanks to Dr. Yogesh Chukewad for recruiting me into the lab and for his mentorship on the projects included in my dissertation. My heartfelt appreciation goes to my friend and collaborator, Dr. Kadierdan Kaheman, for his insightful feedback, engaging research discussions, and unwavering support.

It has been a privilege to work with and learn from the numerous current and past members of the Autonomous Insect Robotics lab. I am particularly thankful to Johannes James, Aaron Weber, Yash Talwekar, and Avinash Singh. The research presented in this dissertation would not have been possible without the funding from the Mechanical Engineering Department at the University of Washington.

Lastly, I would like to acknowledge my personal support system. To my family and friends, thank you for your steadfast support as I pursued my dreams. I am profoundly grateful to all of you for encouraging me during the challenging times and celebrating with me during the joyous moments. Special thanks to my parents, Amrik Singh and Tripta, my sister, Priyanka, and my brother-in-law, Abhijit, for standing by my side through every high and low, and for providing the guidance I needed to navigate the challenges I faced along the way.

Chapter 1

INTRODUCTION

Research in insect-sized robotics has advanced significantly over the past decade, with contributions in fabrication, controls, power, and sensing. Starting with the first feedback-controlled flying insect robot (FIR) demonstrated in [74], various designs of similar size and weight have emerged, including Robobee Xwing [59], quad wing [48], Bee++ [7], robots powered by dielectric elastomer actuators [18, 17] and UW Robofly [30]. All the experiments presented in this thesis are performed on UW Robofly.

Fabrication of these robots has been improved by reducing part count [30], introducing prestacked actuators [60] that reduce variability in actuator performance, and standardizing the manufacturing process [29]. On the controls front, state-of-the-art controllers for FIRs primarily utilize adaptive PID flight control systems for hovering [24]. Maneuvering beyond hover, such as perching [52] and somersaulting [17], requires a sliding mode controller combined with iterative learning of trajectory parameters. Recent research has introduced optimal control strategies like modular Model Predictive Control (MPC) [32], which combines high-level MPC with a low-level controller for torque management. Additionally, data-driven MPC approaches like Tube-MPC [107] show promise for optimizing actuator constraints and trajectory tracking for complex maneuvers such as ramps and infinity loops.

There has also been significant work on studying the dynamics of flapping-wing robots. In [91], the effect of the relative position between the aerodynamic center and the center of gravity on intrinsic attitude stability of the FIR was studied, and [82] provided a better understanding of the dynamics of a flapping-wing mechanism. In [27], a nonlinear aerodynamics model of the robot with attached air dampers was identified. In [45], it was shown that a linearized time-invariant model is sufficient to predict the robot's behavior with reasonable accuracy over a large operating range.

1.1 Challenges Addressed in This Thesis

1.1.1 Manual Trimming

Trimming small flapping-wing robots like FIRs involves unique challenges:

- Fast Dynamics: As the size scale decreases, rotational acceleration rates increase, making the dynamics of small vehicles, such as FIRs, particularly fast and unstable [67, 20].

- Variability in Performance: FIRs' flapping mechanisms, due to their novelty, small size, and complexity, show greater variability in performance than slightly larger quad-rotor helicopters, necessitating larger compensatory trims. Without trimming, the first open-loop flights often result in rapid inversion and crashes. Our trimming device addresses both challenges.

1.1.2 Input Torque Mapping

Measuring torques generated by FIRs is difficult because the torques are too small for off-the-shelf sensor hardware. The smallest commercially available multi-axis torque sensor, the ATI Nano17 Titanium, has a resolution of 8 μNm, which is too high for accurate FIR torque measurements. Capacitive torque sensors have been used in the past but require expensive, specialized hardware and have not yet demonstrated two-axis measurement at a small scale [45].

1.1.3 Limited Actuation

Controlling FIRs is challenging due to their highly nonlinear dynamics, variability between robots, rapid wear and tear, and a high torque-to-moment of inertia ratio (approximately 10^3 rad/sec²). Natural fliers like moths and dragonflies use complex wing dynamics for precise maneuvers, but replicating this in 150 mg robotic systems is difficult. FIRs are highly underactuated, using only two piezoelectric actuators to control six degrees of freedom, and rely on passive wing hinges for energy efficiency and robustness, limiting direct control over wing angle of attack.

1.1.4 Model Identification and Optimal Control

Current control approaches [24] for FIRs require significant ad-hoc tuning and are task-specific [17, 52], failing to consider actuator, state, and environmental constraints. Advanced maneuvers like perching and flipping require parameter learning specific to each task. Model-based controllers have either not shown enough evidence for maneuvers beyond hovering [32] or are too computationally intensive [107] for sub-150 mg robots.

1.1.5 Achieving Time Optimal Flights

Power autonomy is crucial for making FIRs fully autonomous, but current research have resulted in power autonomous flying times of less than one second [62, 59]. As research in power autonomy progresses, designing precise controllers for high-speed maneuvers becomes essential. Increased flying speeds make aerodynamic effects more significant, and flapping wings at the insect scale exhibit highly nonlinear, unsteady aerodynamics, complicating time-optimal flight.

1.2 Contributions

While addressing the challenges mentioned above, the contrubutions in this thesis are listed as

- We developed a method to simultaneously map the input signal of the piezoelectric actuators with the output thrust and torque of the robot. This would ultimately be used for writing more accurate controllers.

- We bridge the gap between flapping wing robots and their biological counterparts by introducing an additional mode of actuation—longitudinal force.

- We developed and validated a first-principle model by comparing it with high-speed trajectory data collected from a sub-150 mg robot. Using that model we present LQR implementation of controlled flight on an FIR.

- Using discrepancy modeling we identified translational forces from the high speed free flight data. The translational forces represent aerodynamic model of the robot, affect of actuating torques, tether stiffness, and affect of translational speed on net thrust.

Chapter 2

TRIMMING

2.1 Rapid, Automated Trimming of Insect-Sized Flying Robots

For this work, we define the torque trimming process as finding a trim torque, τ_t, that exactly cancels the non-zero bias torque τ_b. A bias torque τ_b can arise from a) manufacturing inconsistencies, b) wear, and c) if there is damage to the FIR. The bias torque varies from robot to robot, as well as with changes in the payload carried by the robot, like a camera [6], boost converters for onboard power in [62], or an array of photovoltaic cells in [59], making trimming necessary for each robot. This study introduces a newly developed apparatus designed for the precise, rapid, and automatic trimming of flapping-wing robots. This purpose-built device employs closed-loop feedback to achieve the trimming process efficiently. As the wings flap, the apparatus utilizes real-time Euler angles data to continuously update the trim values, effectively countering bias torques. The primary goal is to minimize two crucial quantities: the pitch bias torque τ_b^{pitch} and the roll bias torque τ_b^{roll}. Through this precise trimming, the robot is capable of achieving near-vertical lift-off, placing the flying system's state within the stability basin of our closed-loop free-flight controller. The subsequent sections comprehensively describe the apparatus's design, operation, and testing, highlighting its effectiveness in achieving stable and controlled flight for the flapping-wing robot.

2.1.1 Trimming About an Elastic Rotation Axis

We begin by introducing the underlying physics of a flying robot attached to a trimming device of our design, whose basic form is shown in Fig. 2.2(b). The principle of operation relies on the fact that the device constrains the robot to rotate around a single axis while keeping all other degrees of freedom fixed. The axis is subject to a spring-like restoring torque so that the robot remains upright at equilibrium. In our device, there are two such axes—pitch and roll—so that both can be trimmed simultaneously. When the wings are flapped, they in general, produce both a thrust and a torque. Our device is designed so that applied torque, such as an undesirable bias torque τ_b, results

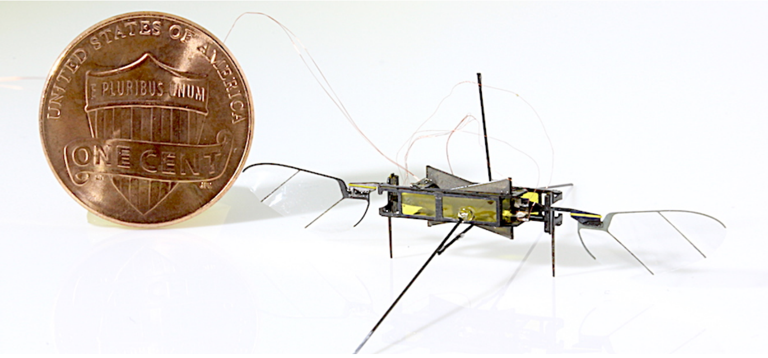


Figure 2.1: Robofly-expanded used in the experiments performed in this research (U.S. penny coin shown for scale).

in rotation in proportion to its magnitude. Whereas its design insures that thrust has no effect on rotation, regardless of magnitude. Trimming the robot then consists of determining the direction and magnitude of compensating trim torque τ_t that reduces rotation approximately to zero.

In our device, we use an integrating feedback control loop to find the compensatory trim torque τ_t. The dynamical equation for how the system's rotation angle evolves with time is:

$$J\ddot{\theta} = \tau_b - k_s\theta - c\dot{\theta} + \tau_t \tag{2.1}$$

Here J is the moment of inertia of the combined robot-trimming device system, θ is the rotation angle, τ_b and τ_t are the bias torque, and trim torque respectively, k_s is a spring-like stiffness constant (derived below), and c is a rotational damping coefficient due to aerodynamic drag forces on the flapping wings and damping in the flexure joints. We can solve for the angle at steady-state, giving

$$\theta = \frac{1}{k_s}\left(\tau_b + \tau_t\right). \tag{2.2}$$

This shows that the only way to make the angle zero is to make $\tau_b = \tau_t$. This provides a means to determine that the robot is trimmed, without the need to know the exact value of k_s. Practical design considerations limit the range of k_s, which we address below. If we can measure the angle θ of the robot in the trimming device, for example by using a motion capture camera or an on-board accelerometer, then we can use an integral controller to integrate this error in time to drive it to zero:

$$\tau_t = k_i \int_0^t \left(\theta_d - \theta\right) dt, \tag{2.3}$$

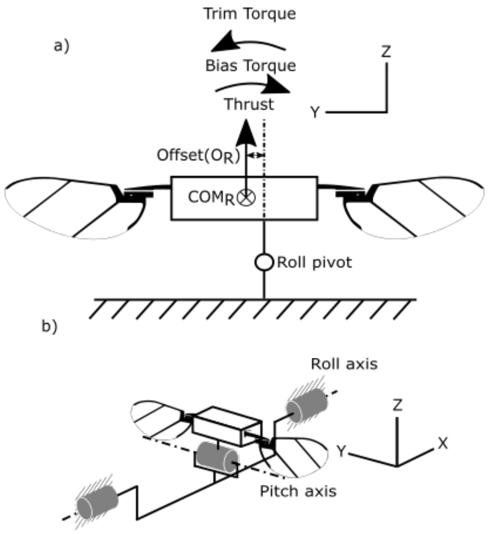


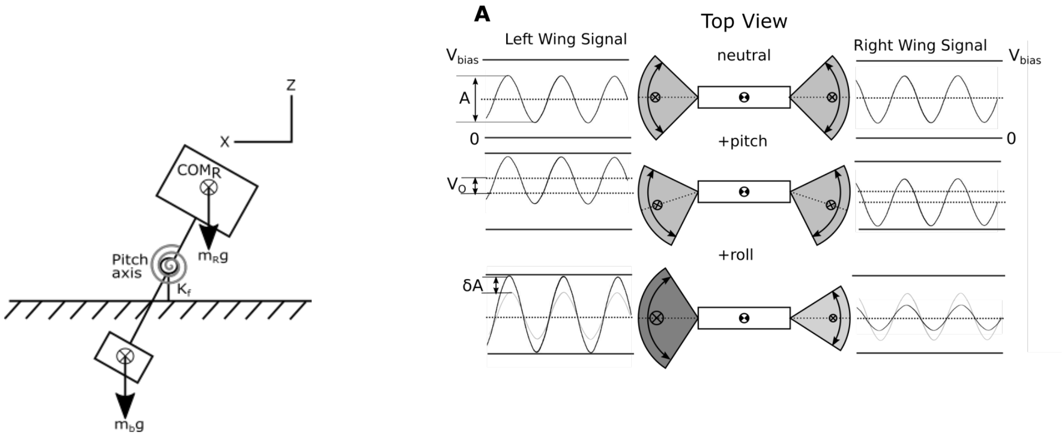
Figure 2.2: (a) A 2D representation of forces and torques acting on the robot attached to a single-axis version of the trimming device. The robot is constrained to rotate freely around a single axis; assuming O_R can be made small, the trim device rotates in response to a bias torque, which can be corrected by adding a compensatory trim torque. (b) A 3D representation of the rotation axes of the two-axis trimming device reported here, which allows trimming roll and pitch axes simultaneously.

where k_i is the integral controller constant and θ_d is the desired angle, which is zero in our case. This concept can be used for finding both roll and pitch trim torques.

2.1.2 Mechanisms of Torque Actuation

For torque trimming, we are not concerned with the exact mechanism by which a bias torque τ_b arises, but we must have the means to compensate for it using a trim torque τ_t. In both quad-rotor and flapping-wing platforms, pitch and roll torques can be actuated approximately independently. In a quad-rotor, if rotor speeds are changed, the resulting differential force between propellers causes a torque (Fig. 2.3-b). Roll torques in the FIRs are produced through a similar means, by increasing the thrust magnitude of one wing (T_1) relative to the other wing (T_2),

$$\tau_{roll} = (T_1 - T_2)b.$$



a

b

Figure 2.3: (a) Free-body diagram of a robot allowed to rotate around a single axis. A counterbalance m_b and torsional spring k_f due to flexure stiffness cause a restoring moment to keep the robot's mass m_R to rest at a near-vertical inclination at equilibrium. (b) A comparison of the mechanisms by which quad-rotor aircraft and the Robofly actuate pitch and roll torques. The centers of the aerodynamic thrust are denoted by a $\otimes$ with a size proportional to thrust magnitude

A pitch torque is produced by altering the position of the stroke-averaged center of aerodynamic thrust (Fig. 2.3-b),

$$\tau_{pitch} = (T_1 + T_2)d.$$

For a vehicle actuated by other means, such as four electrohydrodynamic thrusters [92], actuation is very similar to quad-rotor devices.

2.1.3 Trimming Device Design

We chose as our performance specification a maximum bias torque error of 0.3 μNm. This is motivated by the estimated torque uncertainty induced by the thin wire tether that provides power and control signals to the robot. While it is hard to provide a simple model of its effect due to its widely variable conformation, one reasonable model for the tether is a torsional spring. Experiments performed in [50] indicated that a 45° rotation causes a torque of approximately 0.3 μNm, which we set as our maximum error.

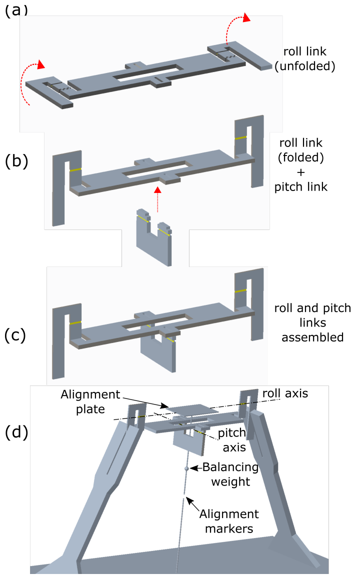


Figure 2.4: (a) The roll link of the trimming device. The link is folded around the castellated joint so the fly can be held perpendicular to the roll flexure and parallel to the ground. (b) The pitch link of the device. Its attached to the roll link via protrusions on top. (c) The whole assembly of trimming device, the fly is attached to the pitch link on the bottom. (d) Trimming device attached with the support columns, alignment fixtures and balancing weight.

We then used our performance target to design the size of the flexure joints. Flexures were made of 12 μm polyimide film ("kapton"). The bending stiffness of the flexure joint, is calculated by using $k_f = \frac{Et^3w}{12L}$ [77]. Here, t is the thickness of the flexure material, L is the length of the flexure joint (500 μm), w is the width of the joint (3 mm) and E is Young's modulus of 2.5 Gpa. At these values, the resulting flexure stiffness is 2.16 μNm/rad. The same value of flexure stiffness is used for both roll and pitch joints. With this stiffness, a bias torque of 0.3 μNm produces a rotation of approximately 8°. This is easily measured by the motion capture arena, which can measure angles to within approximately 1°, indicating that our device can provide the necessary accuracy. Note that we chose the counter-balance mass and position (Fig. 2.3a) so that $m_b l_b \approx m_R l_R$ so that their equivalent stiffness counteracted the "negative stiffness" caused by the robot's COM being above the flexure rotation axis. To mitigate buckling, all flexures were loaded in tension rather than compression. The remaining rigid structure was machined from inexpensive, rigid 254 μm fiberglass (FR4), which sandwiched the polyimide flexure material on both sides, following "smart composite microstructure" fabrication methods first introduced in [57]. A rigid part connects the two perpendicular pitch and roll flexures, which are designed so that their rotation axes

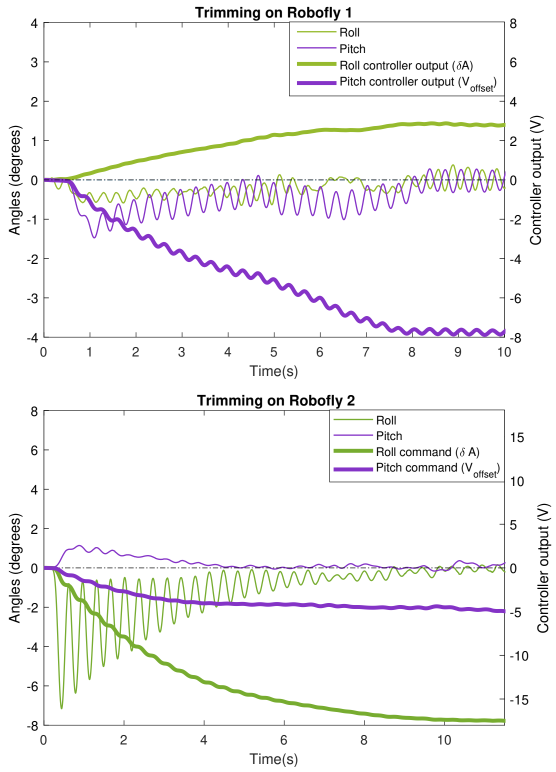


Figure 2.5: Results from trials on two different Robofly devices in the trimming device. Left axes show the angles of the two rotation axes as measured by motion capture, which are driven to zero by the integrating action of the feedback controller. Right axes show how the compensating bias voltages evolve in time as the trimming device settles on the correct trim values.

are as close as possible to the robot's COM (Fig. 2.4 (b)), up to 10 mm for the pitch axis. This can cause an error torque if the true thrust vector direction deviates from vertical, violating assumption 1. Adaptive controllers, e.g. in [74, 48], typically estimate this to be about 2°. This results in an error of approximately 0.3μNm, within the specification. The width of the roll link is minimized, at just 15 mm, so that it does not affect airflow from the wings.

2.1.4 *Trimming Controller*

The motion capture system (four Prime13 cameras, OptiTrack, Inc., Salem, OR) sends the orientation of the trimming device in quaternion form over Ethernet at 240 Hz. A computer running

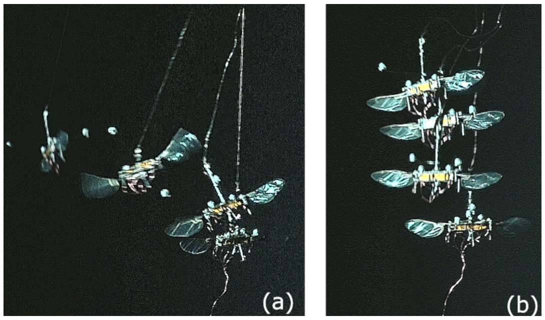


Figure 2.6: (a) Video-composite images taken from an un-trimmed open-loop takeoff of Robofly (offset voltage V_o and voltage differential δA at 0 V). (b) Open-loop of the same robot using trim values derived from the trimming device from the trimming procedure in Fig. 2.5(top). The robot lifts off nearly vertically, producing clearly-discernible slack in the kevlar restraining filament at the top. The vertical flight indicates that the vehicle has been properly trimmed.

Simulink Real-Time (Mathworks, Natick, MA) performs control computations and produces an analog voltage waveform that is amplified using piezo amplifiers (Trek model 2205). The control computer calculates Euler angles roll θ_x and pitch θ_y from the quaternion, with a convention matching those of the flexures in the trimming device. The displacement of the wings varies linearly with the voltage applied to the piezo actuators, with the voltage signal to the left and right wings being

$$\begin{aligned} V_l(t) &= (A+\delta A)\sin(\omega t)+V_o+V_b/2 \\ V_r(t) &= (A-\delta A)\sin(\omega t)+V_o+V_b/2, \end{aligned}$$

respectively. Thrust varies approximately linearly with flapping amplitude [88]. Here, A is the baseline voltage amplitude, ω is the flapping frequency, and V_b is the piezo bias voltage. Roll torque is varied by varying δA, the amplitude difference between left and right wings. Pitch torque is varied by varying V_o, the mean voltage of the sinusoid. To avoid actuator breakage, the controller includes saturation blocks to limit the range of these inputs [116].

2.1.5 Trimming Results

Voltage commands from the integral controller and output angles measured by motion capture for two different Robofly devices are shown in Fig. 2.5. In both cases, trim voltages appear to asymptotically reach steady-state values within approximately 10 s. The resulting trim values are a pitch offset voltage V_o=-8 V and amplitude differential δA=3 V for the first and V_o=-5 V, δA=-17 V for the second. These results show that the system can arrive at a calibration regardless of the polarity of the bias torque.

To establish whether trimming was successful, we compared open-loop flights before and after trimming. In both cases, the wings were driven with a flapping frequency of 150 Hz and an amplitude of 180 V. The device was constrained by a thin Kevlar thread to minimize crashes. For un-trimmed flights, $V_o = V_d = 0$. Using the trim voltages obtained from the trimming device, the robot can take off nearly vertically, indicating a properly-trimmed device (Fig. 2.6 (b)). Similarly, the second Robofly exhibited an equivalent transition from tumbling to vertical flight after trimming.

Chapter 3

MAPPING

The text of this chapter is a draft of a journal paper, which is under review. In this work, we reduce the complexity of controlling flapping-wing flying insect robots by introducing a high resolution device that can map control voltages to robot thrust, roll torques, and pitch torques. This is the first device capable of 3-axis simultaneous force-torque measurement for robots of this scale. Our findings indicate that the relationships between the control voltages and the output torques are linear and have minimal cross-axis coupling, and the upward thrust is not significantly impacted by roll and pitch control. These findings validate current control approaches and can be used to better model the dynamics of similar robots.

Weber, A., Dhingra, D. and Fuller, S. A flexured-gimbal 3-axis force-torque sensor reveals minimal cross-axis coupling in an insect-sized flapping-wing robot, RAL 2025 (Under review).

3.1 Abstract

The mechanical complexity of flapping wings, their unsteady aerodynamic flow, and challenge of making measurements at the scale of a sub-gram flapping-wing flying insect robot (FIR) make its behavior hard to predict. Knowing the precise mapping from voltage input to torque output, however, can be used to improve their mechanical and flight controller design. To address this challenge, we created a sensitive force-torque sensor based on a flexured gimbal that only requires a standard motion capture system or accelerometer for readout. Our device precisely and accurately measures pitch and roll torques simultaneously, as well as thrust, on a tethered flapping-wing FIR in response to changing voltage input signals. With it, we were able to measure cross-axis coupling of both torque and thrust input commands on a 180 mg FIR, the UW Robofly. We validated these measurements using free-flight experiments. Our results showed that roll and pitch have maximum cross-axis coupling errors of 8.58% and 17.24%, respectively, relative to the range of torque that is possible. Similarly, varying the pitch and roll commands resulted in up to a 5.78% deviation from

the commanded thrust, across the entire commanded torque range. Our system, the first to measure two torque axes simultaneously, shows that torque commands have a negligible cross-axis coupling on both torque and thrust.

3.2 Introduction

Flying insect-sized robots (FIRs) are sub-gram robots that use flapping wings inspired by insects. Their small size and low weight gives them an advantage in terms of the ability to access places that are otherwise inaccessible by bigger drones. For this reason, they have promising potential in applications like search and rescue missions, running inspections in manufacturing plants, and detecting gas leaks. Unlike birds and bats, bumblebees and other insects flap wings using a pair of thorax muscles. Inspired by biology, FIRs use a pair of piezoelectric actuators connected to the wings through a transmission system [88], [48], and [118]. In motors, friction forces and heat dissipation in coils increasingly dominate as the size gets smaller. Piezoelectric actuators can operate at high efficiency even at the centimeter scale.

Single-input single-output control at the actuator level is important to achieve high level precision control in flying robots. Mahony et. al. [76] developed a motor model that converts the input PWM signal to the rotor speed for their quadrotor systems. Karasek et. al. [65] used an electronic speed control with customized RPM sensing to achieve precise motor control on their 28g flapping-wing robot. However, modeling the output based on the input voltage in piezo-based systems is more challenging because: 1) piezoelectric actuators and the transmission systems used in the FIRs are manufactured and assembled by hand, resulting in greater manufacturing variability and more variable output thrust and output torque, and 2) FIR components are prone to high wear so even with the same input to the actuator, the output of a flapping wing changes over time. The flapping process puts high stress on the FIR and results in a short device lifespan [77], and can cause the dynamics of the FIR to change during experimentation.

Control of these robots [23] [48] [74] has previously relied on the robustness of feedback control systems to compensate for uncertainties in command-to-output mapping. As a practical matter, however, approaching control in this way leads to significant amounts of trial and error in experiments to tune the gains of the controller specific to a robot. This not only reduces operator productivity, but it can also severely reduce the lifespan of the robot [77].

The goal of this work is to develop a method to simultaneously map the input signal of the piezoelectric actuators with the output thrust and torque of the robot. This would ultimately be used for writing more accurate controllers for FIRs and aid in characterizing the performance of new designs.

Measuring torques at the scale output by FIRs is challenging because the small torques involved preclude using off-the-shelf sensor hardware. The smallest commercially available multi-axis torque sensor, the ATI Nano17 Titanium, has a resolution of 8 μNm. This is an order of magnitude higher than what is needed to accurately measure FIR torques. One alternative that has previously been used to analyze the torque of FIRs is a capacitive torque sensor [45]. The downside of such systems is that they require expensive, specialized capacitive sensing hardware costing $1000 or more. Furthermore, a two-axis version capable of measuring two torque axes at once has not yet been demonstrated at small scale. Here, we propose an alternative approach that, like capacitive and strain-gauge systems, allows the object to move by a small amount. In our system, however, the spring-like restoring torque is fairly low, so that the angular deflections are large enough to be detected by motion capture or inclinometer. By doing so, we are able to use hardware that is already available in many robotics contexts: either a camera-based motion capture system, or potentially an accelerometer. As a consequence, however, the bandwidth of our sensor is reduced, allowing it to only measure torques on a stroke-averaged basis. While we believe the torque or force changes nearly instantaneously in response to a voltage command, this has not been measured, and such dynamic measurements would require a different kind of sensor.

To perform the desired measurements, we introduce a system that is conceptually similar to the device introduced in [36], but that incorporates a number of improvements. These include flexure axes that now intersect the approximate center of mass of the vehicle, the addition of a damper to reduce unwanted oscillations, the addition of a precision scale to measure forces, and precise calibration and validation to ensure that it can measure torque outputs produced by the FIR.

This work is the first to perform simultaneous two axis torque mapping of FIRs. Until now, FIR controllers have assumed they have independent roll and pitch actuation, but this has never been measured directly. Our results show this assumption largely holds and demonstrate that torque actuation has a negligible impact on the upward thrust of the FIR.

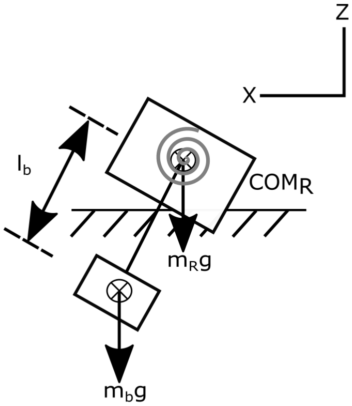


Figure 3.1: Principle of the torque measurement, showing the robot and the counterweight.

3.3 Principle of Operation

3.3.1 Torque measurement about flexure axis

In our experiments we use a device introduced in [36] that constrains the robot to rotate around two axes- pitch and roll- while keeping all other degrees of freedom fixed. The axis is subject to a spring-like restoring torque so that the robot remains upright at equilibrium. When the wings are flapped, in general they produce both a thrust and a torque. The device is designed so that applied torque can be measured by the angular deflection while ensuring that thrust has no effect on rotation, regardless of magnitude.

Sensitivity of the device is the angular deflection per unit torque applied to the system. We want the minimum measurable torque by the device to be 0.3 μNm. This is motivated by the estimated torque uncertainty induced by the thin wire tether that provides power and control signals to the robot. While it is hard to provide a simple model of its effect due to its widely variable conformation, one reasonable model for the tether is a torsional spring. Experiments performed in [50] show that a 45° rotation causes a torque of approximately 0.3μNm. Figure 3.1 shows the principle behind such a device. A robot of weight m_r and moment of inertia I mounted on a torsional spring of flexure stiffness K_f at a distance l_r from the axis of rotation. A counterweight of mass m_b is mounted at a distance of l_b from the axis of rotation. New to this version of the device, a damping rod extends down into a dish of glycerin with damping coefficient b to provide damping. Net torque about point

O is:

$$\tau_{net} = m_b l_b \sin\theta - m_R l_R \sin\theta + k_f \theta + b\dot{\theta} + I\ddot{\theta} \tag{3.1}$$

As the new version of the device makes the robot axis of rotation level with the device axis of rotation, the $m_R l_R \sin\theta$ term can safely be ignored. For our application, the measurements will be taken when the robot is at a steady state, and as such the $\dot{\theta}$ and $\ddot{\theta}$ terms will go to zero. Using the small-angle approximation,

$$k_s = m_b l_b + k_f \tag{3.2}$$

Where $k_s = \frac{\tau}{\theta}$ is the sensitivity of the device. Sensitivity can be increased by reducing the mass of the counterweight m_b. However, making the sensitivity too high could result in deflections that enter the non-linear zone of the elastic flexure joint.

3.3.2 Actuating Torques

The input to the FIR is a sinusoidal signal. The amplitude V_{amp} of the sinusoidal signal is proportional to the thrust magnitude of each wing. Roll torques in flapping-wing platforms are produced by increasing the amplitude of one wing relative to the other wing, $\Delta V = V_{amp1} - V_{amp2}$. A pitch torque is produced by altering the position of the stroke-averaged center of aerodynamic thrust [36]. This is achieved by changing the mean of the sinusoidal signal by an offset voltage, V_{off}. We assumed the torques produced by the robot would be linear with respect to the inputs, which was validated by our results.

While mounted on the device when a robot generates torque it will produce angular acceleration, $\ddot{\theta}$, angular velocity, $\dot{\theta}$, and angular deflection, θ. However, in steady state conditions we can assume $\dot{\theta}$ and $\ddot{\theta}$ terms to be zero.

3.4 Experimental Setup

3.4.1 Flexured-gimbal Device

We introduce a new design with a few improvements over the device introduced in [36]. Like its earlier incarnation, our system has two independent axes of rotation.

The first improvement is that our new system moves the axes of rotation of the roll and pitch degree of freedom so that they both now intersect the approximate location of the center of mass of the FIR (Figure 3.2.) In the earlier design, one of the flexure axes was well below the FIR. This improvement means that now, thrust force force from the wings that is not perfectly vertical, which could happen if the FIR is slightly tilted, no longer causes a torque disturbance in measurements. We made this design change by enlarging the circular gimbal shape so that it encircled the flapping wings with enough distance to avoid impeding flow.

Second, the new system adds a dish of glycerin below to attenuate undesirable resonant oscillations of the system.

And third, it is mounted on a precision balance so that forces can be measured simultaneously with torques.

The joints of the system are made of a flexible layer of 12 μm Kapton sandwiched between two 254 μm fiberglass (FR4) layers. Machining the larger gimbal shape required performing two separate cuts with the galvo-steered laser, with a precise stage move in between. This was because was too large to fit within the $\approx$ 50 mm cutting area of the galvo. The two fiberglass layers were then bonded with the Kapton layer using Pyralux adhesive sheets.

To fabricate the device, we aligned all the layers using tight-fit pins and pressed it under 50 kgf force and 200°C temperature to adhere them. The base joints were then mounted on a platform that positions the robot at a height of 65 mm to minimize the ground effect. Once the device is assembled, a stage is attached and set slightly below the flexure joints, so that when the FIR is mounted its roll and pitch axes of rotation will be level with the flexures of the device. Additionally, a thin rod is attached extending below the stage, which sits in a petri dish filled with glycerin for damping. The device can be seen in Figure 3.3.

3.4.2 UW Robofly

For the experiments we are using the FIR introduced in [29]. The robot weighs 180 mg (including motion capture markers) and flaps its wings at 180 beats per second.

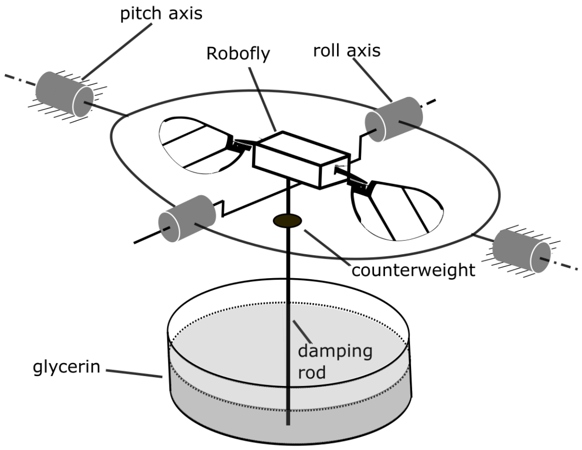


Figure 3.2: Diagram of the principle of operation of the flexure-based force/torque sensor. The flexures are positioned such that the roll and pitch axes of the device intersect, and intersect with the approximate center of mass of the FIR. The addition of a damping rod, below, whose end is immersed in glycerin, provides damping to eliminate oscillations.

3.4.3 Performing measurements

A crucial step before data collection is aligning the center of rotation of the robot with the axis of rotation of the flexures on the gimbal. As done in [36], this calibration procedure involved moving the robot laterally/longitudinally on the device until its pitch and roll angles are equal to what they were before the robot was added. Once the FIR has been mounted on the device, the whole gimballed system was placed on the scale so that thrust values may also be measured. The base of the device has been redesigned from the previous iteration to allow the base to sit on the scale while the FIR hangs out the side off of the measurement platform, to avoid measurement errors. During the testing period, a variety of control signals are sent to the FIR, with different values of V_{off} and ΔV, and the angle values are measured with the motion capture system and the angle data is taken from the steady state portion of each flight. This data can then be used to determine the torque-voltage mapping.

Two FIRs, the “mapping fly” and the “validation fly,” were used for experimental data collection. The mapping fly received various control signals for mapping analysis, but the stress from this

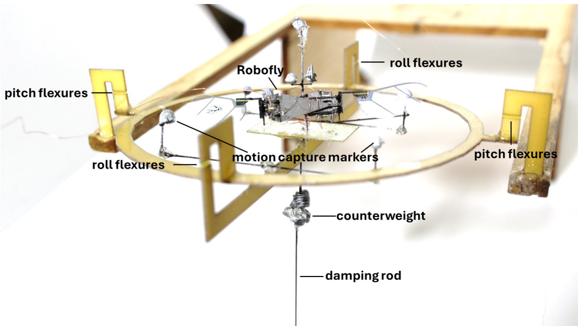


Figure 3.3: Image of the flexured-gimbal device with a FIR attached. Not shown in the figure is the glycerin petri dish below. Readout is accomplished in this case using a camera-based motion capture system.

process made it unusable for further testing. Consequently, the validation fly was used for a shorter mapping process. ”Ground truth” measurements were obtained on the validation fly by applying static free-flight torques with small offset weights on rods extending from the robot’s center and then trimming it in free flight. Using two separate robots with different actuator dynamics, trims, and manufacturing uncertainties demonstrates the generality of our results and shows uniformity and accuracy of our proposed method.

3.5 Results

3.5.1 Flexured-gimbal Device Calibration

To calculate the sensitivity of the flexure joints, we applied known torques about the axis of rotation and noted the resulting static angular deflections in the flexure joints. The torques are applied in both clockwise and counterclockwise directions by hanging known masses at some known distances from the axis of rotation. Figure 3.4 shows the data points of static angular deflections from applied torques about the roll and pitch axis. The slope of these lines is the sensitivity of the flexure joints. For our device, the sensitivity was found to be 1.518 μNm/rad in roll and 1.882 μNm/rad in pitch, using a least-squares fit.

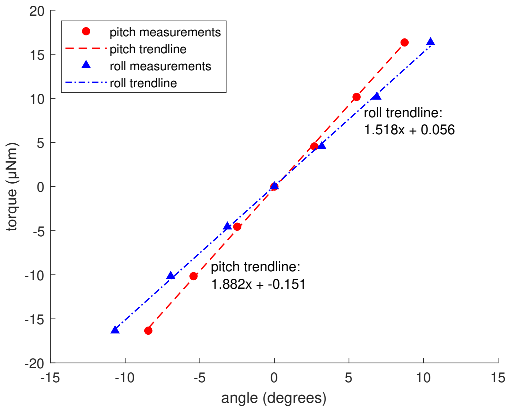


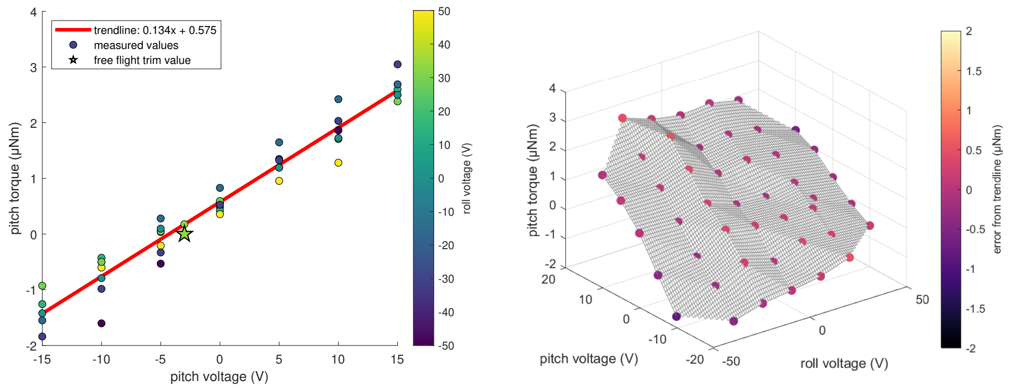
Figure 3.4: Flexured-gimbal device sensitivity measurements in the roll and pitch axes, with calculated trend lines.

3.5.2 Trimming Results

To establish a starting point for the range of torque mapping measurements, the FIRs were trimmed in free flight before being attached to the device to find the values of ΔV and V_{off} (as described in section II) that allow for a straight takeoff while canceling out undesired bias roll and pitch torques from manufacturing errors. The initial mapping fly successfully took off with the trimming values of 34 V roll trim (ΔV) and -3 V pitch trim (V_{off}). The validation fly successfully took off with the trimming values of -20 V roll trim and 7.5 V pitch trim. For the validation fly offset torques, a 31.8 mg mass was added on a rod 4 mm from the center along the roll axis, resulting in a 1.248 μNm roll torque, and the adjusted roll trim required for takeoff was -17 V. Similarly, a 25 mg mass was added on a rod 4mm from the center along the pitch axis, resulting in a -0.981 μNm pitch torque, and the adjusted pitch trim required for takeoff was 5 V. After the mapping measurements the validation fly was trimmed one final time, and took of with values of -23 V roll trim and 5 V pitch trim.

3.5.3 Torque Mapping

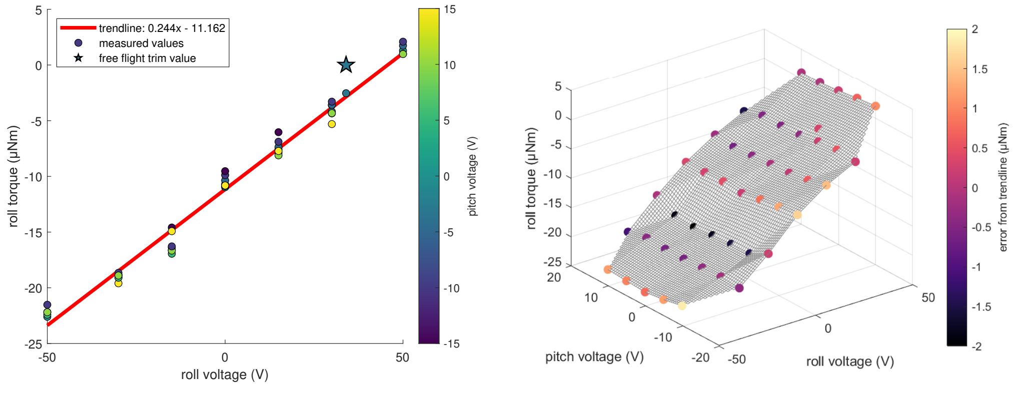
After trimming in free flight, the FIR was mounted on the device and angle measurements were taken at varying roll and pitch voltages. The pitch voltages used were 0 V, ±5 V, ±10 V, and ±15 V,



a b

Figure 3.5: (a). Mapping of the pitch voltage offsets in the control signal to the resulting pitch torque measured by the device, with a color map to show the strength of the roll voltages at each data point. (b). Mapping of the roll and pitch voltage offsets in the control signal to the resulting pitch torque measured by the device. Error from the mapping trendline is shown via the colormap at the measurement points. Pitch torque is not significantly impacted by changes in roll control voltage.

and the roll voltages used were 0 V, ±15 V, ±30 V, and ±50 V. All combinations of the listed roll and pitch voltages were tested, with the exception of ±15 V pitch and ±50 V roll, as these high roll and pitch values were hitting the actuator limits. These limits are set by the bias voltage of 250 V and the analog sinusoidal signal floats between 0 and 250. When the pitch and roll goes higher than ±15 and ±50 respectively, the flapping signal hits the limits of 0 and 250 V. Using the flexured-gimbal device and motion capture setup, the control signal was sent to the FIR on the device and its roll and pitch angles were measured. The control signal was sent for 3 seconds, and the measurements from the last 0.5 seconds were taken and averaged for use in the torque mapping, to allow the FIR to reach a steady angle on the device.



a

b

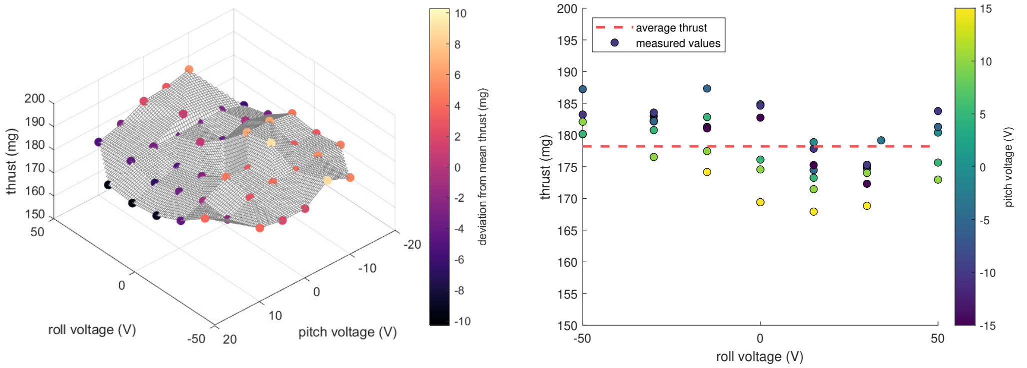
Figure 3.6: (a). Mapping of the roll voltage offsets in the control signal to the resulting roll torque measured by the device, with a color map to show the strength of the pitch voltages at each data point. (b). Mapping of the roll and pitch voltage offsets in the control signal to the resulting roll torque measured by the device. Error from the mapping trendline is shown via the colormap at the measurement points. Roll torque is not significantly impacted by changes in pitch control voltage.

Pitch measurements

Shown in Figure 3.5a is the mapping of the pitch control voltages to the resulting pitch torques calculated using the measured angles and the angle-to-torque mapping found in the flexured-gimbal device calibration section. Multiple measured values are shown at the different values of pitch voltages due to the same pitch voltage being used with multiple different roll voltages. The mapping can be fit to a linear trendline, which fits the measured data with a 0.95 coefficient of determination.

Roll measurements

Shown in Figure 3.6a is the mapping of the roll control voltages to the resulting roll torques, calculated in the same manner as the pitch mapping. As with pitch, multiple measured values at the same roll voltage are where the same roll voltage was used with different pitch voltages. As with the pitch mapping, the roll mapping can be fit to a linear trendline, which fits the measured data with a 0.98



a b

Figure 3.7: (a). Mapping of the roll and pitch voltage offsets in the control signal to the resulting FIR thrust, with a colormap indicating deviation from the mean thrust value. (b). Mapping of roll voltage to resulting FIR thrust with a colormap indicating the corresponding pitch voltages, showing that the thrust is not significantly impacted by the roll voltage but that there is a slight trend of lower thrust values corresponding to higher pitch voltages.

coefficient of determination.

Pitch and Roll Torque Coupling

In addition to mapping the roll and pitch control voltages to the resulting roll and pitch torques, we also wanted to measure if roll voltage had a significant effect on the pitch torque, and if pitch voltage had a significant effect on the roll torque. The results are shown in Figures 3.5b and 3.6b, with error from the trendline indicated by the color mapping. Larger error values are mostly on the edges of the plots where the control voltages are higher,likely due to these control voltages being at the upper limit of the signal range and not representative of typical flight conditions. The cross-axis coupling error was calculated as the offset between the measured torque and the expected torque from the trendline, relative to the total torque range actuated by the FIR. At the points with the most extreme control voltage in the opposite axis, the maximum error in the torque mapping is 2.12 μNm in the roll axis, which gives a percentage error of 8.58% relative to the total roll torque actuated, and

.84 μNm in the pitch axis, which gives a percentage error 17.24% error relative to the total pitch torque actuated.

Relation between thrust and control voltages

The flexured-gimbal device was mounted on a scale to measure the thrust of the FIR during the trials. As seen in Figure 3.7, there is some variation in the thrust with the varying control voltages, but the maximum deviation from the mean thrust is small (5.78%) and there is only a weak trend connecting roll and pitch voltages to thrust (a slope of -0.257 mg/V in pitch and -0.078 mg/V in roll).

3.5.4 Validation

Only five measurements were taken in the mapping stage with the validation fly, to produce a large enough range to develop a mapping trendline fit while preserving the lifespan of the FIR and reducing wear. Following the mapping measurements the FIR was removed from the device, then trimmed in free flight again. Notably, the trim values changed slightly due to wear even with the shorter mapping process, resulting in two different control voltage data points for zero torque in free flight (shown in Figures 3.8 and 3.9). Finally, the FIR was fitted with weights providing torque offsets in pitch and roll and trimmed in free flight, so that the control voltages needed to provide the torque counteracting the offsets could be calculated. Figures 3.8 and 3.9 show the measured data, as well as trend lines for the free flight experiment mapping and the device mapping. The measurement error of the device relative to the “ground truth” free-flight experiments was 9% for roll and 25% for pitch. The larger percentage error in pitch is due to the smaller size of torque being measured relative to the disturbance. The largest error in Figure 3.9, occurring at 0 V pitch, is about equal to the torque disturbance of the wire tether measured in [50] of 0.3 μNm.

3.6 Conclusion and Future Work

This paper reports a device design and measurement process that can be used to map control voltages to the resulting roll and pitch torques produced in a very small flapping-wing robot, even below a gram. This system is an improvement over an earlier system [36] that was only capable of finding

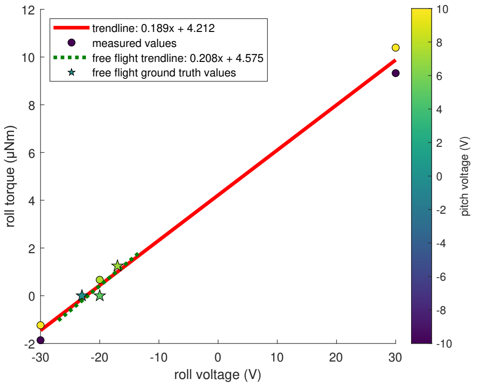


Figure 3.8: Control voltage to output torque mapping results from the device for the roll torque of the validation fly, along with the roll torque values mapped using free flight experiments with offset torques.

compensatory trim values rather than measuring torques directly. Like that system, the system here is constructed entirely using parts that are likely available in a lab or factory creating FIRs. Its gimbal and flexures are machined using the same laser system used to construct the robot itself. Readout is performed using a motion capture system that is standard equipment in many robotics settings. As a consequence of these choices, however, the device is limited to low-frequency measurements with bandwidth of approximately 0.3 Hz.

We found that the roll and pitch of the flying insect robot (FIR) we tested, the 180 mg UW Robofly, are decoupled and therefore can be actuated independently. This finding is consistent with the assumption that has been used to date in the design of the flight controllers of two-winged FIRs, which is that cross-axis coupling of torque commands is negligible. We anticipate this new information can be used to better model the dynamics of flapping-wing robots and control their movements more effectively, especially when undergoing aggressive (high-torque) maneuvers. The device is simple to construct and can be easily adapted for use with other types of flapping-wing robots.

Further improvements can potentially be made to the device to increase the ease of use by using an accelerometer as an inclinometer to measure the angles instead of requiring a motion capture

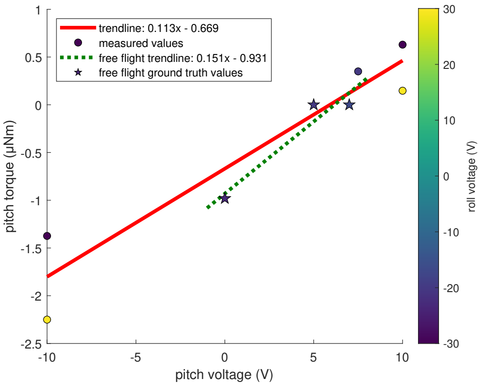


Figure 3.9: Control voltage to output torque mapping results from the device for the pitch torque of the validation fly, along with the pitch torque values mapped using free flight experiments with offset torques.

system. This would simplify its use. It is expected most if not all fully-autonomous FIRs will have an accelerometer as an integral part of their inertial navigation system. Preliminary work, however, indicates that vibrations due to the flapping wings produce too much noise to recover the angle. These vibrations could potentially be attenuated by placing the accelerometer on the damping rod, down near the glycerin.

Chapter 4

ACTUATING PURELY LONGITUDINAL FORCES IN A FLAPPING WING AERIAL ROBOT

The text of this chapter is a draft of a journal paper, which is under review. In this work, we introduce a capacity to generate purely longitudinal forces eliminating the need for inclining the robot before longitudinal movement, and significantly enhancing its agility. It allows the robot to navigate without altering the orientation of its sensors or landing gear, a fundamental advantage in terms of efficiency and control. Moreover, our research lends support to the hypothesis that certain flying insects also employ purely longitudinal forces in their flight dynamics. This builds upon previous findings that had measured such forces in tethered insects like fruit flies and dragonflies.

Dhingra, D., Fuller, S. Actuating Purely Longitudinal Forces in a Flapping Wing Aerial Robot, Science Robotics 2024 (Under review).

4.1 Abstract

There remains untapped potential to increase the maneuverability of flapping-wing aircraft by actuating their wings at a rate higher than the flapping frequency. We report a new capability for such flying robots that exploits this potential: actuating a force in a purely longitudinal direction. To do so, we change the speed of the upstroke relative to the downstroke. We present what we believe are the first purely longitudinal motions in free flight by a flapping-wing robot. Our experiments were conducted on the UW Robofly, an insect-sized aerial vehicle driven by piezoelectric actuators. We show that the quasi-steady aerodynamics model largely predicts the observed forces when simulated with wing motions captured by high-speed video. Our measurements show that the UW Robofly can actuate a longitudinal thrust of up to 0.2 mN while producing a vertical thrust of 1.8 mN. We furthermore introduce a method to estimate and compensate for undesirable pitch and roll torques that arise as a result of changes in wing motions required to actuate longitudinal thrust. Addressing these torques, we devise a method for measurement and correction via feedforward values. The ability to

actuate purely longitudinal forces avoids the need to incline the robot first, which improves agility and allows the robot to move without changing the orientation of its sensors or landing gear. Our work also supports the hypothesis that some flying insects also actuate purely longitudinal forces, building on previous results that measured such forces on tethered animals.

Introduction

Research in insect-scale flying robotics is growing due to the several potential use cases for these robots. Since these robots are low-cost and require low power for operation, one untethered, they can be used for applications like farmland monitoring, space exploration, and artificial pollination. Recent advances in tiny sensors and microcontrollers have propelled the research in these robots and brought them one step closer to achieving power [62] and sensor [47, 108] autonomy. Since the first flight of a piezo-actuated flapping wing robot [75] there have been multiple controller designs ranging from helicopter-like attitude and position controller using PID [74], adaptive controller [24], iterative learning [26] and more recently fast MPC controller [107] for trajectory tracking. Yaw control has recently been shown as a method to precisely control heading [28]. However, the existing research has predominantly focused on utilizing roll, pitch, and yaw actuation to control the six degrees of freedom (6-DOF) of flying insect-sized robots. In contrast, their biological counterparts—insects in nature—display a broader range of motion. In this study, our goal is to bridge the gap between flapping wing robots and their biological inspirations by introducing an additional mode of actuation—longitudinal force. Tethered experiments with biological flying insects have shown the ability to generate longitudinal force for precise motions. Reference [55] demonstrated that moths can increase the mean elevation angle of the wing, thereby altering the overall stroke plane. By manipulating this aspect, moths effectively accelerate themselves forward without inducing pitching motions. Fruit flies also use a propulsion strategy [95] by employing a paddling-type motion to generate force based on drag, propelling themselves forward. Furthermore, it is observed that dragonflies [9] use a steeper angle of attack and more wing velocity in the upstroke to move longitudinally backward.

These natural fliers, including moths, fruit flies, and dragonflies, possess intricate wing dynamics actuated by direct and indirect muscles. These mechanisms grant them the ability to execute precise

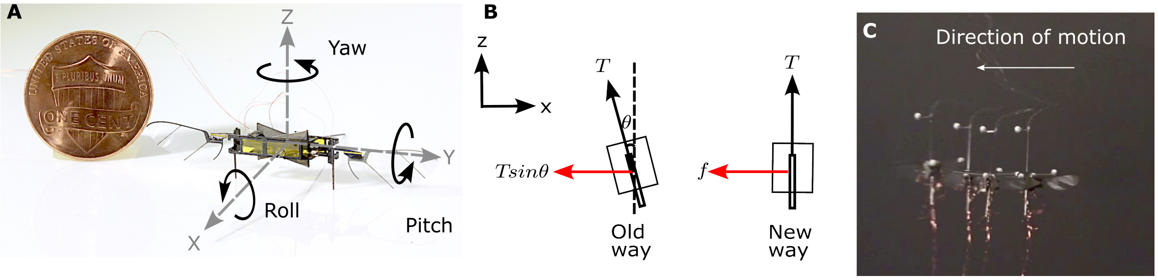


Figure 4.1: **Diagram of the main idea behind drag-based longitudnal force generation and the results in free flight.** **(A)** Robofly with its body translational and rotational axes. **(B)** Conventionality robot is tilted by an angle θ and the component of thrust ($T \sin\theta$) is used to move longitudinally. We show that flapping wing robots can generate longitudinal force (f) to move longitudinally without any tilt. **(C)** Photo composite of a purely longitudinal motion performed by the Robofly using purely stroke-averaged drag forces. The time interval between each frame is approximately 0.1 s.

longitudinal maneuvers and agile flight in response to various environmental conditions. Translating these advanced control mechanisms into robotic flapping wing systems that weigh 180 mg presents formidable challenges: 1) Flying insect robots are highly underactuated as they use only two piezoelectric actuators to control the 6 DOF of the robot. 2) These robots utilize passive wing hinges, a design that offers superior energy efficiency and robustness with fewer points of failure. However, this design also comes with a drawback—it lacks direct control over the angle of attack for the wing. While a design introduced in [111] offers direct control over the angle of attack, it is burdened by complexity and weight, making it less practical for real-world implementations. In 2015, a theoretical approach [81] to generate longitudinal force solely through wing stroke angles was presented. The idea involved employing a split-cycle signal to effectively alter the wing's speed between the upstroke and the downstroke. By doing so, they could generate a net stroke-averaged drag force on a flapping wing vehicle. While this concept holds promise, its practical implementation on physical robots is yet to be demonstrated.

Traditionally, in helicopter control, longitudinal translation of flying robots involves rotating the robot's thrust vector by applying the necessary pitch torque (as depicted in fig. 4.1-b). This rotation leads to the generation of a longitudinal force through the horizontal component of the tilted thrust

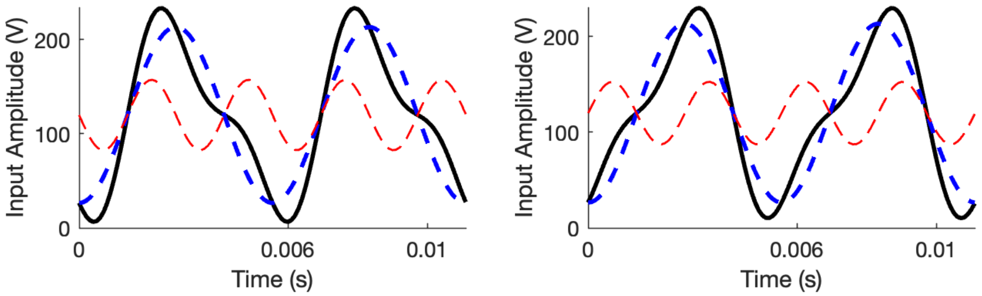


Figure 4.2: **Sinusoidal signal to produce asymmetric wing stroke.** The desired wing stroke kinematics can be obtained by combining the first harmonic (in blue) and the second harmonic (in red). When μ is set to -0.4 (on the left), it results in a faster upstroke compared to the downstroke, while the opposite is true for $\mu = 0.4$ (on the right).

vector. Direct longitudinal actuation is also possible on coaxial helicopters [87] and eight-rotor configuration aerial vehicles [13]. The primary focus of this study is to establish, for the first time, that flapping-wing robots can achieve longitudinal movement during free flight without tilting their thrust vector (fig. 4.1-C), mirroring the behavior of their biological counterparts. This capability holds the potential to mitigate wind disturbances, execute aggressive maneuvers, and enable the robot to move without rotation, thus maintaining a specific orientation for sensors or actuators.

The robot uses a sinusoidal signal to generate the flapping motion, as shown in fig. 4.2. This flapping motion generates a lift and a drag force. At the scale of robofly, (Re $\approx$ 3000) [112], air drag is dominated by inertial forces. This suggests that a flapping cycle's instantaneous drag force F_d varies quadratically with the relative air speed v.

$$F_d = \frac{1}{2}\rho C_D(\alpha) A v^2 \tag{4.1}$$

Here, drag coefficient C_D is a function of the angle of attack α, ρ is the density of the air, and A is the wing area. If the robot flaps its wings at the same speed in the upstroke and the downstroke, the drag force in each stroke would be equal and opposite, and the stroke averaged drag force would be zero. If the wings are flapped faster in the upstroke than the downstroke, relative airspeed, v, would be higher in the upstroke, resulting in a net drag force pushing the robot backward or vice versa. We call this longitudinal force and will use it to move the robot longitudinally. To generate this motion,

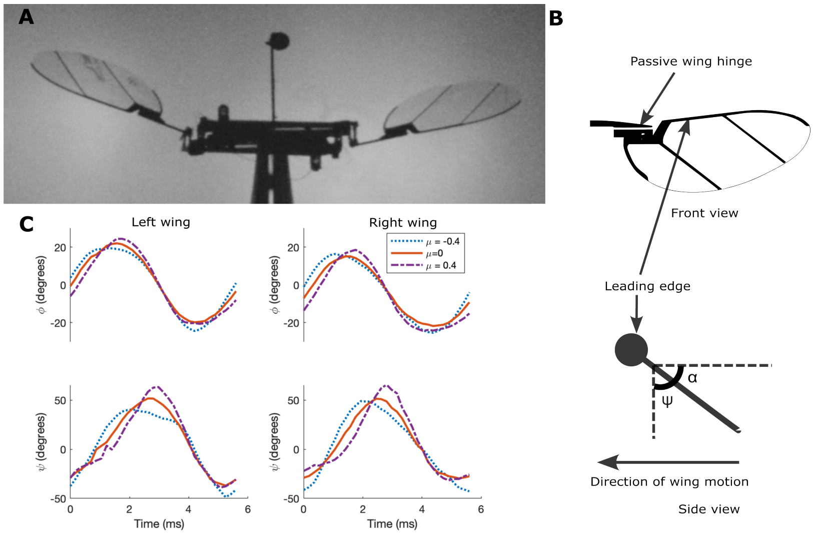


Figure 4.3: **Wing kinematic data was collected with a high-speed camera for calculating aerodynamic forces via quasi-steady simulation.** **(A)** A frame from a slow-motion video used to measure the wing angle and the angle of attack (α) of the Robofly. **(B)** Passive wing hinge used in Robofly, along with the convention for the angle of attack used in the quasi-steady model. **(C)** Shows the wing angle (ϕ) and the wing hinge angle (ψ) data for both the left and right wings. The data clearly illustrates that the required lag in the downstroke is achieved in the wing motion when μ is set to -0.4, while the opposite effect is observed for $\mu = 0.4$.

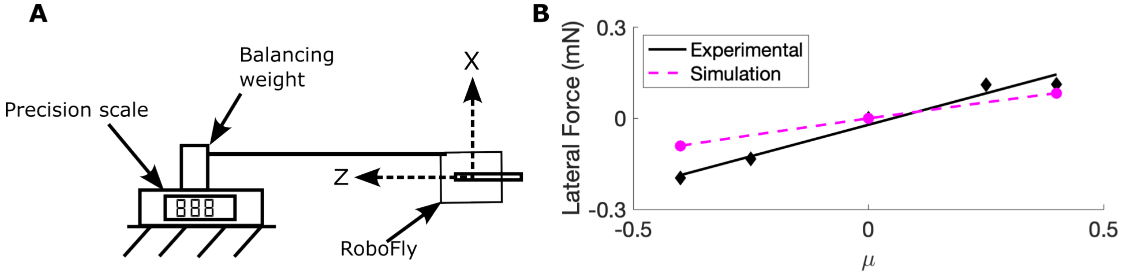


Figure 4.4: **Longitudinal force measurements on Robofly in simulation and experiments. (A)** Experimental setup for the measurement of longitudinal force generated corresponding to the μ value. **(B)** Longitudinal force produced by Robofly as it corresponds to the applied μ value, both in simulation and on the experimental setup. It's worth noting that in this context, a positive longitudinal force is directed toward the positive X direction.

the wings are flapped using an asymmetric signal to get different speeds of the wing between the upstroke and the downstroke. In a symmetric sinusoidal signal, the first half of the signal takes the same time as the second half of the signal; however, in an asymmetric sinusoidal signal, the first half may lead and take less time to complete resulting in the second half of the signal to lag and take more time to complete or vice-versa. The technique relies on the piezo actuator's ability for high-bandwidth actuation by flapping the wing in one direction faster than the other direction. As done in [29], this can be achieved by adding a second harmonic, scaled by μ, to the signal. The equation generates the resulting signal, $V(t) = A\sin(\omega t) + \mu A\sin(2\omega t)$. Here A is the amplitude of the sinusoidal signal, and ω is the flapping frequency. $\mu = 0$ generates a symmetric signal. By increasing μ, the first half of the signal lags and the second half of the signal leads, producing the required lag in the upstroke (fig. 4.2-middle). Similarly, decreasing μ produces the required lead in the upstroke (fig. 4.2-left).

4.2 Results

We employ the quasi-steady model of the robot, utilizing flapping wing data, to theoretically explore the longitudinal actuation mechanism through stroke-averaged drag force. These theoretical insights are substantiated through lab bench experiments aimed at quantifying the generated net stroke-averaged drag force. Additionally, we investigate the coupled torques arising in conjunction with longitudinal actuation, employing a torque measurement device. Ultimately, we validate the robot's longitudinal actuation in free-flight experiments, incorporating feedforward terms to counterbalance the coupled torques.

4.2.1 Quasisteady simulation

Split-cycle actuation entails the division of the constant-period wing stroke into distinct fast and slow strokes. Flapping dynamic theory posits that during the rapid upstroke ($\mu = -0.4$), a greater drag force is generated compared to the slow downstroke, resulting in a net backward thrust. Similarly, a larger drag force is anticipated during the swift downstroke ($\mu = 0.4$) compared to the slow upstroke, yielding net forward thrust.

We employ a quasi-steady blade element model of the Robofly to empirically validate this theory on our robot. This model simulates the net aerodynamic forces exerted on the wing in a flapping cycle. Given that the robot uses a passive wing hinge for wing pitch rotation (as illustrated in fig. 4.3-B), the prediction of the angle of attack can become intricate. To circumvent this complexity, we collected flapping angle and angle of attack data for three distinct scenarios: $\mu = -0.4$, $\mu = 0$, and $\mu = 0.4$. The data acquisition process used a high-speed camera (Phantom v7.3, Vision Research, Inc.) to capture wing motion at 5000 frames per second. This data was subsequently tracked using video analysis software (Tracker). The resulting graph in fig. 4.3-C visually presents the flapping angle (ϕ) and wing hinge angle (ψ) for the three cases.

We use a wing aerodynamics model based on a dynamically scaled physical model of a Drosophila wing [98, 99]. As shown in[115] this model constitutes a good approximation of aerodynamic forces on flapping wing robots of the scale of Robofly. More details on the model are presented in the methods section.

The graph in fig. 4.4-B depicts the net drag force acting on both wings throughout a flapping cycle for the three cases of μ. The graph is normalized to the force at $\mu = 0$ as our focus lies solely on longitudinal force variation. At $\mu = 0$, longitudinal force is compensated during takeoff via pitch trim. The graph shows that as μ is decreased to -0.4, there is more stroke averaged negative drag force and vice versa. This confirms the flapping dynamics theory of net drag force due to asymmetric stroke. According to the simulation, we can generate a drag force of 0.1 mN.

Moreover, the simulation indicates that manufacturing asymmetries between the two wings could result in differential drag forces—one wing experiencing greater drag than the other, culminating in a yaw torque coupled with longitudinal force. We posit that significant differences between the two wings are required for any substantial yaw torque generation due to the high damping around yaw rotation in flapping insects [19]. However, alongside drag force, manufacturing asymmetries also give rise to differences in lift forces between the wings. These differences, in turn, could induce roll and pitch torques. A detailed exploration of this phenomenon is further elaborated in section 4.2.3.

4.2.2 Longitudinal force measurements

To support the theoretical results obtained in the last section, we performed an empirical study on the same robot to measure the longitudinal thrust generated using a high-precision scale (fig. 4.4-A). The scale averages out the measured force at the frequency which is equivalent to measuring stroke-averaged force. The robot is securely held by a ceramic tweezer, which rests on the scale along with the balancing mass. To avoid any potential ground effects and ensure that no objects interfere with the robot's movement, we positioned the robot away from the base of the scale, holding it to the side. Before conducting the experiments, we did a thorough assessment to confirm that loading the scale from the side had no impact on the accuracy of our measurements. The measured mass remained consistent within the precision limits of the scale, whether positioned at the center of the platten or offset at the tip of the tweezer.

The results obtained from fig. 4.4-B indicate that in relation to changes in the μ value, ranging between -0.4 to 0.4, the longitudinal force changed from -0.2 mN to 0.12 mN. These findings were observed during the flapping of the robot's wings at a frequency of 180 Hz and an amplitude of 187 V. To provide context to this longitudinal force, it is noteworthy that this robot can generate a thrust of 1.8 mN under this frequency and amplitude conditions. These results almost match the theoretical prediction. We think the difference could be due to the assumptions used in the quasi-steady model.

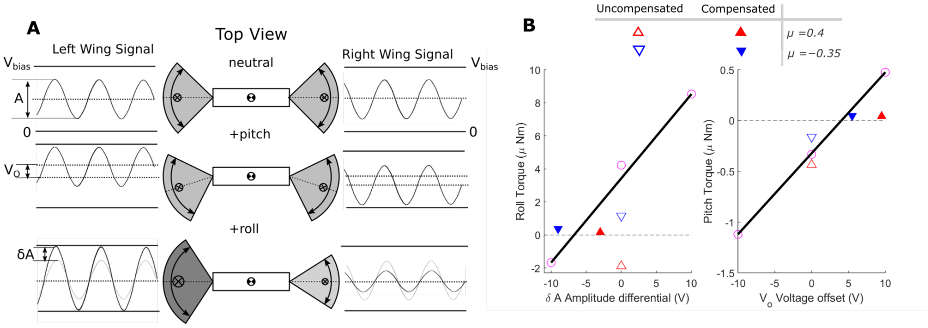


Figure 4.5: **Torque coupling and compensation.** **(A)** The figure illustrates the torque generation mechanism in flapping-wing robots alongside the corresponding input signal. **(B)** Compensation process using our torque measurement device fig. 4.8. The black line represents a least square fit that was used to compensate for torques arising due to non-zero μ values. Hollow triangles on the graph denote the measured coupled torques when no commanded torques were applied. Meanwhile, the filled triangles on the graph illustrate that the feedforward compensation has successfully eliminated the effect of torque coupling.

4.2.3 Torque Coupling and Compensation

Aerodynamic simulation using the flapping data suggests that asymmetries due to manufacturing uncertainties induce coupled roll torques when longitudinal force is actuated. Specifically, the two halves of the robofly exhibit different responses to the applied second harmonic signal, leading one half to generate more vertical thrust than the other, consequently causing a roll torque. Piezoelectric actuators are sensitive to slight changes in the input signal that alter the flapping characteristics of a wing. As we can see in 4.3, introducing mu changes the amplitude and flapping range of the wings and therefore shifts the aerodynamics center. This can also lead to the stroke-averaged aerodynamic center shifting to the front or back of the robot, giving rise to pitch torques. While it is hard to model asymmetries due to manufacturing uncertainties and changes due to the piezoelectric actuators as they are different for every robot, we can use the device introduced in [36] to measure these coupled torques.

Initially, we employed a torque measurement device to establish the correlation between the input voltage and the resultant torque produced by the robot. Subsequently, we measured the torques generated during longitudinal actuation. Detailed information about the device is in this study's Methods and Materials section. As illustrated in fig. 4.5-A, Robofly has the capability to independently generate both roll and pitch torques. Roll torque is generated by flapping one wing at a higher amplitude than the other, and this difference in amplitudes (δA) is linearly related to the roll torque produced by the robot, as previously documented [75]. Similarly, pitch torque is generated by adjusting the wing flapping in front or behind the robot's body. This adjustment is achieved by introducing a voltage offset (Vo) to the sinusoidal signal applied to the wings. The signal applied to the two wings of the robot can be expressed as follows:

$$V_l(t) = \left(A + \frac{\delta A}{2}\right)\sin(\omega t) + \frac{Vo}{2} + \frac{V_{bias}}{2} \tag{4.2}$$

$$V_r(t) = \left(A - \frac{\delta A}{2}\right)\sin(\omega t) - \frac{Vo}{2} + \frac{V_{bias}}{2} \tag{4.3}$$

Here $V_l(t)$ and $V_r(t)$ are the signals used to drive the left and the right wing of the robot, respectively. $\omega = 2\pi$ F, where F is the flapping frequency of the wings. V_{bias} is the bias voltage rail applied to the top of the actuator. Next, we define the linear relation for the torque in roll and pitch in response to the applied input:

$$\delta A = m_1 \tau_{roll} + c_1 \tag{4.4}$$

$$Vo = m_2 \tau_{pitch} + c_2 \tag{4.5}$$

Using the torque measurement device, we found the value of the parameters m_1 = 1.96 V/μNm , m_2 = 12.5 V/μNm, c_1 = -6.7 V, and c_2 = 4 V .

In Figure 4.5-B, we have plotted the linear relationship represented by equations 4 and 4.5. It demonstrates that a 20 V change in δA leads to an 11 μNm roll torque, while a 20 V change in Vo results in a 1.6 μNm pitch torque.

Next, we conducted specific torque measurements related to longitudinal actuation, as indicated by the hollow triangular shaped markers in Figure 4.5-B. When μ = 0.4, it generated a coupled roll torque of -1.9 μNm and a pitch torque of -0.43 μNm. Conversely, μ = -0.35 produced a coupled roll torque of 1.15 μNm and a pitch torque of -0.15 μNm.

To compensate for these coupled torques using a feedforward command, we substitute the negative values of these torques in equations 4 and 4.5. This process allowed us to determine the feedforward inputs Vo_{ff} and δA_{ff}. With μ = -0.35, we require a feedforward adjustment of δA_{ff} =

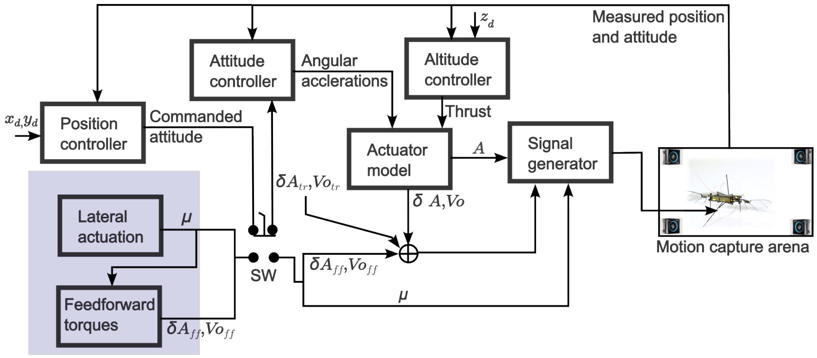


Figure 4.6: **The control architecture utilized to induce longitudinal force on the Robofly.** It comprises a cascaded closed-loop control system consisting of an outer tilt-based position controller and an inner attitude controller loop. x_d, y_d, and z_d define the position set point. The switch, denoted as SW, enables the transition between the position controller and longitudinal actuation. Open-loop longitudinal actuation is executed in combination with the feedforward torques that are directly applied to the signal generator. Here δA_{tr} and Vo_{tr} are the trim inputs and δA_{ff} and Vo_{ff} are the feedforward inputs.

-9 V and $Vo_{ff} = 7$ V to offset the coupled torques during free flight, and for $\mu = 0.4$ it necessitates a change of δA_{ff} of -3 V and Vo_{ff} of 9 V to compensate for the coupled torques. As seen by the filled triangular shaped markers in Figure 4.5-B, the torque measurements after applying these feedforward commands in conjunction with longitudinal actuation are close to zero. This validates our procedure for the compensation of coupled torques through the torque measurement device.

In future, these torques can be compensated during free flight using a fast attitude feedback loop with an onboard gyroscope. A gyroscope can estimate attitude much faster than motion capture cameras.

4.2.4 Longitudinal actuation during hover

To see the longitudinal actuation during a stable hover, we implemented the cascaded attitude and altitude controller with a tilt-based outer loop for position control as described in [74]. As shown in fig. 4.6 we have a longitudinal actuation block along with the hovering controller. Once a stable hover is achieved, we turn off the tilt-based position controller and turn on the drag-based open-

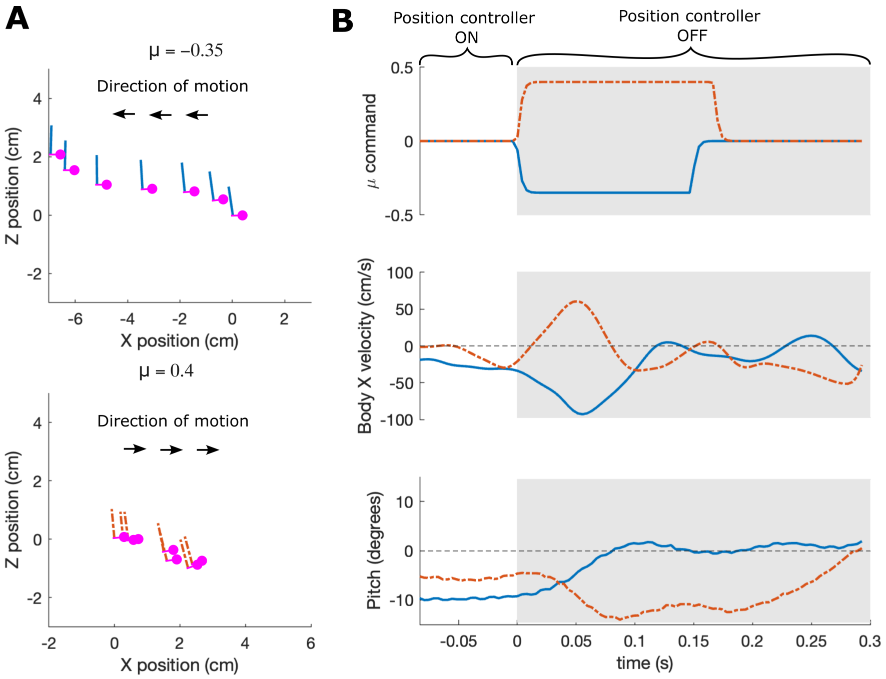


Figure 4.7: **Longitudinal actuation during hover.** **(A)** Trajectory of the robot with inclination during longitudinal actuation in X-Z plane. The pink markers mark the front of the robot. (top) The robot moves backward with negative μ, and (bottom) moves forward with positive μ input. The time interval between each marker is 20 ms **(B)** X velocity of the robot in the body coordinates. The robot moves longitudinally in X as the μ actuation is turned on. The pitch deflection (bottom) is in the opposite direction and does not contribute to the longitudinal motion.

loop longitudinal actuation along with the feedforward inputs. This is done using a switch SW. It's important to note that the attitude controller remains active, maintaining the robot's orientation in the vertical position. Feedforward input δA_{ff} and Vo_{ff} are summed with the trim settings δA_{tr} and Vo_{tr} and the output from the actuator model to feed to the signal generator. For our experiments, we turn the longitudinal actuation switch on for 0.2 s and wait another 0.1 s before we turn the tilt-based position controller on.

Fig. 4.7-A depicts trajectories of longitudinal actuation, both forward and backward, with inclination about the pitch axis. Each stick of the trajectory is plotted at the interval of 20 ms. When μ is set to -0.35, the robot moves backward, as indicated by the direction of motion. Conversely, the robot moves forward when μ is set to 0.4. The graph in fig. 4.7-B illustrates the body horizontal

velocity (x velocity) for these two trajectories. The robot successfully achieves a negative velocity with a negative μ command and a positive velocity with a positive μ command.

Notably, the pitch angle undergoes changes in the opposite direction of what it would experience if the longitudinal movement were due to pitch rotation. We believe this alteration in pitch is attributed to the aerodynamic center of the robot being situated below its center of mass.

4.3 Discussion

To fully realize the potential of insect-sized flapping wing robots, we must enhance their maneuverability to closely match their biological counterparts. At present, these robots are restricted to control in roll, pitch, and yaw; however, our research has successfully introduced a novel mode of actuation. This is possible due to the high bandwidth of piezoelectric actuators. We showed that Robofly can generate drag-based longitudinal force using blade element quasi-steady model. These simulation results were validated through lab bench experiments where we measured the longitudinal force with a high precision scale. We also found that actuating in this way produced torque coupling due to asymmetries in manufacturing; these torques exhibit low magnitudes, approximately 2 μNm in roll and 0.4 μNm in pitch. To mitigate this, we introduced a method for measuring these torques using a torque measurement device. The device is also used to calculate the feedforward values that can correct for these coupled torques. Our research achieved a milestone by successfully demonstrating what we believe is the first direct actuation of the longitudinal thrust on a flapping wing vehicle in free flight.

One critical future task involves devising a design methodology for piezo-actued wings that could maximize the longitudinal thrust. This would be useful for performing a controlled trajectory tracking maneuver. Another promising avenue involves leveraging the low-latency capability of an onboard gyroscope to directly compensate for torques, eliminating the need for a prior calibration step.

Previous works on fruit flies and dragonflies have shown their ability to generate longitudinal force by varying wing speed between the upstroke and downstroke. This was shown on insects that were tethered, breaking their feedback loops and potentially altering their behavior. Here we demonstrated on an at-scale robotic flapping-wing insect that such motions can be physically realized. We believe that a capability for purely longitudinal motions would be adaptive in insects for the same reasons outlined above that would be advantageous for robots.

This advancement also paves a way for control algorithms that can optimally combine the two modalities – longitudinal thrust and pitch torque – subject to their respective actuation constraints. Such a controller would leverage the richness of flapping-wing flight to bring robotic flapping-wing systems closer to the performance of their biological counterparts.

4.4 Materials and Methods

4.4.1 Quasisteady split cycle model

We use a wing aerodynamics model based on a dynamically scaled physical model of a Drosophila wing [98, 99]. As shown in[115] this model constitutes a good approximation of aerodynamic forces on flapping wing robots of the scale of Robofly. Employing the standard blade element method, the wing is divided spanwise into a finite array of blade elements, each subjected to force calculations. The resultant force on each blade element, denoted as F, emerges as the sum of the lift force (F_L), drag force (F_D), and the rotational force (F_r) expressed as $F = F_L + F_D + F_r$. For simplicity, we approximate the wing by a trapezoidal shape with chord length linearly decreasing from 5.07 mm to 3.2 mm and a span of 12.5 mm. The wing root is located at 4.9 mm from the axis of wing rotation. The total area of the trapezoidal wing is 51.6 mm^2, which is close to 52.5 mm^2, the area of the wing used in Robofly.

The lift and the drag force are directly proportional to the product of air density (ρ), element chord length (c), element width (δr), and the square of the incident flow velocity within the plane of the blade element (v). The expressions denoting the magnitudes of lift and drag are presented as follows.

$$F_L = \rho c|v|^2 C_L(\alpha)\delta r \tag{4.6}$$

$$F_D = \rho c|v|^2 C_D(\alpha)\delta r \tag{4.7}$$

Where α is the local angle of attack of the blade element, and the lift and drag coefficients are $C_L(\alpha)$ and $C_D(\alpha)$. Empirical values of lift and drag coefficient are used from the dynamically scaled model of Drosophila wing [38].

$$C_L(\alpha) = 1.8\sin(2\alpha) \tag{4.8}$$

$$C_D(\alpha) = 1.9 - 1.5\cos(2\alpha) \tag{4.9}$$

The rotational force always acts in a direction normal to the surface of the wing, and an expression for its magnitude is given by

$$F_r = C_r \rho \alpha c^2 |v| \delta r \tag{4.10}$$

where C_r is the rotational force coefficient. A value of $C_r = 1.55$ is suggested by Sane [99] as appropriate for the Drosophila wing.

The model employs manually measured angles of attack obtained from high-speed video frames captured from an overhead perspective. In this model, the added mass effect is disregarded due to its dependence on the rate of change in the angle of attack—a parameter susceptible to measurement errors in the angle of attack. For simplicity, the model also neglects wing flexion and deformation. Furthermore, it assumes that aerodynamic forces remain unaffected by wing-wing and wing-body interactions.

4.4.2 Torque measurement device

The device introduced in [36] restricts the robot's movement to rotational motion along two axes: pitch and roll (fig. 4.8). When the robot is mounted on the device and turned on with applied μ, it applies a specific torque, leading to an angular deflection of the device. This angular deflection is linearly related to the applied torque. The angular deflection is measured using the motion capture system (four Prime 13 cameras, OptiTrak, Inc., Salem, OR).

The device is manually calibrated to convert the angular deflection into torque. This involves suspending a known weight at a known distance from the rotational axis and recording the corresponding deflection. The calibration constant about the pitch axis is 1.62 μNm/degree and 2.07 μNm/degree about the roll axis.

The functionality of the device relies on the assumption that the robot's center of mass aligns with the center of rotation of the device, thereby minimizing errors originating from eccentric load-

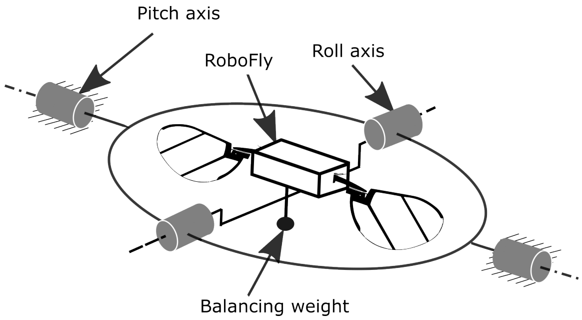


Figure 4.8: **Principle of torque measurement device.** The robot is allowed to rotate about the roll and pitch axis. Torque is estimated by measuring the extent of rotation about the given axis using motion capture cameras and the knowledge of the stiffness of the device. The design of the device ensures that the robot's center of mass approximately aligns with the axis of rotation in pitch and roll.

ing. As mentioned in [36], this alignment is ensured by mounting the robot within the motion capture arena and verifying that placing the robot on the device does not induce any angular rotation.

Chapter 5

MODELING AND LQR CONTROL OF INSECT SIZED FLAPPING WING ROBOT

The text of this chapter is a draft of a journal paper, which is under review. In this work, we developed a first-principle dynamic model of the robot using torque mapping, thrust mapping, and the CAD model of the robot. This allowed us to do comparison of high-speed free flight accelerations with those predicted by our stroke-averaged model. We successfully performed hovering and trajectory tracking maneuvers using an LQR controller based on this model. We anticipate that our model and LQR controller will be key components in fast onboard receding horizon optimal controllers, optimizing performance under actuator limits and state constraints. These controllers could enable more aggressive maneuvers by adjusting control gains for high translation speeds and attitude angles, leveraging the robots' high torque-to-inertia ratios.

Dhingra, D., Kaheman, K. and Fuller, S. Modeling and LQR Control of Insect Sized Flapping Wing Robot, npj Robotics 2024 (Under review).

5.1 Abstract

Flying insects can perform rapid, sophisticated maneuvers like backflips, sharp banked turns, and in-flight collision recovery. To emulate these in aerial robots weighing less than a gram, known as flying insect robots (FIRs), a fast and responsive control system is essential. To date, these have largely been, at their core, elaborations of proportional-integral-derivative (PID)-type feedback control. Without exception, their gains have been painstakingly tuned by hand. Aggressive maneuvers have further required task-specific tuning. Optimal control has the potential to mitigate these issues, but has to date only been demonstrated using approxiate models and receding horizon controllers (RHC) that are too computationally demanding to be carried out onboard the robot. Here we used a more accurate stroke-averaged model of forces and torques to implement the first demonstration of optimal control on an FIR that is computationally efficient enough to be performed by a micro-

processor carried onboard. We took force and torque measurements from a 150 mg FIR, the UW Robofly, using a custom-built sensitive force-torque sensor, and validated them using motion capture data in free flight. We demonstrated stable hovering (RMS error of about 4 cm) and trajectory tracking maneuvers at translational velocities up to 25 cm/s using an optimal linear quadratic regulator (LQR)[1]. These results were enabled by a more accurate model and lay the foundation for future work that uses our improved model and optimal controller in conjunction with recent advances in low-power receding horizon control to perform accurate aggressive maneuvers without iterative, task-specific tuning.

5.2 Introduction

Research in flapping wing insect-sized robots (FIRs) is motivated by their potential applications. These robots are small in size and are inexpensive to manufacture at a large scale, which makes them suitable for applications like detecting gas leaks, looking for survivors in disaster-prone areas, automated farm monitoring, running inspections on manufacturing lines, and weather monitoring. While still tethered and limited to operation inside a lab environment recent advances in tiny sensors and microcontrollers have brought them one step closer to achieving power [62] and sensor [108] autonomy.

Controlling FIRs presents significant challenges due to their highly nonlinear dynamics, manufacturing inconsistencies resulting in variability between robots, rapid wear and tear, and a high torque to moment of inertia ratio, approximately 10^3 rad/sec^2, which leads to extremely fast dynamics.

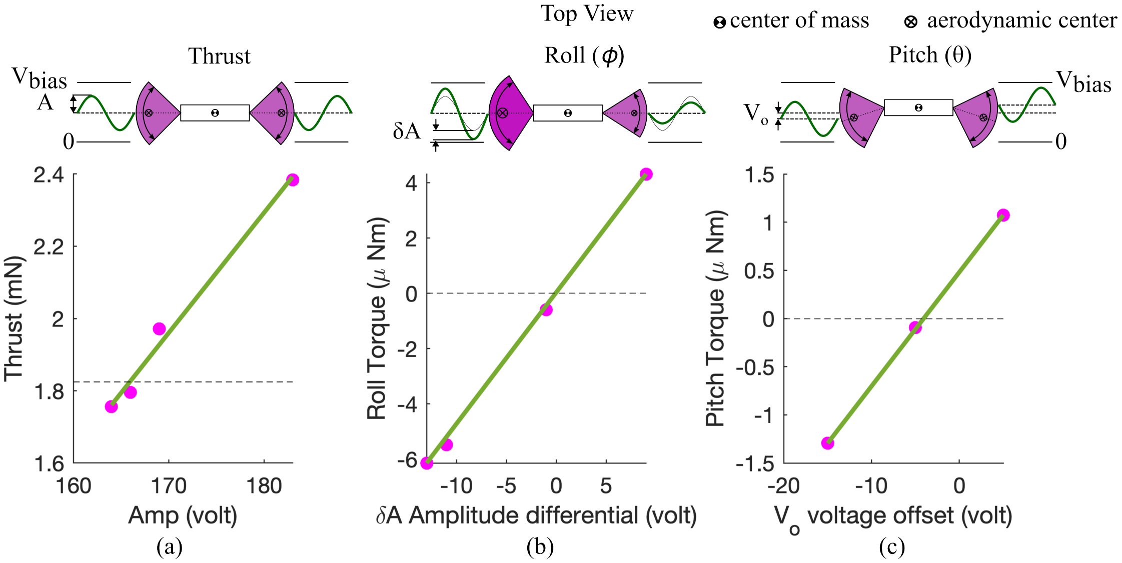
Current state-of-the-art controllers for FIRs primarily utilize adaptive PID flight control systems for hovering [24]. Despite their widespread use, these controllers require substantial ad-hoc tuning and are task-specific, often failing to consider actuator, state, and environmental constraints. Maneuvering beyond basic linear responses, such as perching [52] and somersault [17] , requires a sliding mode controller combined with iterative learning of trajectory parameters. However, the parameters that were derived to perform an aggressive perch in [52] are specific to that task and cannot be applied to any other task.

Recent research has introduced optimal control strategies like modular Model Predictive Con-

Figure 5.1: RoboFly, an insect-sized flapping robot weighing 146 milligrams, hovers next to a flower using feedback from motion capture cameras. The robot performs this hovering maneuver using the LQR controller reported in this work.

trol (MPC) [32], which combines high-level MPC with a low-level controller for torque management, enabling operation beyond hovering. However, these systems have not been demonstrated on actual hardware for maneuvers beyond hovering. Additionally, data-driven MPC approaches like Tube-MPC [107] show promise for optimizing under actuator constraints and trajectory tracking for complex maneuvers such as ramps and infinity loops but remain too computationally intensive for implementation on sub-150 mg robots. Microprocessors small enough to be carried onboard, such as the 10 mg, 120 MHz STM32F4 used in the first wireless liftoff of an FIR, the UW Robofly in [62], are capable of floating-point math operations. Nevertheless, their performance will be limited to a fraction of desktop capabilities, just a few hundred MHz, for the foreseeable future.

The primary focus of the work in [7] is the precise tracking of the yaw angle, originally introduced in [30]. The PID controller used in [7] is tuned based on the robot's dynamics. This enables the robot to hover and follow trajectories, such as an infinity loop. Our approach to controlling FIRs is based on the premise that an accurate model eliminates the need for laborious and unsatisfactory hand-tuning of PID gains. By framing the control problem within an optimal framework, we can de-



(a) (b) (c)

Figure 5.2: Torque and Thrust Generation Mechanism in FIRs: (Top) Inspired by the work in [36], this figure shows that changing the signal parameters δA and V_o introduces roll and pitch torques, respectively. Here, V_{bias} is the bias voltage. (Bottom) Mapping of (a) thrust, (b) roll torque, and (c) pitch torque of the RoboFly used in this work. The thrust mapping is obtained using a high-precision scale, while the torque mappings are obtained using the torque measurement device introduced in [36]. Pink dots represent the collected data points, and the green line represents the linear fit of the data. The corresponding equations for these linear fits are provided in Table 5.1.

sign performance to maximize metrics like power efficiency or completion time. Using this model, we compute optimal gains around a fixed point for control. This was first demonstrated in [11], where the performance of PID and LQR controllers were compared on quadrotors with the target to stabilizing the orientation angles. The LQR controller demonstrated faster response in reaching the reference signal from high initial angles. The recent development of low-cost and accurate torque measurement device [36] has improved our ability to map the torque characteristics of these robots accurately. Furthermore, innovations such as pre-stacked actuators [60] and standardized manufacturing processes [29] have reduced variability in actuator performance and robot construction. In this work, we have made the following contributions:

1. For the first time, we developed and validated a stroke-averaged first-principle model by com-

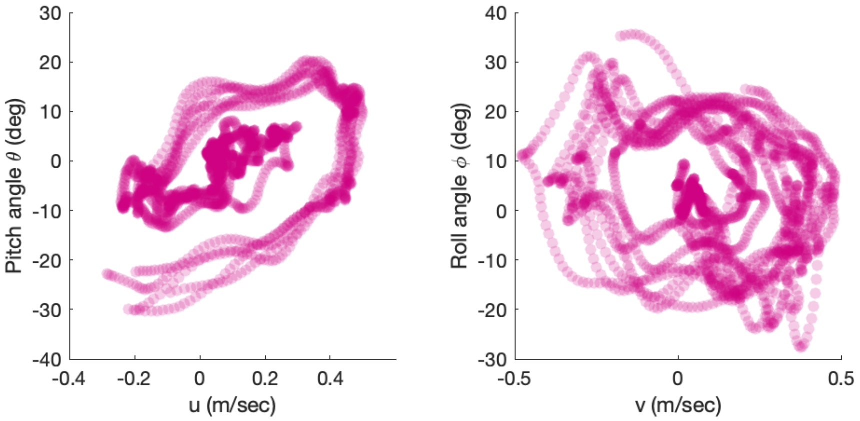


Figure 5.3: Visualization of the collected data: The graph shows the robot achieving high attitude angles greater than 30° and corresponding lateral/longitudinal speeds exceeding 0.4 m/s in the collected data. This highlights significant perturbations, which will be used to validate the stroke-averaged dynamics developed in this work. The color intensity on the graph represents the density of data points.

paring it with high-speed trajectories collected from a sub-150 mg robot.

2. To the best of our knowledge, we present the first LQR implementation of controlled flight on an FIR.

This control strategy is computationally efficient, requiring a relatively small number of multiply-accumulate operations and trigonometric calculations per control step making it feasible for integration on tiny microcontrollers like the STM32F4, suitable for sub-150 mg robots. We expect our model and LQR controller will be able to serve as integral elements in a fast onboard receding horizon optimal controllers, such as those discussed in [4] and [44], that can optimize under actuator limits and state constraints. With these receding horizon controllers, the presented model can enable more aggressive maneuvers due to the robots' high torque-to-inertia ratios, by scheduling the control gains for high translation speeds and attitude angles.

5.3 Results

5.3.1 RoboFly

RoboFly (shown in Fig. 5.1) is a flapping-wing robot that weighs 150 mg. The robot features two piezoelectric actuators as muscles to flap its wings. These actuators are linked to a transmission mechanism that amplifies the actuators' displacement of approximately 200 μm to a wing motion of about 60°. RoboFly has the capability to carry a payload up to 1.5 times its own weight and is powered via a wire-tether, which comprises four wires transmitting signals to operate the actuators.

RoboFly, like other piezo-actuated flapping wing robots [74][118], is operated by low-power 180 volt sinusoidal signals. It can generate roll torque, pitch torque, and thrust almost independently [114]. The analog voltage signal is generated using the equation,

$$V_{signal} = \frac{A+\delta A}{2}\sin\omega t + \frac{V_{bias}}{2} + \frac{V_o}{2} \tag{5.1}$$

Here, V_{bias} is a constant bias signal voltage of 250 volt supplied to the top layer of the actuator. As shown in Fig. 5.2 (a), increasing the amplitude (A) of the sinusoidal signal increases the wing flapping amplitude, thereby generating greater thrust. Creating an amplitude differential (δA) between the wings, shown in Fig. 5.2 (b), increases thrust on one side while decreasing it on the other, which produces roll torque. Pitch torque (Fig. 5.2 (c)) is generated by adjusting the wing flapping either forward or backward relative to the robot's body through a voltage offset (V_o) applied to the sinusoidal signal.

5.3.2 Theoretical Model

The dynamics of a flapping wing robot of the size of RoboFly can be defined by the same first principle model as a quadrotor [7]. Using the convention of (X,Y,Z) as coordinates in an inertial frame and (x_b,y_b,z_b) as coordinates in the body frame, first-order differential equations of the body velocity $V_b = [u,v,w]^T$ of the robot is defined by

$$\dot{u} = g\sin\theta + f_{a_1} - (qw - rv) \tag{5.2}$$

$$\dot{v} = -g\cos\theta\sin\phi + f_{a_2} - (rv - pw) \tag{5.3}$$

$$\dot{w} = -g\cos\theta\cos\phi + f_{a_3} - (pv - qu) + \frac{\Gamma}{m+m_M} \tag{5.4}$$

Here, $\omega_b = [p, q, r]^T$ is the body angular velocity, g is the acceleration due to gravity, m is the mass of the robot and, m_M is the mass of the MoCap markers. Rotation is defined by the widely used 321 rotation matrix. The 321 rotation follows the yaw (ψ) → pitch (θ) → roll (ϕ) sequence. Γ is the thrust applied to the robot in the positive z_b direction. $F_a = [f_{a_1}, f_{a_2}, f_{a_3}]^T$ is the unmodeled dynamic force, which includes stroke averaged aerodynamic force in body frame caused by the drag due to flapping wings. Similarly, rotational dynamics can be defined by the first-order differential equation in body angular velocity ($\omega_b = [p, q, r]^T$).

$$\dot{p} = L + \frac{\tau_r}{J_{xx}} - \frac{J_{zz} - J_{yy}}{J_{xx}} qr \tag{5.5}$$

$$\dot{q} = M + \frac{\tau_p}{J_{yy}} - \frac{J_{xx} - J_{zz}}{J_{yy}} rp \tag{5.6}$$

$$\dot{r} = N - \frac{J_{yy} - J_{xx}}{J_{zz}} pq \tag{5.7}$$

Here, $J = diag([J_{xx}, J_{yy}, J_{zz}])$ is the diagonal moment of inertia matrix of the robot. $\tau_c = [\tau_r, \tau_p, 0]^T$ is defined as the input torque vector comprising of roll torque (τ_r) about x_b and pitch torque (τ_p) about y_b. $\tau_a = [L, M, N]^T$ is the unmodeled dynamic moment, which also includes averaged aerodynamic torque caused by the drag due to flapping wings.

Model Parameters

We precisely measured the robot's parameters using a sensitive, custom-build force torque sensor. Results are tabulated in Table 5.1. The mass of the robot was determined using a high-precision balance with a resolution of 0.1 mg. To estimate the robot's moment of inertia, we measured the mass of its various components, including the piezoelectric actuators, wings, airframes, and motion capture markers. These measurements were then input into a CAD model to calculate the robot's moment of inertia matrix. The high-precision scale also facilitated in the calculation of thrust mapping, where the thrust generated by the robot at specific flapping amplitudes (A) was recorded. During the experiment, the wings were flapped for a period of 1 sec at the constant frequency of 180 hz. The scale can take accurate stroke averaged measurements of thrust. The test setup also makes sure that the robot is away from any surrounding objects to avoid ground effects. We employed a least squares fit method to model the thrust based on this data (Fig. 5.2-left). The learned linear fit of the thrust mapping is shown in Table 5.1. Given that the lifetime of these robots is about 10 minutes [77], we aimed to minimize the total operating time to prevent mechanical fatigue. Therefore, for the mapping of thrust and torques, we took only two or three measurements to establish a trendline that can be used in the model.

To measure the torque response to different voltage inputs, that is, torque mapping, we used a device similar to the one in [114]. By applying inputs V_o and δV, we generate roll and pitch torques, respectively. These torques induced angular deflections on the device, which are linearly correlated with the applied torques, which was observed in practice in [114]. These deflections were accurately measured using the motion capture system, allowing us to map torques effectively (Fig. 5.2-middle and right). This comprehensive approach to parameter measurement ensures a robust foundation for our model.

5.3.3 Trajectory Data

We collected 8 seconds of trajectory data from 7 separate flights with wings flapping at a frequency of 180 Hz, controlled by a PID flight controller [29]. The data is provided in the supplementary materials (Supplementary Data−Model_validation_traj1 to Supplementary Data−Model_validation_traj7). To capture flight perturbations, we set the desired points away from the robot's initial position, focusing on collecting more data with perturbations in lateral, longitudinal, and vertical dynamics. To avoid capturing too much stable hovering data, most flight trajectories were shortened to the duration required for the robot to reach the set positions. The robot was equipped with four retroreflective markers to track its position and orientation through a motion capture system comprised of four Prime 13 cameras by OptiTrak, Inc., Salem, OR. Position and quaternions from the motion capture system running at 240 hz were used to calculate V_b, ω_b, and Euler angles of the robot offline.

Body Offset

A critical difference between the robot's trajectory data and the modeled dynamics from equations 5.2-6.6 can stem from the misalignment between the body z-axis, as defined in the motion capture software, and the robot's thrust vector. The dynamics, detailed in equations 5.2-5.4, assume that the thrust vector is perfectly aligned with the body z-axis, an assumption that may not hold in practice. This misalignment issue occurs because the thrust vector's direction is not known at the time that the robot's body coordinates are defined in the motion capture software. A tilted thrust vector introduces lateral and longitudinal forces. To reduce this error, we redefined the body coordinate system after performing a short 0.3 s uncontrolled trimmed flight. Since during a trimmed flight, the robot takes off approximately vertically, so its trajectory can be used to estimate the direction of the thrust vector and therefore align the z-axis of the body coordinate system to that direction. This, however, is based on the assumption that the body coordinate system is not redefined in between the experiments and remains the same throughout the data collection and control process.

Parameter	**Symbol**	**Value**
Mass of the robot	m	150×10^{-6} kg
Mass of the MoCap markers	m_M	36×10^{-6} kg
Moment of inertia	J	diag([3.12×10^{-9}, 2.97×10^{-9}, 0.55×10^{-9}]) kg.m^2
Thrust	Γ	$3.27 \times 10^{-5} A - 0.0024$ N
Roll Torque	τ_r	$0.48 \times 10^{-6} \delta A$ Nm
Pitch Torque	τ_p	$0.11 \times 10^{-6} V_o$ Nm

Table 5.1: Measured parameters of RoboFly. Calculated Moment of inertia J is a diagonal matrix of moment of inertia about the principal axes. Mapping from A to Γ, δA to τ_r, and V_o to τ_p are the equations of linear fit from Fig 5.2 (a),(b) and (c) respectively.

Visualization of Collected Data

Unlike traditional aircraft, where the flight envelope is defined by changes in velocity against variations in angle of attack, flapping wing robots exhibit continuous angle of attack variations throughout the flapping cycle. Therefore, we determined that representing the flight state in terms of angular orientation and velocity would be more pertinent characterization. Characterizing in this way elucidates the robot's ability to maintain controlled flight across different tilt angles and the associated longitudinal/lateral speeds at these angles, which are critical for maneuverability. A robot that can sustain higher tilt angles and translational speeds in controlled flight is indicative of superior maneuverability, enabling it to execute tighter turns. As shown in Fig. 5.3, based on data collected from the flight trajectories of the RoboFly, the robot is able to get to attitude angles of approximately 30° and body velocity of around 0.4 m/sec.

5.3.4 Controller Implementation

In our experiments, we did not control the yaw rotation of the robot. Hence the LQR was designed to optimize the dynamics in the body coordinate system, which are independent of the yaw rotation. The state vector of the robot is defined by $\sigma = [d_x, d_y, d_z, u, v, w, \phi, \theta, p, q]^T$. Here, $d = [d_x, d_y, d_z]^T$ is the position in body coordinates and $V_b = \dot{d} = [u, v, w]$. The controller assumes that the angular velocity about z_b, denoted as r, which arises from manufacturing uncertainties, remains constant. Its value affects fictitious forces and torques in body coordinates that appear in equations 5.2-6.6. The controller calculates the inputs $u^* = [A, \delta A, V_o]$, which directly controls the acceleration in z_b, torque about x_b, and torque about y_b axis. The Q matrix used in our experiments is a diagonal matrix, $Q = diag([0.02, 0.02, 0.01, 0.1, 0.1, 0.1, 1, 1, 4, 4])$; R is also a diagonal matrix, $R = diag([2, 1, 1])$. The ratio of Q and R matrix used here was obtained with the knowledge of our model and only one experimental flight. The feedback loop includes a motion capture system that provides state feedback in terms of the robot's position and orientation (expressed as quaternions). This data feeds into a Simulink real-time system in which the body angular velocities, euler angles, and velocity in world frame are calculated. The controller receives the desired position in the world coordinates. Finally, the resulting error in the world coordinates is converted to the body coordinates and is multiplied by the pre-calculated LQR gain matrix k to determine the control inputs, $u = -k(\sigma_{des} - \sigma)$, for the robot. Control loop used for our experiments is shown in Fig. 5.5(a).

5.3.5 Model Validation

For simplicity, the model we used in this work excludes damping and drag components, thus setting the force and moment vectors $[f_{a_1}, f_{a_2}, f_{a_3}]^T$ and $[L, M, N]^T$ to zero in equations (5.2-6.6). We validated the theoretical model accelerations by comparing them with measured acceleration data from flight tests, as depicted in Fig. 5.4. This comparison includes seven separate flight trajectories stacked in time.

The model can predict translational accelerations in the body coordinate system, represented

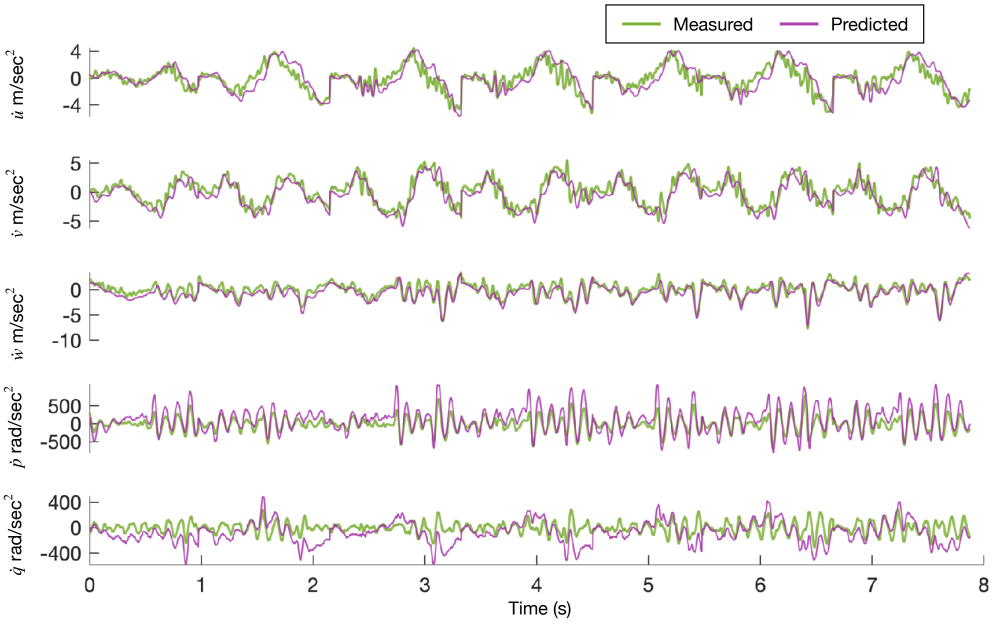
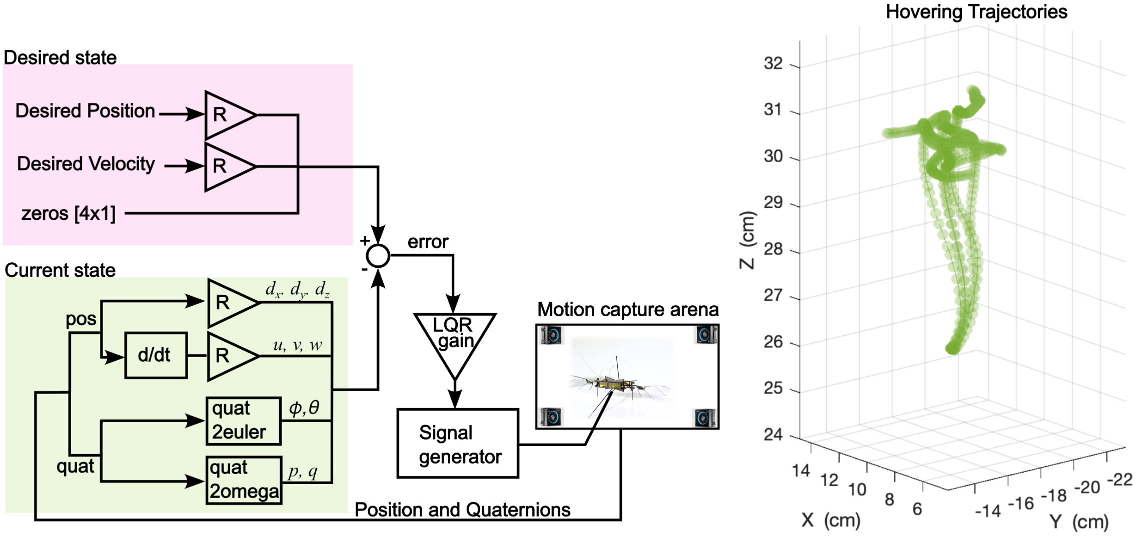


Figure 5.4: Model validation plots: Measured accelerations (green) from the RoboFly trajectories plotted with the predicted accelerations (pink) calculated using the theoretical model.

by $[\dot{u}, \dot{v}, \dot{w}]^T$, with the root mean squared (L^2) error of 53.4 m/sec^2, 56.9 m/sec^2, and 36.7 m/sec^2, respectively. The errors in rotational accelerations, $[\dot{p}, \dot{q}]^T$, are 9.2e^3 rad/sec^2 and 7.6e^3 rad/sec^2, respectively. The more substantial errors in the rotational domain can largely be attributed to the aforementioned aspect of small flight vehicles that their angular accelerations are large.

We believe, and our results show, that the model is still adequate for controller design purposes. This is based on two key considerations: Firstly, the actuation delay for the rotational system is minimal, as rotational acceleration occurs almost instantaneously once torque is applied. Secondly, the robot will eventually have a gyroscope onboard which is capable of providing rapid rotational velocity feedback at 1 to 16 kHz, which will significantly enhance the ability to perform rapid feedback corrections. Thus, even with its limitations, this model provides a sufficient foundation for developing an effective controller.

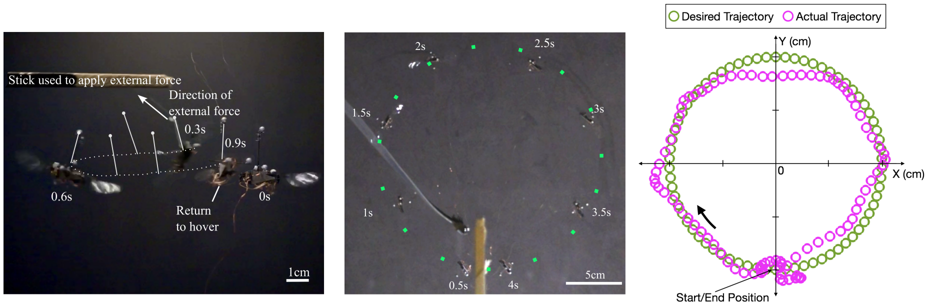


(a) (b)

Figure 5.5: LQR control loop and hovering trajectories: (a) The LQR control loop used to perform hovering and trajectory tracking maneuvers. Here, R represents the 3-2-1 rotation matrix. (b) The plot displays five different hovering trajectories, showing the robot maintaining a stable attitude and remaining close to the starting position. The mean RMS error and standard deviation for the trajectories are 4.17± 0.37 cm.

5.3.6 Hovering

The initial task was to hover around a desired position. We did five such flights, each lasting 2 seconds, and the robot managed to hover using the pre-calculated LQR gain, with RMS errors of 4.2 cm, 3.8 cm, 4.05 cm, 4.03 cm, and 4.8 cm. For hovering tasks, a PID controller, as referenced in [74] and [48], outperforms this with an average RMS error of 2 cm. We think this is due to the fact that PID controllers are manually tuned for specific tasks, whereas the LQR controller used here was only tuned once for determining the Q and R matrices and is more general, as it can be linearized about different states. Five hover trajectories are shown in Fig. 5.5(b). Trajectory data of a hovering flight is provided in the Supplementary Materials (Supplementary Data-hovering).



(a) (b) (c)

Figure 5.6: Results of LQR implementation on the RoboFly: (a) Response to the external disturbance applied via kevlar thread is shown in the photo composite. The robot does a recovery maneuver to get back to the stable attitude and ultimately flies to the desired position.(b) The photo composite shows the RoboFly tracking a 10 cm radius circular trajectory over a 4.5-second flight using a constant pre-calculated LQR gain. (c) The desired trajectory, depicted in green, was provided to the controller in the form of position and velocity set points while magenta is the actual trajectory followed by the robot. The RMS error for this maneuver for x-y position tracking is 2.8 cm. The video link of the experiments is shown in the abstract page.

5.3.7 Response to External Disturbance

To prevent crashes during our experiments, we suspend the robot using a lightweight Kevlar thread. There is slack in the Kevlar thread in all our experiment videos, which demonstrates that the thread does not exert any force on the robot. However, in this particular experiment, we intentionally applied force by pulling the robot with the Kevlar thread to test the controller's response to external disturbances. This causes the accelerations $>2.5g$. As show in Fig. 5.6(a), the robot was able to stabilize itself and flies towards the desired position. Trajectory data of this flight is provided in the Supplementary Materials (Supplementary Data-disturbance_rejection).

5.3.8 Trajectory Tracking

Here the robot was asked to follow a pre-computed circular trajectory of 10 cm radius. The robot was able to follow the given trajectory in a 4.5-second flight with an RMS error of 2.8 cm in x-y position tracking. Photo composite of the maneuver is shown in Fig. 5.6(b). The desired waypoints on the trajectory were given in the form of position and velocity set points. The controller used the same Q and R matrices as the hovering maneuver. For comparison, in [107] authors tracked a 5 cm radius circular trajectory with the reported x-y position error of 1.8 cm. Our greater position tracking error is likely due to a much higher flight velocity (25 cm/sec vs a maximum speed of 5.2 cm/s in [107]) and a larger circular trajectory, which would result in larger disturbances by the wire tether. Trajectory data of this flight is provided in the Supplementary Materials (Supplementary Data-trajectory_tracking).

5.4 Discussion

In our study, we developed and validated a theoretical model of the UW RoboFly. Utilizing this model, we successfully implemented an infinite horizon Linear Quadratic Regulator (LQR) control strategy. This enabled us to achieve stable hovering, recovery maneuver, and trajectory tracking using the pre-calculated LQR gain. While its RMS position error was higher than other recent reports, it was following a much faster trajectory. Notably, our controller works with minimal computational demands, making it ideal for integration into microcontrollers suited for tiny robotic platforms.

5.4.1 Limitations and Future Work

This work provides a foundation for some important next steps. First, the single LQR gain could be replaced with one that is “gain-scheduled” for linearizations about different states, such as forward flight or flight into the wind. This would also impose a minimal computational load on any conceivable microcontroller. Second, collecting data over a broader flight envelope could improve model identification and therefore flight control. Third, our current controller does not perform any sort of adaptivity. Future models could account for the offset between the thrust vector and the body-z axis and estimate the drag model using the discrepancy between expected and actual translational velocity as input[74, 48]. Such refinements are expected to yield even lower RMS errors for specific tasks and potentially enable more agile maneuvers.

While gain scheduling has the potential to broaden the envelope in which the LQR controller operates well, it does not take into account an important aspect of FIRs. This is that they are subject to constraints on the magnitude of outputs that the actuators can produce. A more advanced control technique, known as Receding Horizon Control (RHC) or Model Predictive Control (MPC), can accurately factor those into the optimization process. Although [107] applied this technique, their controllers were too demanding for a microcontroller. We expect that the advances that were used in Tiny-MPC [4] to perform RHC on the 120 MHz microcontroller onboard the the 30 g Crazyflie helicopter could be adapted to the different dynamics and constraints of the Robofly. By taking into account actuator constraints and any improved system characterization, more precise and more aggressive agile maneuvers should be possible without hand-tuning.

5.5 Methods

5.5.1 Infinite Horizon LQR

Infinite horizon LQR controller [5] optimizes the quadratic cost function subject to linearized dynamics constraints. In this work we use the continuous time formulation of LQR.

$$J = \int_0^\infty \sigma(t)^T Q \sigma(t) + u(t)^T R u(t) \tag{5.8}$$

$$\text{subject to } \dot{\sigma}(t) = A_d \sigma(t) + B_d u(t) \tag{5.9}$$

Here, $\sigma(t)$ is the state vector of the robot at time t, A_d and B_d are the linearized dynamics matrices. If the system is completely stabilizable, we can write this optimal control problem in terms of the Hamiltonian function, which incorporates both the system dynamics and the cost function. The Hamiltonian for the LQR problem is given by $H(x,u,\lambda) = \sigma^T Q \sigma + u^T R u + \lambda^T (A_d \sigma + B_d u)$. λ is the costate vector, Q and R are positive semi-definite and positive definite matrices respectively. By taking the derivative of Hamiltonian with respect to σ, u, and λ and setting it to zero we get the closed loop dynamics as,

$$\begin{bmatrix} \dot{\sigma} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} \sigma \\ \lambda \end{bmatrix} \tag{5.10}$$

The steady state solution of the Riccati equation can be described in terms of the eigenvectors of the Hamiltonian matrix in the closed loop dynamic equation 5.10 associated with the negative real part eigen values. These negative real part eigenvalues of the matrix are also the eigenvalues of the closed-loop matrix $A - BR^{-1}BP$. The feedback gain $k = -R^{-1}B'P$ can be obtained using P as the Schur form of the closed loop dynamics matrix such that:

$$u(t) = -k(\sigma_{des}(t) - \sigma(t)) \tag{5.11}$$

Here $\sigma_{des}(t)$ is the desired state of the robot at time t.

Chapter 6

SPARSE MODEL IDENTIFICATION OF TRANSLATIONAL FORCES FOR IMPROVED FLIGHT CONTROL OF FLAPPING WING ROBOTS

The text of this chapter is a draft of a journal paper. It uses discrepancy modeling via sparse identification of nonlinear dynamics (SINDy) algorithm to learn stroke-averaged translational models of two robots using their free-flight data. By integrating this model into a Linear-Quadratic Regulator (LQR) controller, we demonstrated substantial improvements in hovering precision and trajectory tracking.

Dhingra, D., Kaheman, K. and Fuller, S. Sparse Model Identification of Translational Forces for Improved Flight Control of Flapping Wing Robots, IEEE Transactions on Robotics (T-RO) 2024 (Draft prepared).

6.1 Abstract

High-speed maneuvering enables flapping-wing robots to explore more quickly with limited onboard power. However, these maneuvers are often not possible on these robots due to complicated unsteady flapping wing aerodynamics, manufacturing uncertainties, and rapid wear. In this work, we use discrepancy modeling via sparse identification of nonlinear dynamics algorithm (SINDy) to learn stroke-averaged translational models of two robots using their free-flight data. Besides drag coefficients, we found that 20% of the discrepancy in the longitudinal direction is due to input pitch torque, and 30% of the discrepancy in the lateral direction is due to input roll torque. We use this compensated model to achieve precise controlled maneuvers using Linear-Quadratic Regulator (LQR). Our results show a 49% increase in hovering precision and a 21% increase in trajectory tracking precision while achieving speeds as high as 26.5 cm/sec. This work can enable high speed time optimal maneuvers on insect-sized flapping wing robots.

6.2 Introduction

Biological insects have developed their flight capabilities through millions of years of evolution, enabling them to execute complex maneuvers such as carrying food, capturing prey, and evading predators. Flapping-wing insect-sized robots (FIRs) draw inspiration from these natural fliers. They are small, inexpensive to manufacture, capable of rapid maneuvers, and are safer than bigger drone to operate around humans [46] making them attractive for applications like search and rescue missions in inaccessible areas, gas leak detection, crop inspections, and potentially planetary exploration.

Success of these applications hinges on how quickly the tiny robot can reach a desired position. High speed also contributes to more power-efficient flights [54]. Currently, FIRs are powered by wire tethers, but recent research [62, 59] aims to make them wireless. However, the flying time so far is less than 1 second. As research in power autonomy progresses, exploring time-efficient controls becomes an important gap to address. Designing a precise controller for these robots that can achieve high maneuvering speeds is a challenging task because, 1) as flying speeds increase, aerodynamic effects become more significant for achieving time-optimal flight [51], 2) flapping wings at the insect scale exhibit highly nonlinear behavior characterized by unsteady aerodynamics [98], and 3) FIRs has limited onboard computing power, and suffer from rapid wear of components, so the control algorithms also needs to be robust and resource efficient [33].

In our previous work [37], we developed and validated a first-principles model of the Robofly using measured thrust and torque mappings [114]. In this paper, we enhance this model by performing discrepancy modeling via SINDy approach using high-speed flight data. To promote generalization in the model the discrepancy terms are selected by using group sparsity analysis on two different robots. The new improved model with discrepancy terms not only reduced model prediction errors but also improved controller precision. Our work makes four major contributions:

- We developed aerodynamic models using free flight data for longitudinal and lateral flights.
- We established relationships between torque and translational speeds.
- We performed precise flight maneuvers, such as hovering and trajectory tracking using the

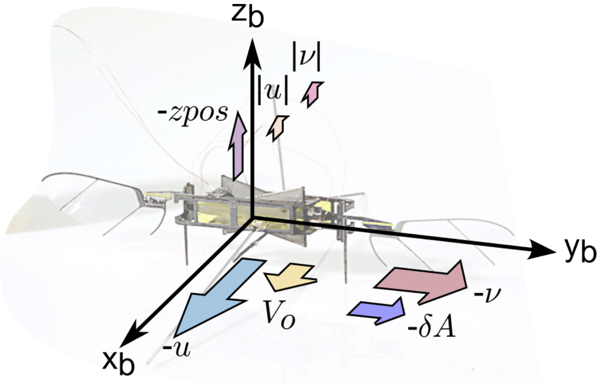


Figure 6.1: Robofly, a 150 mg insect-sized robot used in this study, is shown with translational forces identified from free-flight data. The arrow sizes are proportional to the force magnitudes. Here, V_o and δA are input voltages causing pitch and roll torques, respectively, u and v represent body velocities in the x_b and y_b directions, and $zpos$ is the robot's altitude.

compensated model, along with an LQR controller, resulting in improved prediction and controller performance.

- We open-sourced our flight data so it can benefit the modeling community. We hope this dataset can serve as a valuable resource for testing different system identification algorithms' performance on insect size robots.

We utilized real flight data from two Roboflies, spanning 52 seconds across 42 distinct flights. This dataset encompasses the stable flight envelope, with body angular velocities reaching up to 543° per second and linear velocities up to 0.56 m/s.Inspired by insect intelligence [33], we build a parsimonious model that can will easy to compute forward in time and can be used onboard the resource constrained insect-sized robots. We choose SINDy algorithm to identify the unmodeled terms instead of other modeling approaches such as Guassian process [104], neural networks [71], as SINDy algorithm provides us an interpretable and parsimonious model that can be directly used with conventional controller, such as LQR. Moreover, as SINDy algorithm provides us the analytical form of the unmodeled terms, it further facilitates us to analyze the missing dynamics that is not captured by the first-principled model. As opposed to the models used in previous works [22,

Parameter	**Value**
m	150×10^{-6} kg
J	diag($[3.12 \times 10^{-9}, 2.97 \times 10^{-9}, 0.55 \times 10^{-9}]$) kg.m^2
Γ	$3.27 \times 10^{-5} A - 0.0024$ N
τ_r	$0.48 \times 10^{-6} \delta A$ Nm
τ_p	$0.11 \times 10^{-6} V_o$ Nm

Table 6.1: Measured parameters of the RoboFly. Calculated Moment of inertia J is a diagonal matrix of moment of inertia about the principal axes. Mapping from amplitude (A) to thrust (Γ), differential amplitude (δA) to roll torque (τ_r), and offset (V_o) to pitch torque (τ_p) are the equations of the linear fit.

51, 105], our findings reveal that unmodeled dynamics does not solely depend on body velocities and has insignificant dependency on the angular velocities. We also found that the dependency of torques on the translational speeds plays an important part in accounting for the discrepancy. In addition, we learned the affect of wire tether used to power the robot in the dynamic model. In our work, we only model the discrepancy in translational dynamics and not rotational dynamics due to two key considerations: Firstly, the actuation delay for the rotational system is minimal, as rotational acceleration occurs almost instantaneously once torque is applied. Secondly, the robot will eventually have a gyroscope onboard which is capable of providing rapid rotational velocity feedback at 1 Khz to 16 kHz, which will significantly enhance the ability to perform rapid feedback corrections. Thus, any discrepancy present in the rotational model can be corrected with minimal delay using fast feedback control.

While the model developed in this work is specific to Robofly [29], same principles can be applied to learn stroke averaged dynamics using discrepancy data on other designs of insect-sized flapping-wing platforms like the four-wing four-actuator Bee++ [7], aerial robots powered by dielectric elastomer actuators[17], the RoboBee [74] and the RoboBee X-Wing [59].

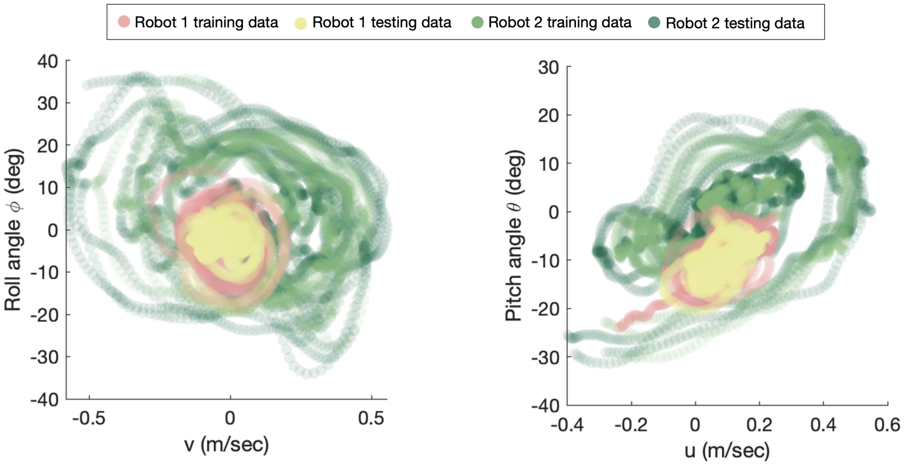


Figure 6.2: Data visualization: The flight envelope of two robots from 42 different flight trajectories of Robofly, sampled at 250 Hz. To capture dynamics from different flight regimes, robot 1 includes more data near hover, while robot 2 includes data with higher attitude angles and translational velocities.

6.3 Robot Description and Dynamics

In our study, we analyzed flight trajectory data from the University of Washington's Robofly, a piezo-actuated flapping wing robot. Robofly, detailed in [29], boasts a 0.045 m wingspan and a weight of $150e^{-6}$ kg (shown in Fig. 6.1). The robot features two piezoelectric actuators as muscles to flap its wings. These actuators are linked to a transmission mechanism that amplifies the actuators' displacement of approximately 200 μm to a wing motion of about 60°. RoboFly has the capability to carry a payload up to 1.5 times its own weight and is powered via a wire-tether, which comprises four wires transmitting signals to operate the actuators.

RoboFly, like other piezo-actuated flapping wing robots [74][118], is operated by low-power 180 volt peak-to-peak sinusoidal signals. It can generate roll torque, pitch torque, and thrust almost independently [114]. The analog voltage signal is generated using the equation,

$$V_{signal} = \left(2A + \frac{\delta A}{2}\right) \sin \omega t + \frac{V_{bias}}{2} + \frac{V_o}{2} \tag{6.1}$$

Here, V_{bias} is a constant bias signal voltage of 250 volt supplied to the top layer of the actuator.

Increasing the amplitude (A) of the sinusoidal signal increases the wing flapping amplitude, thereby generating greater thrust. Creating an amplitude differential (δA) between the wings increases thrust on one side while decreasing it on the other, which produces roll torque. Pitch torque is generated by adjusting the wing flapping either forward or backward relative to the robot's body through a voltage offset (V_o) applied to the sinusoidal signal.

6.3.1 Newton-Euler equations

We use the same first principle model as presented in [37]. Inertial frame is represented by (X,Y,Z) and the body frame is represted by (x_b, y_b, z_b). Body model is written as the first order differential equations of the body velocity, $V_b = [u, v, w]^T$, and the angular velocity, $\omega_b = [p, q, r]^T$, of the robot.

$$\dot{u} = g\sin\theta + a_{d_1} - (qw - rv) \tag{6.2}$$

$$\dot{v} = -g\cos\theta\sin\phi + a_{d_2} - (rv - pw) \tag{6.3}$$

$$\dot{w} = -g\cos\theta\cos\phi + a_{d_3} - (pv - qu) + \frac{\Gamma}{m} \tag{6.4}$$

$$\dot{p} = \alpha_{d_1} + \frac{\tau_r}{J_{xx}} - \frac{J_{zz} - J_{yy}}{J_{xx}} qr \tag{6.5}$$

$$\dot{q} = \alpha_{d_2} + \frac{\tau_p}{J_{yy}} - \frac{J_{xx} - J_{zz}}{J_{yy}} rp \tag{6.6}$$

$$\dot{r} = \alpha_{d_3} - \frac{J_{yy} - J_{xx}}{J_{zz}} pq \tag{6.7}$$

Here, angular rotations are yaw (ψ), pitch (θ) and roll (ϕ). Γ is the thrust applied to the robot in positive z_b direction, $J = diag([J_{xx}, J_{yy}, J_{zz}])$ is the diagonal moment of inertia matrix of the robot and $\tau_c = [\tau_r, \tau_p, 0]^T$ is defined as the input torque vector comprising of roll torque (τ_r) about x_b and pitch torque (τ_p) about y_b. Unmodeled dynamic accelerations are define by $a_d = [a_{d_1}, a_{d_2}, a_{d_3}]^T$ and $\alpha_d = [\alpha_{d_1}, \alpha_{d_2}, \alpha_{d_3}]^T$. These unmodeled accelerations includes stroke averaged aerodynamic accelerations in body frame caused by the drag due to flapping wings.

6.3.2 Model parameters

Parameters of the robot used in equations 6.2 to 6.7 are obtained in the similar way as described in our previous work [37]. These parameters are detailed in the accompanying table 6.1. Mass of the robot was determined using a high-precision scale. Robot's moment of inertia was calculated using a CAD model. The high-precision scale was also used for the calculation of thrust mapping, where the thrust generated by the robot at specific flapping amplitudes (A) was recorded and a least squares fit of the data was calculated to map the thrust with the input flapping amplitude.

For torque mapping, we utilized a device as described in [36]. By applying inputs V_o and δA we generate roll and pitch torques respectively. These torques produced steady state angular deflections on the device, which are linearly correlated with the applied torques. These deflections were accurately measured using the motion capture system, allowing us to map torques effectively.

6.4 Free flight data

Two Roboflies were used to collect 52 seconds of trajectory data with wings flapping at a frequency of 180 Hz, controlled by a PID flight controller [29]. To capture dynamics of different flight regimes, robot 1 recorded 21 trajectories near hover, while robot 2 recorded 21 trajectories with high translational velocities. For robot 2, we set desired points away from the initial position to focus on lateral and longitudinal perturbations. Consequently, trajectories were shortened to the duration required to reach these points.

Each robot was equipped with four retro-reflective markers tracked by a motion capture system (MoCap) with four Prime 13 cameras (OptiTrak, Inc., Salem, OR) running at 250 Hz. While the markers are mounted above the centre of mass of the robot (Fig. 6.1), the pivot point of the robot's body, as described in the MoCap, was moved close to the centre of mass of the robot. Position and quaternion data from the motion capture system were used to calculate V_b, ω_b, and Euler angles offline. Our data includes speeds ranging from near-hover to 0.56 m/s. We split the trajectories into training and testing sets using a 60-40 split for model validation.

Fig. 6.2 shows the free flight data for the two Roboflies. Light green and dark green points represent the training and test data for robot 2, while pink and yellow points represent the training and test data for robot 1.

6.4.1 Body-z offset

As seen in Fig. 6.2 (right), the data for robot 1 is centered around $\approx -10^\circ$. This is due to a misalignment between the body's z-axis (as defined in the MoCap software) and the robot's thrust vector.

The dynamics, described in equations 6.2-6.7, assume perfect alignment of the thrust vector with the body's z-axis, which may not be true in practice. This misalignment occurs because the thrust vector's direction is unknown when defining the robot's body coordinates in the motion capture software. Although less prominent, similar misalignments exist in the rest of the data. A tilted thrust vector introduces lateral and longitudinal forces. To correct this, the lateral and longitudinal forces can be modeled as a constant bias in equations 6.2 and 5.3, assuming the body coordinate system remains consistent throughout the data collection and control process.

6.5 Discrepancy Modeling using SINDy

6.5.1 SINDy Algorithm

The SINDy algorithm is a data-driven algorithm that identifies a parsimonious and interpretable model of the dynamical system [14]. For a dynamical system with state vector

$$\mathbf{x}(t) = [x_1(t),\ x_2(t),\ \dots,\ x_n(t)]^T \in \mathbb{R}^{n\times 1} \tag{6.8}$$

and dynamics

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t), p), \tag{6.9}$$

the SINDy algorithm seeks to identify a sparse representation of the dynamics $f(\mathbf{x}(t))$ such that each state of the system $x_i(t)$ satisfies

$$\frac{d}{dt}x_i(t) = f_i(\mathbf{x}(t), p) \approx \Theta(\mathbf{x}(t))\xi_i, \tag{6.10}$$

where $p \in \mathbb{R}^l$ is the parameter of the system dynamic,

$$\Theta(\mathbf{x}(t)) = [\theta_1(\mathbf{x}(t))\ \theta_2(\mathbf{x}(t))\ \dots\ \theta_q(\mathbf{x}(t))] \tag{6.11}$$

is a library of q candidate functions, and $\xi_i \in \mathbb{R}^{q\times 1}$ is a sparse vector that selects a few terms from the library $\Theta(\mathbf{x}(t))$ to represents the dynamics of state $x_i(t)$ while maintaining the accuracy. Given the measurement of system dynamics

$$\mathbf{X} = [\mathbf{x}(t_1)\ \mathbf{x}(t_2)\ \dots\ \mathbf{x}(t_m)]^T \in \mathbb{R}^{m\times n} \tag{6.12}$$

and its derivative

$$\dot{\mathbf{X}} = [\dot{\mathbf{x}}(t_1)\ \dot{\mathbf{x}}(t_2)\ \dots\ \dot{\mathbf{x}}(t_m)]^T \in \mathbb{R}^{m\times n} \tag{6.13}$$

with m sample points ($m > q$), the value of ξ_i vector can be identified by solving

$$\begin{aligned} \min_{\Xi} \quad & \|\Xi\|_0, \\ \text{s.t.} \quad & \dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi. \end{aligned} \tag{6.14}$$

However, the Eq. (6.14) is NP-hard and can not be solved when the number of library terms q is large. To tackle this, the Brunton et al [14] used sequentially thresholded least squares (STLSQ) algorithm to relax l_0 norm shown in Eq. (6.14). The STLSQ algorithm iteratively solves the least square problem

$$\min_{\Xi} \quad \|\dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi\|_2^2, \tag{6.15}$$

and after the solution of Ξ is obtained, it thresholds the elements in Ξ to zero if its magnitude is smaller than some predefined sparsity promoting parameter λ. The iteration stops when the the value if Ξ is converged, and the convergence study of STLSQ is studied in the work of Zhang et al [120]. Beside STLSQ, many other techniques exist that can solve Eq. (6.14) efficiently, including but not limited to LASSO [113], RIDGE [56], sequentially thresholded ridge regression (STRidge) [97], unified sparse dynamics learning (USDL) [86], sparse relaxed regularized regression (SR3) [122, 16], Bayesian approaches [121, 85], non-convex penalty least-squares algorithm (NPSLS) [73]. It is noteworthy that depends on the strength of sparsity promoting parameter λ, there will be a poll of candidate models that describe the measured dynamics. Thus, a model selection process needs to be performed that strike the balance between interoperability and complexity of the model. Several common ways to select the candidate model includes using Akaike information criteria (AIC) [3, 1, 78], Bayes information criteria (BIC) [103], and cross validation (CV) [8]. In this work, we use STLSQ to solve Eq. (6.14) and use cross validation approach to pick a model that balance the model sparsity and prediction accuracy.

When control input is presented in the dynamical system, the system dynamics becomes

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t), \mathbf{u}(t), p), \tag{6.16}$$

where $\mathbf{u}(t) = [u_1(t),\ u_2(t),\ ...,\ u_r(t)]^T \in \mathbb{R}^{r\times 1}$ represents there are r control inputs to the system. By adding the control input terms in our library function, the system dynamics of the controlled system can be approximated by

$$\frac{d}{dt}x_i(t) = f_i(\mathbf{x}(t), \mathbf{u}(t), p) \approx \Theta(\mathbf{x}(t), \mathbf{u}(t))\xi_i, \tag{6.17}$$

where

$$\Theta(\mathbf{x}(t),\mathbf{u}(t)) = [\theta_1(\mathbf{x}(t),\mathbf{u}(t))\ \theta_2(\mathbf{x}(t),\mathbf{u}(t))\ ...\ \theta_q(\mathbf{x}(t),\mathbf{u}(t))]. \tag{6.18}$$

Thus, by additionally measuring the control input of the system

$$\mathbf{U} = [\mathbf{u}(t_1)\ \mathbf{u}(t_2)\ ...\ \mathbf{u}(t_m)]^T \in \mathbb{R}^{m\times r}, \tag{6.19}$$

the value of ξ_i vector for controlled system can be identified by solving

$$\min_{\Xi} \quad \|\dot{\mathbf{X}} - \Theta(\mathbf{X},\mathbf{U})\Xi\|_2^2, \tag{6.20}$$

using STLSQ. This extension of the SINDy algorithm called SINDy-c [15] allows the identification of parsimonious model for controlled dynamical system.

6.5.2 *Discrepancy modeling*

The physics based first principle modeling approach is a great way to model the dominant motions of system of interest. However, albeit having high interpretability, such modeling approach often requires expert knowledge and careful reasoning. Moreover, the first principled modeling approach often makes many simplification on the actual dynamics while deriving the model. Such simplification will result in discrepancy between the measurement data and model's predicted output. This type of discrepancy can be categorized as model inadequacy. Other types of discrepancy shows up when the parameters of the dynamical model is different from the true underlying value, result in parameter mismatch discrepancy [63, 43]. The discrepancy modeling framework assumes there's a mismatch between the measured dynamics and modeled dynamics such that

$$\frac{d}{dt}\mathbf{x}(t) = \frac{d}{dt}\mathbf{x}_t(t) + g(\mathbf{x}(t),\mathbf{u}(t)), \tag{6.21}$$

where

$$\frac{d}{dt}\mathbf{x}_t(t) = f(\mathbf{x}(t),\mathbf{u}(t),p_t), \tag{6.22}$$

is the output of theoretical model under theoretical model's parameter p_t, and $g(\mathbf{x}(t),\mathbf{u}(t))$ is the dynamics not model by the theoretical model that we want to capture.

Given the measurement data $\mathbf{x}(t)$, the $\frac{d}{dt}\mathbf{x}(t)$ term can be calculated using numerous derivative calculation approach and $\frac{d}{dt}\mathbf{x}_t(t)$ term can be calculated easily by plug in the measured value to the theoretical model $f(\mathbf{x}(t),\mathbf{u}(t),p_t)$. By ignoring the numerical differentiation error and measurement noise, we can approximately have

$$\frac{d}{dt}\delta\mathbf{x}(t) = \frac{d}{dt}\mathbf{x}(t) - \frac{d}{dt}\mathbf{x}_t(t) \approx g(\mathbf{x}(t),\mathbf{u}(t)), \tag{6.23}$$

and the mismatch term $g(\mathbf{x}(t),\mathbf{u}(t))$ for our work is defined by time series data of the vector, $[a_{d_1}, a_{d_2}, a_{d_3}]^T$ from equations 6.2 to 6.7. The mismatch term can be modeled using multiple data-driven approaches such as Neural Networks [71, 41], linear and nonlinear auto-regressive models [2], dynamic mode decomposition (DMD) [69, 93, 101], Gaussian process regression [104, 102], SINDy [14, 15, 64], etc. In this work, we use SINDy to model the mismatch between the FIRs' state derivative data and its theoretical model's output.

6.5.3 *Candidate function selection*

Candidate functions $\Theta(x(t))$ used in our work for modeling the discrepancy are,

$$\Theta(\mathbf{x}(t)) = [V_b, \quad \omega_b, \quad V_b|V_b|, \quad \omega_b|\omega_b|, \quad |\omega_b|, \quad |V_b|, \quad A, \quad \delta A, \quad V_o, \quad 1]^T \tag{6.24}$$

Body velocity, body angular velocity, and their signed quadratic functions are used to model the drag from flapping wings. Absolute velocity functions capture the effects of encountering new airflow, resulting in changes in acceleration. Inputs A, δA and V_o (in volts) capture the direct effects of thrust and torque discrepancies. The unit term in the candidate function is used to learn the bias force caused by the body-z offset referred in section 6.4.1. For the a_{d_3} model, we used the robot's z-position in world coordinates ($zpos$) to calculate the stiffness of the wire tether.

We also tested other quadratic candidate functions, such as the square of body velocities (V_b^2, ω_b^2) and all components from the outer product of body angular and translational velocities ($V_b \otimes \omega_b$) but found no significant improvement in the models.

6.6 **Results**

The results section is structured into three parts: 1) model identification, 2) model validation, and 3) hardware validation. In the model identification section, we visualize the coefficients learned by the SINDy model on the training data for both robots. Subsequently, we employ group sparsity

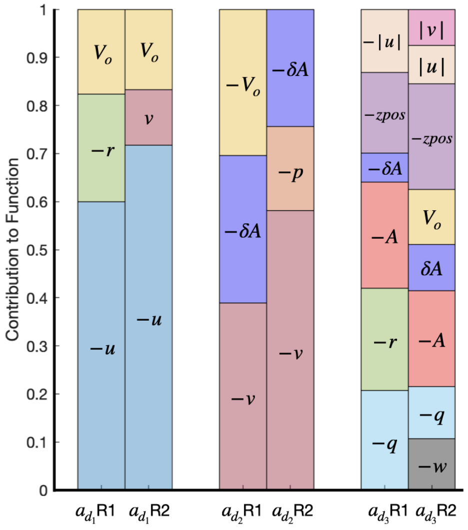


Figure 6.3: Bar plot of the SINDy model coefficients. The coefficients are normalized to sum to 1 to visualize the effect of each coefficient on modeling the discrepancy. This plot does not include any bias term, as the bias changes with every definition of the MoCap body. R1 here refers to the model learned using the data of Robot 1 and R2 is the model learned using the data of Robot 2.

to identify common terms between the two robots to derive generalized final models. The model validation section compares the predicted discrepancy with the measured discrepancy on the test data for both robots. Additionally, it compares translational accelerations using and not using the SINDy model on their respective test data. Finally, in the hardware validation section, an LQR controller based on these final models is employed to evaluate its performance against a controller that does not incorporate any discrepancy model.

6.6.1 Model Identification

Learned coefficients using SINDy for the two robots are represented in the Fig. 6.3. Note that the barplot doesn't show any bias term. As the bias term changes with every definition of robot body in the MoCap software and is not consistent between different robots.

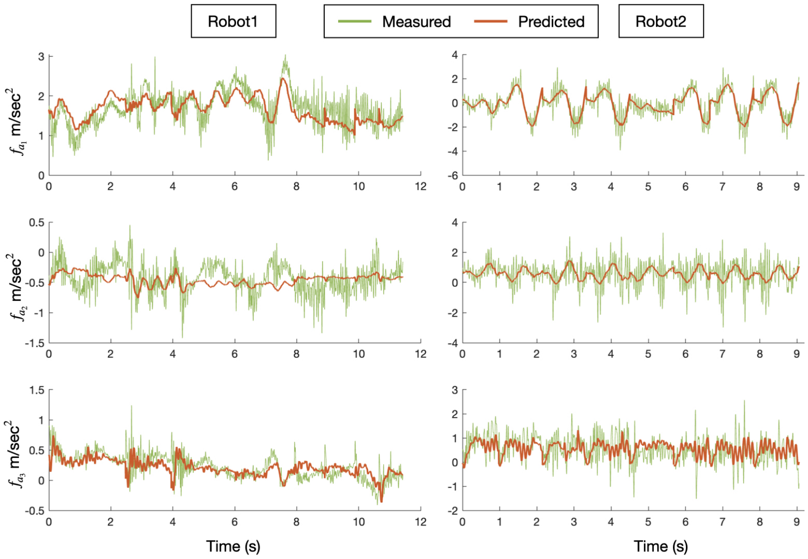


Figure 6.4: Validation plot of SINDy model: The plot shows the discrepancy data (green) from the test trajectories for the two robots and the predictions of the SINDy model (red).

a_{d_1} *model*

Around 60-70% of the discrepancy in the x_b direction can be attributed to the robot's body velocity in that direction, which is expected. In a flapping-wing robot, if the forward and backward strokes have the same velocity, the net drag force can be assumed to be directly proportional to the body velocity since drag force increases with the square of velocity [51]. The model did not learn any discrepancy acceleration caused by the angular velocity around the y_b-axis. This might be attributed to the short moment arm between the robot's center of mass and its aerodynamic center. Additionally, the model demonstrates a clear reliance on the offset voltage that produces the pitch torque. This is likely because each wing's thrust vector is not perfectly vertical but slightly angled inward towards the center of mass. Consequently, when pitch torque is applied and moves the aerodynamic center behind or in front of the center of mass, this inward-pointing thrust vector also generates force in the direction of motion. This represents first of its kind quantification of this discrepancy for flapping robots of this scale in literature. Finally we picked the common candidate functions between the two robots and defined a_{d_1} as a least square fit with two terms $[u, V_o]$. The coefficient determined

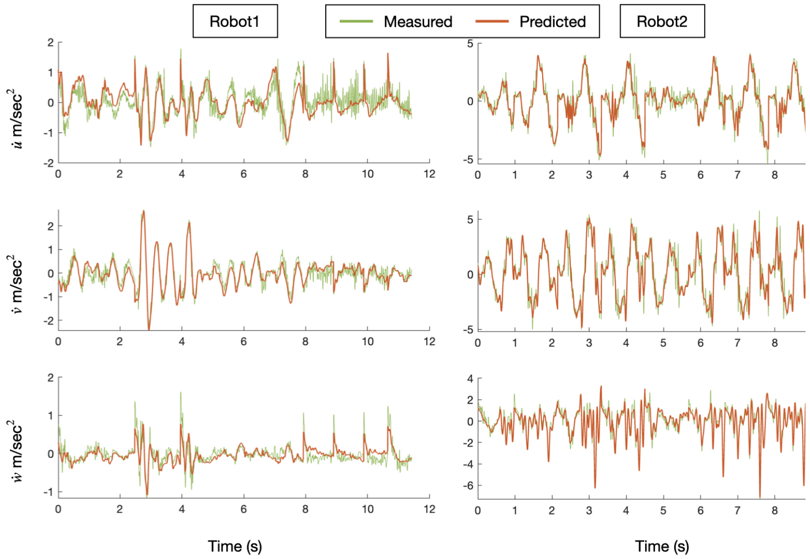


Figure 6.5: Modeled vs predicted accelerations: The plot shows the acceleration data (green) from the test trajectories for the two robots. Shown in red are the predicted accelerations obtained by combining the SINDy model with the robot parametric model.

for body velocity in our experiments for both robots is approximately $-4.2\mathrm{s}^{-1}$, corresponding to a drag coefficient of -7.8×10^{-4} Ns/m. This drag coefficient is roughly four times greater than that reported in [51], which was -2×10^{-4} Ns/m.

a_{d_2} *model*

In analyzing the discrepancy in the body's y-direction, it is evident that 40-60% of the discrepancy is attributed to the robot's body velocity, as anticipated. Similar to the a_{d_1} model, there was no consistent correlation observed with the robot's body angular velocity about the x-axis. Furthermore, the model indicates a distinct 25% discrepancy resulting from differential amplitude between the two wings, δA, which generates roll torque. Therefore, for the final a_{d_2} model, we learned a least square fit with the two common terms between the two robots $[v,\ \delta A]$. The determined drag coefficient for the a_{d_2} model is calculated to be -2.5×10^{-4} Ns/m, closer to that reported in [51].

a_{d_3} *model*

The magnitude of the discrepancy in the body's z direction is smaller compared to the discrepancies in the x and y directions. Nevertheless, certain common terms can be identified between both robots, including q, A, δA, $zpos$. The flapping amplitude (A) term serves as a correction factor for the originally obtained amplitude mapping from the precision scale experiment. The dependency on $zpos$ accounts for tether stiffness. Terms $|u|$ and $|v|$ represent factors that consider an increase in thrust when the robot encounters a new flow. Understandably this dependency was not observed in robot 1 because it has more data close to the hover position. However, no clear explanation for dependency on q and δA exists within this model. For the final model of a_{d_3}, we will utilize a least square fit with the common terms $[q, A, \delta A, zpos]$.

6.6.2 Model Validation

After performing the least square estimation on the selected terms in the previous section, we utilized this final model to predict the discrepancy in the test data of both robots. The graph in Fig. 6.4 illustrates these predictions. It is evident from the graphs that while the models can predict low frequency accelerations, they are unable to predict high frequency discrepancies in the data. One possible reason is that we are using stroke averaged model in this work that is based on stroke averaged body velocities and inputs while this model can predict stroke averaged accelerations, high frequency predictions might require in-stroke dynamics like angle of attack and flapping angle. Fig. 6.5 displays a comparison between the acceleration of the robots in the test data and their predicted accelerations. The predicted accelerations are calculated as a combination of accelerations from both robot parametric model and SINDy model learned in this paper. Furthermore, Table 6.2 provides a list of l_2 norm errors for these predictions and compares them with errors resulting from using only robot's parametric model.

6.6.3 Hardware Validation

The hardware tests are done using a feedback LQR controller with the UW Robofly. The LQR controller uses the discrepancy model learned in this work. A and B matrices are obtained by linearizing the full non-linear dynamics in combination with the discrepancy model.

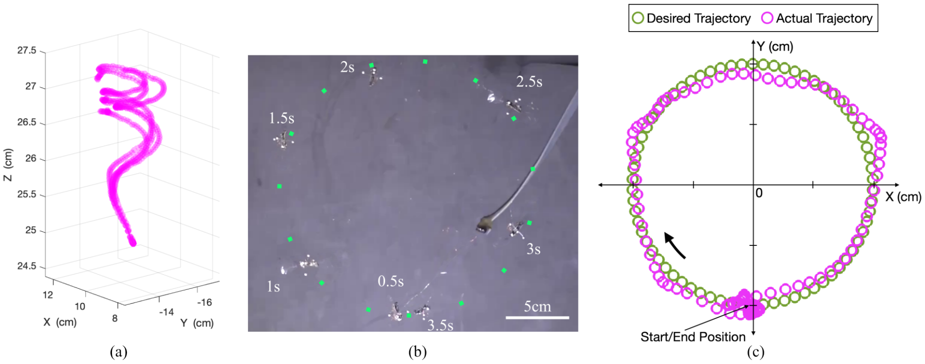
Error Norms	**Robot Paramteric model**		**Robot Paramteric model + SINDy Model**	
	Robot 1	**Robot 2**	**Robot 1**	**Robot 2**
$\dot{u}$ m/sec^2	92.6	51.2	17.4	25.1
$\dot{v}$ m/sec^2	25.9	45.4	12.3	32.2
$\dot{w}$ m/sec^2	16	35.8	8.7	20.6

Table 6.2: L2 norm of the errors between the measured acceleration data and predicted accelerations. Columns 2 and 3 list the errors by using just the parametric model. Column 4 and 5 list the errors of prediction by the SINDy model combined with the parametric model.

Bias correction: The linearized version of the model, however, does not include the bias and *zpos* terms. Therefore, a feed-forward correction is used to adjust the bias term. Although the *zpos* term was utilized for validation graphs in previous sections, it was not employed in the hardware experiments. As previously mentioned, bias terms arise when the thrust vector does not align with the body-z axis as defined in the motion capture system. Assuming that thrust is equal to the weight of the robot, we can use the learned bias term to determine the angular difference between the body-z axis and thrust vector. This angular difference can be directly corrected in the MoCap software by redefining the coordinate frame with calculated angular deflections. In subsequent hardware experiments, we applied this rotational correction.

Hovering

We did five hovering flights, each lasting 3 seconds. The robot was able to hover with RMS errors in position of 1.32 cm, 1.19 cm, 1.345 cm, 1.343 cm, and 1.343 cm. As shown in table 6.3 this outperforms the errors obtained in our previous work [37], which doesn't use any discrepancy model. Our RMS errors are higher than an adaptive PID controller proposed in [23] that has an average error of 0.77 cm. We think this is due to the fact that adaptive PID controller are finely tuned for specific tasks, whereas the LQR controller used here was only tuned once for determining the Q and R matrices and is more general, as it can be linearized about different states. These five hover



(a) (b) (c)

Figure 6.6: Maneuvers performed on Robofly using the SINDy model in an LQR controller: a) five hover trajectories are shown in a 3D plot, b) photo composite of top view of Robofly tracking a circular trajectory, the desired trajectory is shown in green, and c) 2D plot of the desired and actual circular trajectory traced by the robot.

trajectories are shown in Fig. 6.6(a).

Response to external disturbance

Similar to the work in [37] we deliberately pulled the robot with the Kevlar thread to test the controller's response to external disturbances, resulting in accelerations over 2.8g. The robot was able to correct its orientation and moved towards the hovering point once the external force is removed.

Trajectory tracking

Here the robot was asked to follow a pre-computed circular trajectory of 10 cm radius in a 3.5-second flight. Robofly performed that maneuver with an RMS error of 1.58 cm in x-y position tracking. As shown in table 6.3 this outperforms the errors obtained in our previous work [37], which doesn't use any discrepancy model. Photo composite of the maneuver is shown in Fig. 6.6(b). Just like the previous work, the desired waypoints on the trajectory were given in the form of position and velocity set points. The controller used the same Q and R matrices as the hovering maneuver and as used in [37]. For comparison, in [107] authors tracked a 5 cm radius circular trajectory with a higher reported x-y position error of 1.8 cm. Our position tracking error is lower while we achieve

Maneuver	**No discrepancy model with LQR** [37]	**SINDy Model with LQR**
Hovering	2.58± 0.14 cm	1.31± 0.06 cm
Trajectory Tracking	2.02 cm	1.58 cm

Table 6.3: Root means square position error comparison of hardware experiments performed in this work with the previous work.

a much higher flight velocity (26.5 cm/sec vs a maximum speed of 5.2 cm/s in [107]).

6.7 Conclusion and Future Work

In this paper, we introduce a novel approach to enhance the precision and speed of flapping-wing insect-sized robots by incorporating discrepancy modeling and learning stroke-averaged translational models from free-flight data. Our research identified discrepancies in longitudinal and lateral flight dynamics due to translational velocities causing drag forces. We also observed discrepancies due to input pitch and roll torques, which lead to translational accelerations because of the inward-pointing thrust vectors unique to flapping wings. Additionally, we observed an increase in thrust as the robot's translational speed is increased. By integrating these findings into an LQR controller, we demonstrated significant improvements in hovering precision and trajectory tracking.

These advancements highlight the potential for high-speed, time-optimal maneuvers in FIRs. Future improvements could include modeling high-frequency discrepancies and employing more accurate integration techniques like RK4 to reduce high-frequency noise in the discrepancy data. Another direction is to learn the dependency of the damping coefficient on the robot's speed. While we established that the damping coefficient's magnitude changes with speed, future work can explore this relationship for a more accurate model. These improved models could then be used to achieve higher speeds than achieved in this work, with improved precision.

Chapter 7

CONCLUSION AND FUTURE WORK

This thesis presents significant advancements in the development, calibration, and control of insect-sized flapping-wing robots (FIRs). Through five distinct but interconnected projects, we addressed several critical challenges in the fabrication, calibration, control, and performance optimization of these tiny robotic systems.

- Bias Torque Correction Device: We developed a device to automatically correct bias torques in small-sized airborne robots, facilitating high-volume manufacturing of FIRs. The device significantly reduces manual trimming time, providing a predictable and less wear-prone method for ensuring stable take-off flight. Additionally, it can trim robots with misaligned payloads and can be adapted for various FIR designs and thruster types.

- Torque Measurement System: We introduced a system to map control voltages to the resulting roll and pitch torques in flapping-wing robots. This system improves upon previous methods by directly measuring torques, aiding in the better modeling of FIR dynamics and enhancing control performance, especially during aggressive maneuvers.

- Longitudinal Force Actuation: Our research successfully introduced a novel mode of actuation to enhance the maneuverability of FIRs. By generating drag-based longitudinal force and mitigating coupled torques, we demonstrated the first direct actuation of longitudinal thrust on a flapping-wing vehicle in free flight. This advancement opens the door for more agile and biologically inspired robotic flight.

- LQR Control Implementation: We developed and validated a theoretical model of the UW Robofly, implementing an infinite horizon Linear Quadratic Regulator (LQR) control strategy. This enabled stable hovering, recovery maneuvers, and trajectory tracking, demonstrating the controller's potential for integration into microcontrollers on tiny robotic platforms.

- Discrepancy Modeling and Learning: We enhanced the precision and speed of FIRs by incorporating discrepancy modeling and learning stroke-averaged translational models from free-flight data. These improvements led to significant gains in hovering precision and trajectory tracking, addressing challenges posed by unsteady aerodynamics and manufacturing inconsistencies.

7.1 Future Work

Despite the progress made, several areas remain for future exploration and improvement:

- Yaw Trimming : Incorporating yaw trimming could further enhance the calibration process. Adding a vertical-axis flexure could provide comprehensive aerodynamic characterization and improve control performance.

- Accelerometer Integration: Future improvements to the torque measurement system could involve integrating an accelerometer to measure angles, simplifying the device's use. Addressing the noise caused by wing vibrations through vibration attenuation techniques could make this approach feasible.

- Maximize longitudinal thrust: Updating wing and transmission designs to maximize longitudinal thrust would be useful for enabling greater agility of the robot. Leveraging the low-latency capability of onboard gyroscopes to directly compensate for coupled torques could eliminate the need for prior calibration.

- Adaptive and Gain-Scheduled Controllers: Replacing the single LQR gain with a gain-scheduled approach for different flight states could improve control across a broader flight envelope. Incorporating adaptive control models to account for thrust vector offsets could further enhance flight precision and agility.

- Receding Horizon Control (RHC) / Model Predictive Control (MPC): Implementing RHC or MPC techniques could accurately factor in actuator constraints. Using RHC can also enable more precise and aggressive maneuvers by combining longitudinal actuation model with pitch torque. Adapting advancements from Tiny-MPC for the Robofly could yield substantial improvements in controller performance.

By addressing these future work areas, we can bring robotic flapping-wing systems closer to the performance of their biological counterparts, paving the way for more agile, precise, and efficient FIRs.

BIBLIOGRAPHY

[1] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In Selected papers of hirotugu akaike, pages 199–213. Springer, 1998.

[2] Hirotugu Akaike. Fitting autoregressive models for prediction. Ann Inst Stat Math, 21(1):243–247, 1969.

[3] Hirotugu Akaike. A new look at the statistical model identification. IEEE transactions on automatic control, 19(6):716–723, 1974.

[4] Anoushka Alavilli, Khai Nguyen, Sam Schoedel, Brian Plancher, and Zachary Manchester. TinyMPC: Model-predictive control on resource-constrained microcontrollers. In IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, May. 2024.

[5] B.D.O. Anderson and J.B. Moore. Optimal control: Linear quadratic methods. 2007.

[6] Siuakumar Balasubramanian, Yogesh M Chukewad, Johannes M James, Geoffrey L Barrows, and Sawyer B Fuller. An insect-sized robot that uses a custom-built onboard camera and a neural network to classify and respond to visual input. In 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), pages 1297–1302. IEEE, 2018.

[7] Ryan M. Bena, Xiufeng Yang, Ariel A. Calderón, and Néstor O. Pérez-Arancibia. High-performance six-DOF flight control of the Bee^{++}: An inclined-stroke-plane approach. IEEE Transactions on Robotics, 39(2):1668–1684, 2023.

[8] Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[9] Ayodeji T Bode-Oke, Samane Zeyghami, and Haibo Dong. Flying in reverse: kinematics and aerodynamics of a dragonfly in backward free flight. Journal of The Royal Society Interface, 15(143):20180102, 2018.

[10] Lorenzo Boninsegna, Feliks Nüske, and Cecilia Clementi. Sparse learning of stochastic dynamical equations. The Journal of Chemical Physics, 148(24):241723, 2018.

[11] S. Bouabdallah, A. Noth, and R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), volume 3, pages 2451–2456 vol.3, 2004.

[12] Tammaso Bresciani. Modelling, identification and control of a quadrotor helicopter. MSc Theses, 2008.

[13] Dario Brescianini and Raffaello D'Andrea. Design, modeling and control of an omni-directional aerial vehicle. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 3261–3266, 2016.

[14] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the national academy of sciences, 113(15):3932–3937, 2016.

[15] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Sparse identification of nonlinear dynamics with control (SINDYc). IFAC-PapersOnLine, 49(18):710–715, 2016.

[16] Kathleen Champion, Aleksandr Zheng, Peng Aravkin, Steven Brunton, and J Nathan Kutz. A unified sparse optimization framework to learn parsimonious physics-informed models from data. arxiv, 0:1906.10612v1, 2019.

[17] YuFeng Chen, Siyi Xu, Zhijian Ren, and Pakpong Chirarattananon. Collision resilient insect-scale soft-actuated aerial robots with high agility. IEEE Transactions on Robotics, 37(5):1752–1764, 2021.

[18] Yufeng Chen, Huichan Zhao, Jie Mao, Pakpong Chirarattananon, E Farrell Helbling, Nak-seung Patrick Hyun, David R Clarke, and Robert J Wood. Controlled flight of a microrobot powered by soft artificial muscles. Nature, 575(7782):324–329, 2019.

[19] B. Cheng, S.N. Fry, Q. Huang, W.B. Dickson, M.H. Dickinson, and X. Deng. Turning dynamics and passive damping in flapping flight. In 2009 IEEE International Conference on Robotics and Automation, pages 1889–1896, 2009.

[20] Bo Cheng, Xinyan Deng, and Tyson L Hedrick. The mechanics and control of pitching manoeuvres in a freely flying hawkmoth (manduca sexta). Journal of Experimental Biology, 214(24):4092–4106, 2011.

[21] Pakpong Chirarattananon, Yufeng Chen, E Farrell Helbling, Kevin Y Ma, Richard Cheng, and Robert J Wood. Dynamics and flight control of a flapping-wing robotic insect in the presence of wind gusts. Interface focus, 7(1):20160080, 2017.

[22] Pakpong Chirarattananon, Kevin Y. Ma, Richard Cheng, and Robert J. Wood. Wind disturbance rejection for an insect-scale flapping-wing robot. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 60–67, 2015.

[23] Pakpong Chirarattananon, Kevin Y Ma, and Robert J Wood. Adaptive control for take-off, hovering, and landing of a robotic fly. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3808–3815. IEEE, 2013.

[24] Pakpong Chirarattananon, Kevin Y Ma, and Robert J Wood. Adaptive control of a millimeter-scale flapping-wing robot. Bioinspiration & biomimetics, 9(2):025004, 2014.

[25] Pakpong Chirarattananon, Kevin Y. Ma, and Robert J. Wood. Fly on the wall. In 5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, pages 1001–1008, 2014.

[26] Pakpong Chirarattananon, Kevin Y Ma, and Robert J Wood. Perching with a robotic insect using adaptive tracking control and iterative learning control. The International Journal of Robotics Research, 35(10):1185–1206, 2016.

[27] Pakpong Chirarattananon and Robert J Wood. Identification of flight aerodynamics for flapping-wing microrobots. In 2013 IEEE International Conference on Robotics and Automation, pages 1389–1396, 2013.

[28] Yogesh M. Chukewad and Sawyer Fuller. Yaw control of a hovering flapping-wing aerial vehicle with a passive wing hinge. IEEE Robotics and Automation Letters, 6(2):1864–1871, 2021.

[29] Yogesh M. Chukewad, Johannes James, Avinash Singh, and Sawyer Fuller. Robofly: An insect-sized robot with simplified fabrication that is capable of flight, ground, and water surface locomotion. IEEE Transactions on Robotics, 37(6):2025–2040, 2021.

[30] Yogesh M Chukewad, Avinash T Singh, Johannes M James, and Sawyer B Fuller. A new robot fly design that is easier to fabricate and capable of flight and ground locomotion. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4875–4882. IEEE, 2018.

[31] Taylor S Clawson, Silvia Ferrari, E Farrell Helbling, Robert J Wood, Bo Fu, Andy Ruina, and Z Jane Wang. Full flight envelope and trim map of flapping-wing micro aerial vehicles. Journal of Guidance, Control, and Dynamics, 43(12):2218–2236, 2020.

[32] Avik De, Rebecca McGill, and Robert J Wood. An efficient, modular controller for flapping flight composing model-based and model-free components. The International Journal of Robotics Research, 41(4):441–457, 2022.

[33] G. C. H. E. de Croon, J. J. G. Dupeyroux, S. B. Fuller, and J. A. R. Marshall. Insect-inspired AI for autonomous robots. Science Robotics, 7(67):eabl6334, 2022.

[34] G C H E de Croon, M A Groen, C De Wagter, B Remes, R Ruijsink, and B W van Oudheusden. Design, aerodynamics and autonomy of the DelFly. Bioinspiration & Biomimetics, 7(2):025003, may 2012.

[35] Daksh Dhaingra and Sawyer Fuller. Longitudinal actuation without tilt in flapping-wing robots. [under prepration].

[36] Daksh Dhingra, Yogesh M. Chukewad, and Sawyer B. Fuller. A device for rapid, automated trimming of insect-sized flying robots. IEEE Robotics and Automation Letters, 5(2):1373–1380, 2020.

[37] Daksh Dhingra, Kadierdan Kaheman, and Sawyer B. Fuller. Modeling and LQR control of insect sized flapping wing robot, 2024.

[38] Michael H Dickinson, Fritz-Olaf Lehmann, and Sanjay P Sane. Wing rotation and the aerodynamic basis of insect flight. Science, 284(5422):1954–1960, 1999.

[39] William B Dickson and Michael H Dickinson. The effect of advance ratio on the aerodynamics of revolving wings. Journal of Experimental Biology, 207(24):4269–4281, 2004.

[40] David B. Doman, Michael W. Oppenheimer, and David O. Sigthorsson. Wingbeat shape modulation for flapping-wing micro-air-vehicle control during hover. Journal of Guidance, Control, and Dynamics, 33(3):724–739, 2010.

[41] Andreas Draeger, Sebastian Engell, and Horst Ranke. Model predictive control using neural networks. IEEE Control Systems Magazine, 15(5):61–66, 1995.

[42] Daniel S Drew, Nathan O Lambert, Craig B Schindler, and Kristofer SJ Pister. Toward controlled flight of the ionocraft: a flying microrobot using electrohydrodynamic thrust with onboard sensing and no moving parts. IEEE Robotics and Automation Letters, 3(4):2807–2813, 2018.

[43] Megan R Ebers, Katherine M Steele, and J Nathan Kutz. Discrepancy modeling framework: Learning missing physics, modeling systematic residuals, and disambiguating between deterministic and random effects. arXiv preprint arXiv:2203.05164, 2022.

[44] Tobias Englert, Andreas Völz, Felix Mesmer, Sönke Rhein, and Knut Graichen. A software framework for embedded nonlinear model predictive control using a gradient-based augmented lagrangian approach (grampc), 2018.

[45] Benjamin M. Finio, Néstor O. Pérez-Arancibia, and Robert J. Wood. System identification and linear time-invariant modeling of an insect-sized flapping-wing micro air vehicle. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1107–1114, 2011.

[46] Dario Floreano and Robert J. Wood. Science, technology and the future of small autonomous drones. Nature, pages 460–466, 2015.

[47] Sawyer Fuller, Zhitao Yu, and Yash P. Talwekar. A gyroscope-free visual-inertial flight control and wind sensing system for 10-mg robots. Science Robotics, 7(72):eabq8184, 2022.

[48] Sawyer B Fuller. Four wings: An insect-sized aerial robot with steering ability and payload capacity for autonomy. IEEE Robotics and Automation Letters, 4(2):570–577, 2019.

[49] Sawyer B Fuller, E Farrell Helbling, Pakpong Chirarattananon, and Robert J Wood. Using a mems gyroscope to stabilize the attitude of a fly-sized hovering robot. In MAV 2014: Int. Micro Air Vehicle Conf.and Competition 2014, Delft, The Netherlands. Citeseer, 2014.

[50] Sawyer B Fuller, Michael Karpelson, Andrea Censi, Kevin Y Ma, and Robert J Wood. Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli. Journal of The Royal Society Interface, 11(97):20140281, 2014.

[51] Sawyer B Fuller, Zhi Ern Teoh, Pakpong Chirarattananon, Néstor O Pérez-Arancibia, Jack Greenberg, and Robert J Wood. Stabilizing air dampers for hovering aerial robotics: Design, insect-scale flight tests, and scaling. Autonomous Robots, 41:1555–1573, 2017.

[52] M. A. Graule, P. Chirarattananon, S. B. Fuller, N. T. Jafferis, K. Y. Ma, M. Spenko, R. Kornbluh, and R. J. Wood. Perching and takeoff of a robotic insect on overhangs using switchable electrostatic adhesion. Science, 352(6288):978–982, 2016.

[53] Nick Gravish and Robert J Wood. Anomalous yaw torque generation from passively pitching wings. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 3282–3287. IEEE, 2016.

[54] Drew Hanover, Antonio Loquercio, Leonard Bauersfeld, Angel Romero, Robert Penicka, Yunlong Song, Giovanni Cioffi, Elia Kaufmann, and Davide Scaramuzza. Autonomous drone racing: A survey. IEEE Transactions on Robotics, 40:3044–3067, 2024.

[55] T. L. Hedrick and T. L. Daniel. Flight control in the hawkmoth Manduca sexta: the inverse problem of hovering. Journal of Experimental Biology, 209(16):3114–3130, 08 2006.

[56] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970.

[57] Aaron M Hoover, Erik Steltz, and Ronald S Fearing. Roach: An autonomous 2.4 g crawling hexapod robot. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 26–33. IEEE, 2008.

[58] Vikram Iyer, Ali Najafi, Johannes James, Sawyer Fuller, and Shyamnath Gollakota. Wireless steerable vision for live insects and insect-scale robots. Science Robotics, 5(44):eabb0839, 2020.

[59] Noah Jafferis, E. Helbling, Michael Karpelson, and Robert Wood. Untethered flight of an insect-sized flapping-wing microscale aerial vehicle. Nature, 570:491–495, 06 2019.

[60] Noah T Jafferis, Michael J Smith, and Robert J Wood. Design and manufacturing rules for maximizing the performance of polycrystalline piezoelectric bending actuators. Smart Materials and Structures, 24(6):065023, may 2015.

[61] Johannes James and Sawyer B Fuller. A high-voltage power electronics unit for flying insect robots that can modulate wing thrust. In 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021 [accepted].

[62] Johannes James, Vikram Iyer, Yogesh Chukewad, Shyamnath Gollakota, and Sawyer B Fuller. Liftoff of a 190 mg laser-powered aerial vehicle: The lightest wireless robot to fly. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1–8. IEEE, 2018.

[63] Kadierdan Kaheman, Eurika Kaiser, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. Learning discrepancy models from experimental data. In 58th IEEE Conference on Decision and Control, pages 7389–7396. IEEE, 2019.

[64] Kadierdan Kaheman, J Nathan Kutz, and Steven L Brunton. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. Proceedings of the Royal Society A, 476(2242):20200279, 2020.

[65] Matěj Karásek, Florian T. Muijres, Christophe De Wagter, Bart D. W. Remes, and Guido C. H. E. de Croon. A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns. Science, 361(6407):1089–1094, 2018.

[66] Matthew Keennon, Karl Klingebiel, and Henry Won. Development of the nano hummingbird: A tailless flapping wing micro air vehicle. In 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, page 588, 2012.

[67] Vijay Kumar and Nathan Michael. Opportunities and challenges with autonomous micro aerial vehicles. The International Journal of Robotics Research, 31(11):1279–1291, 2012.

[68] J Nathan Kutz. Data-driven modeling & scientific computation: Methods for complex systems & big data. Oxford University Press, 2013.

[69] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. Dynamic mode decomposition: data-driven modeling of complex systems. SIAM, 2016.

[70] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. Optimal control. 2012.

[71] Richard Lippmann. An introduction to computing with neural nets. IEEE Assp magazine, 4(2):4–22, 1987.

[72] Lennart Ljung. System Identification: Theory for the User. Prentice-Hall, Inc., USA, 1986.

[73] Yisha Lu, Wei Xu, Yiyu Jiao, and Minjuan Yuan. Sparse identification of nonlinear dynamical systems via non-convex penalty least squares. Chaos: An Interdisciplinary Journal of Nonlinear Science, 32(2):023113, 2022.

[74] Kevin Y Ma, Pakpong Chirarattananon, Sawyer B Fuller, and Robert J Wood. Controlled flight of a biologically inspired, insect-scale robot. Science, 340(6132):603–607, 2013.

[75] Kevin Y Ma, Samuel M Felton, and Robert J Wood. Design, fabrication, and modeling of the split actuator microrobotic bee. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1133–1140. IEEE, 2012.

[76] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robotics & Automation Magazine, 19(3):20–32, 2012.

[77] Ronit Malka, Alexis Lussier Desbiens, Yufeng Chen, and Robert J Wood. Principles of microscale flexure hinge design for enhanced endurance. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2879–2885. IEEE, 2014.

[78] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor. Model selection for dynamical systems via sparse regression and information criteria. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, 473(2204):20170009, Aug 2017.

[79] Mark B Milam, Ryan Franz, John E Hauser, and Richard M Murray. Receding horizon control of vectored thrust flight experiment. IEE Proceedings-Control Theory and Applications, 152(3):340–348, 2005.

[80] Robert C Nelson et al. Flight stability and automatic control, volume 2. WCB/McGraw Hill New York, 1998.

[81] Michael W. Oppenheimer, Isaac E. Weintraub, David O. Sigthorsson, and David B. Doman. Control of a minimally actuated biomimetic vehicle using quarter-cycle wingbeat modulation. Journal of Guidance, Control, and Dynamics, 38(7):1187–1196, 2015.

[82] Christopher T Orlowski and Anouck R Girard. Dynamics, stability, and control analyses of flapping wing micro-air vehicles. Progress in Aerospace Sciences, 51:18–30, 2012.

[83] Gareth D Padfield. Helicopter flight dynamics. Wiley Online Library, 2008.

[84] Daniele Palossi, Antonio Loquercio, Francesco Conti, Eric Flamand, Davide Scaramuzza, and Luca Benini. A 64mw DNN-based visual navigation engine for autonomous nano-drones. IEEE Internet of Things Journal, 2019.

[85] W. Pan, Y. Yuan, J. Gonçalves, and G. Stan. A sparse Bayesian approach to the identification of nonlinear state-space systems. IEEE Transactions on Automatic Control, 61(1):182–187, January 2016.

[86] Yannis Pantazis and Ioannis Tsamardinos. A unified approach for sparse dynamical system inference from temporal measurements. Bioinformatics, 35(18):3387–3396, 2019.

[87] James Paulos, Bennet Caraher, and Mark Yim. Emulating a fully actuated aerial vehicle using two actuators. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 7011–7016, 2018.

[88] Néstor O Pérez-Arancibia, Kevin Y Ma, Kevin C Galloway, Jack D Greenberg, and Robert J Wood. First controlled vertical flight of a biologically inspired microrobot. Bioinspiration & Biomimetics, 6(3):036009, 2011.

[89] Néstor O Pérez-Arancibia, Kevin Y Ma, Kevin C Galloway, Jack D Greenberg, and Robert J Wood. First controlled vertical flight of a biologically inspired microrobot. Bioinspiration & Biomimetics, 6(3):036009, Sep 2011.

[90] David A Peters and Dinesh Barwey. A general theory of rotorcraft trim. Mathematical Problems in Engineering, 2(1):1–34, 1996.

[91] Hoang Vu Phan, Quoc Viet Nguyen, Quang Tri Truong, Tien Van Truong, Hoon Cheol Park, Nam Seo Goo, Doyoung Byun, and Min Jun Kim. Stable vertical takeoff of an insect-mimicking flapping-wing system without guide implementing inherent pitching stability. Journal of Bionic Engineering, 9(4):391–401, 2012.

[92] Hari Krishna Hari Prasad, Ravi Sankar Vaddi, Yogesh M Chukewad, Elma Dedic, Igor Novosselov, and Sawyer B Fuller. A laser-microfabricated electrohydrodynamic thruster for centimeter-scale aerial robots, 2019.

[93] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. SIAM Journal on Applied Dynamical Systems, 15(1):142–161, 2016.

[94] Néstor O. Pérez-Arancibia, Pierre-Emile J. Duhamel, Kevin Y. Ma, and Robert J. Wood. Model-free control of a flapping-wing flying microrobot. In 2013 16th International Conference on Advanced Robotics (ICAR), pages 1–8, 2013.

[95] Leif Ristroph, Attila J Bergou, John Guckenheimer, Z Jane Wang, and Itai Cohen. Paddling mode of forward flight in insects. Physical review letters, 106(17):178103, 2011.

[96] Andrew Roberts and Abdelhamid Tayebi. Adaptive position tracking of VTOL UAVs. IEEE Transactions on Robotics, 27(1):129–142, 2010.

[97] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. Science Advances, 3(4):e1602614, 2017.

[98] Sanjay P. Sane and Michael H. Dickinson. The control of flight force by a flapping wing: lift and drag production. Journal of Experimental Biology, 204(15):2607–2626, 08 2001.

[99] Sanjay P. Sane and Michael H. Dickinson. The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight. Journal of Experimental Biology, 205(8):1087–1096, 04 2002.

[100] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. In Proc. R. Soc. A, volume 473, page 20160446. The Royal Society, 2017.

[101] Peter J Schmid. Dynamic mode decomposition and its variants. Annual Review of Fluid Mechanics, 54:225–254, 2022.

[102] Eric Schulz, Maarten Speekenbrink, and Andreas Krause. A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. Journal of Mathematical Psychology, 85:1–16, 2018.

[103] Gideon Schwarz. Estimating the dimension of a model. Ann. Statist., 6(2):461–464, 03 1978.

[104] Matthias Seeger. Gaussian processes for machine learning. International journal of neural systems, 14(02):69–106, 2004.

[105] Nak seung P. Hyun, Rebecca McGill, Robert J. Wood, and Scott Kuindersma. A new control framework for flapping-wing vehicles based on 3d pendulum dynamics. Automatica, 123:109293, 2021.

[106] Mao Sun and Yan Xiong. Dynamic flight stability of a hovering bumblebee. Journal of experimental biology, 208(3):447–459, 2005.

[107] Andrea Tagliabue, Yi-Hsuan Hsiao, Urban Fasel, J. Nathan Kutz, Steven L. Brunton, YuFeng Chen, and Jonathan P. How. Robust, high-rate trajectory tracking on insect-scale soft-actuated aerial robots with deep-learned tube mpc, 2022.

[108] Yash P. Talwekar, Andrew Adie, Vikram Iyer, and Sawyer B. Fuller. Towards sensor autonomy in sub-gram flying insect robots: A lightweight and power-efficient avionics system. In 2022 International Conference on Robotics and Automation (ICRA), pages 9675–9681, 2022.

[109] GK Taylor and ALR Thomas. Animal flight dynamics II. longitudinal stability in flapping flight. Journal of theoretical biology, 214(3):351–370, 2002.

[110] Z. E. Teoh, S. B. Fuller, P. Chirarattananon, N. O. Préz-Arancibia, J. D. Greenberg, and R. J. Wood. A hovering flapping-wing microrobot with altitude control and passive upright stability. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3209–3216, 2012.

[111] Z. E. Teoh and R. J. Wood. A flapping-wing microrobot with a differential angle-of-attack mechanism. In 2013 IEEE International Conference on Robotics and Automation, pages 1381–1388, 2013.

[112] Zhi Ern Teoh, Sawyer B. Fuller, Pakpong C. Chirarattananon, Néstor O. Pérez-Arancibia, Jack D. Greenberg, and Robert J. Wood. A hovering flapping-wing microrobot with altitude control and passive upright stability. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conf., pages 3209–3216, Vilamoura, Algarve, Portugal, 7–12 October 2012.

[113] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.

[114] Aaron Weber, Daksh Dhingra, and Sawyer B. Fuller. A flexured-gimbal 3-axis force-torque sensor reveals minimal cross-axis coupling in an insect-sized flapping-wing robot, 2024.

[115] J. P. WHITNEY and R. J. WOOD. Aeromechanics of passive rotation in flapping flight. Journal of Fluid Mechanics, 660:197–220, 2010.

[116] RJ Wood, E Steltz, and RS Fearing. Optimal energy density piezoelectric bending actuators. Sensors and Actuators A: Physical, 119(2):476–488, 2005.

[117] Robert J. Wood, Ben Finio, Michael Karpelson, Kevin Ma, Néstor O. Pérez-Arancibia, Pratheev S. Sreetharan, Hiroto Tanaka, and John P. Whitney. Progress on “pico” air vehicles. Int. J. Robotics Research, 31(11):1292–1302, 2012.

[118] Xiufeng Yang, Ying Chen, Longlong Chang, Ariel A. Calderón, and Néstor O. Pérez-Arancibia. Bee+: A 95-mg four-winged insect-scale flying robot driven by twinned unimorph actuators. IEEE Robotics and Automation Letters, 4(4):4270–4277, 2019.

[119] Ronald W Yeung. On the interactions of slender ships in shallow water. Journal of Fluid Mechanics, 85(1):143–159, 1978.

[120] Linan Zhang and Hayden Schaeffer. On the convergence of the sindy algorithm. Multiscale Modeling & Simulation, 17(3):948–972, 2019.

[121] Sheng Zhang and Guang Lin. Robust data-driven discovery of governing physical laws with error bars. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 474(2217):20180305, 2018.

[122] Peng Zheng, Travis Askham, Steven L Brunton, J Nathan Kutz, and Aleksandr Y Aravkin. A unified framework for sparse relaxed regularized regression: SR3. IEEE Access, 7:1404–1423, 2018.