

Data-driven modeling of a flapping bat robot with a single flexible wing surface

Jonathan Hoff* and Seth Hutchinson†

*University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

†Georgia Institute of Technology, Atlanta, GA 30308, USA

Email: jehoff2@illinois.edu; seth@gatech.edu

Abstract—Flapping wing aerial vehicles rely heavily on accurate models for a variety of different tasks. There have been significant efforts in creating both analytical and data-driven models for many of these types of vehicles including ornithopters and small aerial vehicles mimicking insects. However, very few works have explored modeling for aerial vehicles with a skeletal structure throughout the wings and a single flexible membrane that covers the wings and tail such as is found in robots with bat morphology. In this paper, we build upon previous efforts to model a bat robot using a combination of first-principles and data-driven tools. We record a series of load cell tests and free-flight experiments, and we optimize the model parameters to improve long-term flight prediction. We introduce several extra terms in the model including a term explaining the coupling between wings and tail in order to maximize the effectiveness of collected flight data. The result is a model that performs well in prediction for a range of different tail actuator configurations as demonstrated by our flight results using a bat robot.

I. INTRODUCTION

In the field of robotics, creating an accurate model is critical for many applications such as trajectory planning, long-term prediction, control, and state estimation. This is particularly important for flapping systems because their dynamics are highly underactuated. While many systems possess actuators that can shape the dynamics, the path of flight can be largely determined by the initial launch conditions, and even a good controller cannot compensate for certain flight scenarios. Additionally, this is a challenging task because there are unsteady aerodynamic effects due to flapping and flexibility in the wings of these systems. Numerous works have undertaken this task of modeling flapping aerial vehicles, most prominently FWMAVs (flapping-wing micro aerial vehicles). These systems consist of insect-scale fliers [6, 15], ornithopters mimicking birds [8, 17], robotic hummingbirds [13, 27], and FWMAVs inspired from insects [5]. However, far fewer have created models for robots with bat morphology [21]. Biological bats are fundamentally different than these other fliers because they have a more complex skeletal system throughout the wings with both active and passive joints. Where many fliers have a set of wings and a separate tail, bats have a membrane that is stretched across their wings and hindlimbs. Similarly, the bat robot called Bat Bot (B2) developed by [21] has a single membrane that is anchored to the hindlimbs and wings.

Previous works have used analytical models for B2 [18–21], and one has used load cell data to improve model prediction [12]. However, these models are limited in their capability



Fig. 1: Images of B2 (left) used in load cell and flight experiments and a free-flight experiment (right). The reflective markers are attached to the top carbon fiber structure in order to be elevated enough to minimize marker occlusions from the Vicon system.

of long-term flight path prediction (1-second duration for this application), and consequently limited in more extensive use of trajectory optimization and state estimation. In developing a new model, there are two paths to take. One approach is to create a more sophisticated aerodynamic model. However, this comes at the cost of computational complexity and significant time spent designing the model. An alternative is to utilize data in some type of framework to produce a data-driven model, either by creating a new model structure or by optimizing the parameters of the existing structure. In this paper, we consider the second approach of using data to optimize the model parameters along with a few additions to the model structure.

The primary contributions of this paper are summarized by the following. First, we have developed a systematic data-driven procedure that incorporates both load cell data and free-flight data in order to optimize the parameters of an existing model of the bat robot B2. Second, we have made several additions to the existing model structure including a term that couples the wing movement with the tail position in order to capture the properties of the single wing surface and to maximize the use of the collected flight data. The result is a significant improvement in long-term flight prediction for a set of flight tests with a variety of different tail configurations using a minimal number of short flight tests. To the best of the authors’ knowledge, this is the first study to validate the long-term flight prediction of a model for a robot with bat morphology using free-flight experiments data. We provide extensive results to demonstrate its accuracy in long-term prediction, and we show dramatic improvements in flight

predictions for a range of different tail configuration after training with even a single flight test.

Previous work regarding bat robots has addressed the challenge of kinematic design of the robot [10, 11]. The authors took a data-driven approach by using data from a biological bat to optimize the geometric properties of the wings to best match the biological data. Analogously, in this paper we address the challenge of creating an accurate dynamics model of a robotic bat with a data-driven approach. We use data from a robotic bat to optimize parameters of the dynamics model to best match the experimental data. This improves upon the existing model from [12].

Because of the nature of data collection, it is challenging to obtain large amounts of data to cover the full state space of the robot. The state space is particularly large because the oscillations produced from flapping are nontrivial, contributing up to 10° of change in pitch and 2 cm of change in height over one wingbeat period. As a result, we cannot use some of the modern data-driven methods that require an extensive collection of data. For example, Lee et al. [14] modeled the dynamics of a FWMAV with a deep neural network, but the training data set consisted of 30 minutes of flight data. The methodology we have developed maximizes both our knowledge of the robot and the available data to optimize the analytical model's parameters, and as a result, only a handful of load cell tests and 1-second flights are required to achieve excellent prediction results.

The remainder of the paper is organized as follows. Section II discusses previous related works for data-driven models of flapping flight. We describe our additions to the existing model in Section III. The experimental data collection is covered in Section IV, and the parameter estimation routine used to optimize the model with the data is presented in Section V. The prediction results from the parameter estimation are given in Section VI. We conclude with some remarks in Section VII. A supplementary video¹ describes the methods and results of this paper.

II. RELATED WORK

It is both challenging and computationally expensive to achieve high fidelity models of flapping flight, and thus many researchers have considered system identification methods for improving flapping flight models. Time domain system identification methods have been popular in the flapping flight community [1, 4, 7, 9]. These methods typically begin by collecting a time series of acceleration data in the x (forward-backward) and z (upward-downward) directions from either dynamic free flight experiments or static wind tunnel experiments. They use linear least-squares estimation for linear models or gradient-based optimization for nonlinear models to estimate the parameters of the model. Hoburg and Tedrake [9] identified model parameters of a fixed-wing by recording free flight data and running least-squares estimation with the recorded acceleration data. Grauer et al. [7] used similar

methods to identify a linear model for the lift and drag coefficients of an ornithopter. Armanini et al. [1] used a Gauss-Newton algorithm to estimate aerodynamic model parameters of a FWMAV from wind tunnel measurements on a force sensor. Chirarattananon and Wood [4] collected acceleration data of a FWMAV and used least-squares optimization with gradient descent to identify parameters such as aerodynamic coefficients, moments of inertia, center of mass offsets, etc. These methods are considered one-step prediction, because they do not minimize long-term errors from simulating the dynamics forward in time. Consequently, several have refined their initial identification method by running a gradient-based optimization routine to minimize prediction errors between the free-flight data and the model integrated forward in time [4, 9].

Other past works in system identification of flapping flight have taken the approach of using linear model structures [2, 3, 7, 24]. These have shown to be effective in modeling ornithopters and FWMAs. However, each linear model is most effective at a specific operating point, and model prediction becomes less accurate further from this point. Additionally, one of the limitations of purely data-driven approaches is the need for new data for unexplored areas of the state space. Rose et al. [24] effectively modeled an ornithopter for diving using a series of free-flight tests and a family of linear models, but tailoring the model to a new maneuver would require collecting new flight experiments.

On the contrary, our approach could be effective for a range of maneuvers given the physics-based model structure, and it requires few flight experiments to train because much of the relevant information is built into the model. Other works have utilized a similar approach of creating a first-principles model with parameters selected based on real world data. Ritz and D'Andrea identified coefficients of a parametric aerodynamic model for a tailsitter aircraft over a large flight envelope [22]. Peng et al. [16] used nonlinear optimization to identify parameters of a FWMAV model by simulating the model through forward integration and minimizing the difference between the estimated position and orientation of simulation and that collected from data. This is also similar to previously mentioned works of system identification of flapping flight with nonlinear aerodynamic models [1, 4, 14].

Our strategy of using a physics-driven model with parameters tuned using a data-driven method has additional advantages. The model is adaptable to changes to the physical system because the parameters have physical meaning. For example, if the wing span of the robot is increased, instead of collecting a new data set, either the wingspan parameter is increased or only a few data points may be needed to correct for the change. This is a drawback for model structures that lack physical intuition and are purely data-driven because they are more inflexible in adjusting to changes in the physical system without obtaining new data sets.

Another advantage is the ability to perform design optimization for development of future versions of the robot because the model can provide insight into how a robot's design can be improved. For example, Zhang et al. [27] optimized the design

¹<https://youtu.be/mDiCt2Tjneck>

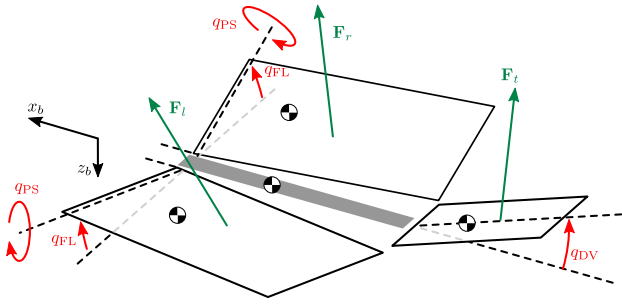


Fig. 2: Planar model of B2 dynamics and aerodynamics. This image was taken from [12].

of a FWMAV using an analytical model in order to improve flight and stability characteristics. A similar approach could be used to maximize lift, thrust, efficiency, agility, and other aspects of B2 in the same way that Hoff et al. [11] optimized a robotic bat for kinematic similarity to biology. This would require an analytical model in which the parameters have some physical meaning.

III. MODELING

We have selected the model of B2 developed previously in [12] as a baseline and starting point for the works in this paper. This model was used successfully in a trajectory optimization framework validated with flight results, thus providing a strong initial framework to build upon. We make several significant improvements to the existing model. First, we have designed a custom coupling function to express the interaction between the tail and wing surfaces. Second, we add an additional aerodynamic surface to represent the part of the membrane that is fixed to the body. Third, we add a nonlinearity for the tail angle to be used in the aerodynamic force calculation. Finally, we use a data-driven approach to optimize the model parameters to improve long-term prediction. This section outlines the 2D longitudinal model from [12], and it presents these new additions to the model. The data-driven approach is presented in Section V. A more in-depth description of the original model can be found in [12].

A. Original Model [12]

B2 is complex to model for several reasons. It propels itself by flapping its wings, and flapping flight on its own is challenging to model because of the unsteady aerodynamic forces generated in flapping. In addition to this, both the wing structure of B2 and the thin silicone membrane stretched across the wings and tail are flexible and contribute passive degrees of freedom (DoF) in the system. Additionally, the mass of B2's wings cannot be ignored because they incorporate roughly 15% of the robot's weight. Given these complexities, this model makes the following modeling choices: 1) B2's wings are modeled as flat plates to ignore flexibility. 2) The two hindlimbs are represented as one plate. 3) The aerodynamic forces on the wings and hindlimbs are considered independently. 4) The aerodynamic center is located at the quarter chord of the wing from thin airfoil theory. 5) The

aerodynamic forces on the flapping wings are assumed to be quasi-steady such that the aerodynamic coefficients are algebraic functions of angle of attack. These assumptions have been made in past models of B2 [18–21].

Figure 2 depicts the model of B2. When the wings are flapping, there is pronation and supination in the wings (twisting of the wings in the spanwise direction) because of wing flexibility. This angle is denoted as q_{PS} . This is an important aspect of the model because this passive twisting contributes the thrust generation to propel B2 in flight. The angle between the xy plane and each wing is the flapping angle, and it is labeled q_{FL} . The two hindlimbs are represented as one flat plate, and the dorsoventral angle (tilting up and down) between this and the xy plane is labeled q_{DV} . These DoFs are actuated by the corresponding torques u_{FL} , u_{PS} , and u_{DV} . q_{PS} is actuated in the model such that it can maintain a periodic trajectory. Both q_{FL} and q_{PS} are controlled to track sinusoidal reference trajectories q_{FL}^r and q_{PS}^r using partial feedback linearization to create periodic flapping.

The center of mass (CoM) of B2 can pitch up and down and translate in the x and z directions. These unactuated DoFs are labeled q_y , p_x , and p_z . The system can be fully described using these coordinates, and thus they are grouped together in the configuration variable vector \mathbf{q} and the inputs in the vector \mathbf{u} :

$$\begin{aligned} \mathbf{q} &= [q_y \ p_x \ p_z \ q_{FL} \ q_{PS} \ q_{DV}]^T \\ \mathbf{u} &= [u_{FL} \ u_{PS} \ u_{DV}]^T. \end{aligned} \quad (1)$$

The Euler-Lagrange formulation is used to derive the equations of motion because this is a multi-body system. These equations are written as

$$D(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = B\mathbf{u} + \mathbf{\Gamma}(\mathbf{q}, \dot{\mathbf{q}}). \quad (2)$$

The aerodynamic forces are mapped to the configuration space and are defined as $\mathbf{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})$. The inputs are mapped to the configuration space with matrix $B = [\mathbf{0}_{3 \times 3} \ \mathbf{I}_{3 \times 3}]^T$. The state space representation of system is then given as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x} = [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T \quad (3)$$

by solving for $\ddot{\mathbf{q}}$ and combining \mathbf{q} and $\dot{\mathbf{q}}$ into the state vector

The aerodynamic coefficients of the model are optimized in the parameter estimation routine in Section V, so we provide a brief formulation here. Hoff et al. [12] uses the model structure

$$\begin{aligned} C_L &= C_{L_1} \sin 2\alpha \\ C_D &= C_{D_0} + C_{D_1} \cos 2\alpha \end{aligned} \quad (4)$$

which is the formulation proposed by Wang [25, 26]. The lift C_L and drag C_D coefficients are functions only of angle of attack α (angle between orientation and velocity direction) and are simple algebraic expressions. The constants C_{L_1} , C_{D_0} , and C_{D_1} will be determined through optimization in Section V.

B. Wing and tail coupling

We have observed an interesting phenomenon in B2 that arises from the coupling between the membrane surfaces of

the wings and tail. The membrane is a single piece of silicone that is stretched across B2. It is anchored to the three digits of each wing, the front of the forelimbs, the hindlimbs, and the top of the body. As a result, movement from the tail affects the state of the wings. When the hindlimbs are in the up position, the membrane starting at the front of the wings gently slopes upward toward the tail. This gives the pronation angle of the wings a downward tilt because the front of the wings is below the rear membrane of the wings. Likewise when the hindlimbs are positioned down, the membrane of the wings slopes downward as it gets closer to the tail. The pronation angle of the wings will have an upward tilt as the front of the wings is above the rear membrane of the wings.

We can incorporate this phenomena in our model by augmenting the reference pronation angle q_{PS}^r to become

$$\bar{q}_{PS}^r(t) = q_{PS}^r(t) + a_{\text{coup}} q_{DV}(t) \quad (5)$$

where a_{coup} is the scaling factor for q_{DV} and \bar{q}_{PS}^r is the new reference pronation angle. The scaling factor must be a negative constant in order for an upward position of the tail to give the pronation angle a downward tilt. When B2 is flying and the hindlimbs move up, the sinusoidal pronation angle reference will be negatively offset from zero and the pronation angle will track this new reference. Likewise, when the hindlimbs are moved down, the pronation angle will track a positively offset reference angle. It is important that we modify the reference angle and not q_{PS} directly because q_{PS} is determined by the control input u_{PS} and cannot be directly modified. The value for the scaling factor a_{coup} is tuned through optimization in Section V.

The implications of this coupling between wings and tail are significant. Future mechanical design of this robot is affected by the amount of coupling between these, and the effects this coupling has on flight performance such as agility and efficiency will need to be studied.

C. Body plate

Another difficulty for properly modeling B2 has appeared because the simulated trajectories of the model often overshoot the actual experimental trajectories. Part of the reason is that the model contains three aerodynamic forces: one for the right wing, one for the left wing, and one for the tail. The problem is that the membrane is attached to the body at the centerline, so while the model accounts for the flapping wings, it ignores an element of the surface that remains flat against the body during the flight. When the robot is flying at a larger angle of attack, this surface would contribute more drag to the robot than is currently modeled. As a result, we add an additional aerodynamic surface at the body to compensate for this.

We define the area of this plate to be A_b , and the x -displacement between its center and the body CoM to be r_b . We roughly estimate this offset to be -3 cm. The value for A_b is difficult to determine with certainty as the wing surface is continuous, and it is a bit arbitrary as to how much area is included in the wings and in this plate. Therefore, A_b is a decision variable in the optimization in Section V.

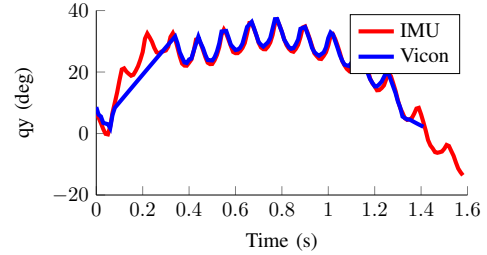


Fig. 3: Pitch angle recorded by Vicon system (blue) and IMU (red) for one flight experiment.

D. Tail mapping function

In spite of these additions, simulations of flights with a tail angle of $q_{DV} = 0^\circ$ drop significantly faster than the actual experimental data for this tail position even after optimizing all parameters with data using the methods in Section V. Other positive angles $q_{DV} \geq 0^\circ$ likewise underestimate the altitude gain. We have noticed a high sensitivity for angles near this 0° mark, and this occurs partly because the membrane on the tail is stretched the tightest at the neutral position. As a result, there is a sharp jump between positive and negative angles at this transition point. We model this by creating the mapping of the tail angle

$$\bar{q}_{DV} = \begin{cases} q_{DV} + c_{DV}^+, & q_{DV} \geq 0 \\ q_{DV}, & q_{DV} < 0 \end{cases} \quad (6)$$

where c_{DV}^+ is a positive constant. The new term \bar{q}_{DV} is used in place of q_{DV} to compute the aerodynamic forces on the tail, but q_{DV} is still used in the dynamics. The value of c_{DV}^+ will be determined in the optimization in Section V. We note that though there is a discontinuity here, the model can be used for control and planning in the future by fitting a function to smooth the transition.

E. Strip theory

One final change to the model is that we now use strip theory for computing the aerodynamic forces. We have selected to use 3 strips to approximate the rectangular wing shape by running simulation from 1 to 100 strips and finding 3 as the threshold for the error between k strips and 100 to be less than 1% for force in both the x and z directions over a wingbeat.

IV. DATA COLLECTION EXPERIMENTS

We incorporate the use of data to improve the prediction capabilities of the analytical model we have presented in the previous section. We have performed two types of experimental means of gathering data of B2. First, we have collected force sensor data using a load cell over a range of flapping frequencies. Second, we have collected position and orientation data of B2 launched into free flight for a series of different tail configurations.

B2's electronics consist of a set of actuators, sensors, and a processor. A brushless DC (BLDC) motor powers the robot to flap by driving a crank shaft that moves the wings

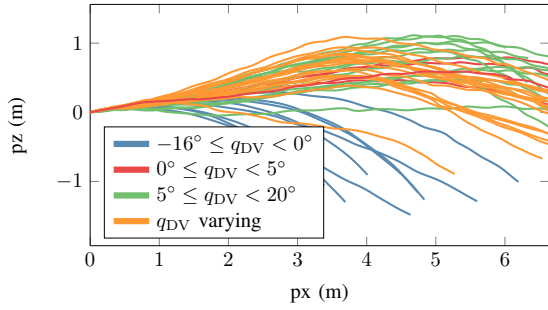


Fig. 4: All flight experiments plotted onto one figure with p_z vs. p_x displayed. This represents a variety of different tail configurations making the robot dive for some flights and gain altitude for others.

upward and downward. The two hindlimbs are independently controlled by two servo motors. B2’s wings and hindlimbs are constructed from hollow carbon fiber rods and nylon 3D-printed parts, and a thin silicone membrane is stretched across the entire structure. B2 has an on-board IMU (VectorNav VN-100) that estimates roll, pitch, and yaw attitude and relays them to the on-board computer (STM32f429II, 180 MHz 32-bit Arm CPU). A 2S LiPo battery powers B2’s electronics and actuators. We use this configuration for the following experiments.

A. Load cell data

Force data was collected using the same methods as [12]. We powered B2 with a voltage of 8.4 V and secured it to an analog six-axis JR3 force-torque sensor (model #30E12A4). We used a dSPACE CLP1104 I/O box and DS1104 R&D Controller Board to measure and record the load cell signals at a sampling rate of 1000 Hz. B2 was driven to flap at a range of different frequencies from 2.5 Hz to 7.8 Hz. The wind speed for all trials is zero. The mean nominal force readings (no flapping) were subtracted from each trial to eliminate any sensor biases and the force of gravity. Finally, we low-pass filtered the data with a 6th-order Butterworth filter with 50 Hz as the cutoff frequency. We compute the average net force $\bar{\mathbf{F}}_{x,\text{exp}}^i$ in the x -direction (direction of thrust) for each trial i along with its flapping frequency ω_{FL}^i and group them as $\{\omega_{\text{FL}}^i, \bar{\mathbf{F}}_{x,\text{exp}}^i, i = 1, \dots, N_l\}$ for a total of $N_l = 9$ tests.

B. Free-flight data

We recorded 43 free-flight experiments in the Intelligent Robotics Laboratory (IRL) flight arena at the University of Illinois at Urbana-Champaign (UIUC). This facility is equipped with eight Vicon T40 motion capture cameras. We attached four reflective markers to the top of the robot as shown in Figure 1. We used the Vicon Tracker 3.4 software to compute the position and orientation of the rigid body formed by these markers. Using the Vicon DataStream SDK, we recorded this data at a rate of 100 Hz. The IMU on B2 concurrently estimated orientation, and the on-board computer logged this to a microSD card. This is a high-performance IMU with a

propriety algorithm for estimating orientation, and thus its estimation is comparable to the Vicon system as seen in Figure 3. No post-processing was necessary for the IMU data.

While past work [12] has used IMU only for recording data, we deemed it necessary to collect Vicon data as well to improve the accuracy of the z position data and to additionally record x position. The Vicon position data are quite accurate, but Vicon Tracker software occasionally miscalculates the orientation of the robot. This behavior has been reported in previous work for Vicon measurements of a bat robot [11]. The oscillation of the robot and the occlusion of the markers from the wings are likely causes to this problem because this issue was not present when the wings are fixed and not flapping. We process the data to ignore miscalculations. Ignoring jumps resulting from angle wrapping, data points are discarded if any of the Euler angles jump beyond 90° from the mean angle for that given flight. Any gaps left in the data are fixed with linear interpolation. We estimate the velocity and acceleration using forward finite differencing. The data collected has a 3D representation, with roll, pitch, and yaw Euler angles and x , y , and z positions. The model in this paper is 2D and all flights in this study are straight with minimal roll, yaw, and y movement, so we consider only the pitch angle q_y , as well as translational coordinates p_x and p_z (after aligning the x axis with the direction of flight).

Figure 3 shows the correspondence between the Euler angles measured from IMU and Vicon. This figure demonstrates two things. First, the IMU produces very accurate measurements with little to no drift. Second, the Vicon measurements have jumps and discrepancies in the orientation of the robot even after processing. Thus, we use the orientation data of the IMU and the position data of the Vicon system. We synchronize the time between the two data sets using the cross correlation of each pitch angle q_y to compute the number of time samples the data needs to be shifted by. Lastly, the robot does not have a sensor for measuring the flapping angle q_{FL} , so we estimate this angle using the measured pitch q_y . The flapping angle is the same frequency as pitch oscillation, and in fact they both are in phase with each other. The pronation angle q_{PS} is passive in B2, but in our model it tracks a reference 90° out of phase with q_{FL} , and thus we estimate this from q_{FL} .

B2 was launched for each flight test using a custom built launcher that accelerated the robot to a speed of roughly 8 m/s. We set the throttle of the BLDC motor by hand to drive flapping at roughly 8.5 Hz immediately before launch. The computer began logging IMU data upon sensing a large x -acceleration value. The launcher was considerably more reliable than hand launch for minimizing variation in initial conditions. Specifically, minimizing the roll rate and keeping the initial roll position at 0° is critical because we are using a 2D model. The flight paths of all the tests recorded are shown in Figure 4.

The data set consists of 43 flights with 12 different tail configurations of q_{DV} between -16° and 20° . Each flight has a duration of roughly 1 s. The flapping frequencies for all tests are between 8.1 Hz and 9.3 Hz, and the initial launch

velocities are between 7.7 m/s and 8.7 m/s. The first series of tests were fixed to constant $q_{DV} < 0$ tail positions and the second series were fixed to constant $q_{DV} \geq 0$ positions. The third series started at a constant $q_{DV} = 3^\circ$, then the tail was moved down to a negative angle, and then shortly after up to a positive angle. We group the variables \mathbf{q} and derivatives $\dot{\mathbf{q}}$ of each recorded trajectory i with N_i time samples into the time series of state vectors $\{\mathbf{x}_{\text{exp}}^i(t_k), k = 0, \dots, N_i - 1, i = 1, \dots, N_f\}$ for all N_f flights.

V. PARAMETER ESTIMATION

The analytical model of B2 is based on many approximations, and as a result there are unmodeled dynamics and parameters that need to be adjusted. In this section, we outline the use of data recorded of B2 to optimize these model parameters. We utilize a novel methodology for using load cell data and free-flight data to adjust the model parameters. Initially, we use load cell data to tune the amplitude of the pronation angle sinusoidal reference trajectory in order to match the thrust produced in simulation to that recorded on the load cell. The other model parameters are then tuned through optimization by matching simulated flights with actual flight experiments.

A. Optimization with load cell data

Hoff et al. [12] used load cell data of a bat robot to estimate a_{PS} , the amplitude of the pronation angle reference trajectory q_{PS}^r . We likewise use these methods for this purpose. We improve on this method by setting up an optimization routine for this estimation. The cost function is

$$\mathcal{J}(a_{\text{PS}}) = \sum_{i=1}^{N_f} \frac{1}{|\bar{\mathbf{F}}_{x,\text{exp}}^i|} (\bar{\mathbf{F}}_{x,\text{sim}}^i - \bar{\mathbf{F}}_{x,\text{exp}}^i)^2 \quad (7)$$

to minimize the error between the average of the simulated model and the average of the load cell data of the force in the x -direction. This formulation matches the average thrust of the model to the average thrust of the experiments. The term $\bar{\mathbf{F}}_{x,\text{exp}}^i$ is the mean of the recorded load cell force in the x -direction, and $\bar{\mathbf{F}}_{x,\text{sim}}^i$ is the mean of the simulated force in the x -direction. $\bar{\mathbf{F}}_{x,\text{sim}}^i$ is a function of the decision variable a_{PS} . The optimization is subject to the bounds $0^\circ \leq a_{\text{PS}} \leq 45^\circ$. We divide by $|\bar{\mathbf{F}}_{x,\text{exp}}^i|$ to normalize the weight at each frequency.

We provide the initial guess of $a_{\text{PS}} = 11.5^\circ$ by recording B2 flapping at full speed and measuring the pronation angle of the video frame when $q_{\text{FL}} = 0^\circ$, which is the maximum amplitude for q_{PS} because of the 90° phase offset. This is the initial guess for the optimizer.

B. Optimization with flight experiments

Past works state the limitations of using only load cell data for flapping systems due to the system being constrained rigidly to a platform and not allowed to oscillate naturally [23]. While the optimization with load cell data has improved thrust prediction of B2 by estimating a_{PS} , many of the other parameters must also be tuned in order to have accurate free-flight prediction. The current values for many of the

parameters of the first-principles model are estimates based on physical measurements of B2. There are unmodeled dynamics in the system, and we can account for some of these effects by optimizing these model parameters. We proceed to use the model from [12] as an initial guess for estimating new parameters using data from free-flight experiments.

We select the decision variables to be w_c (wing chord length), t_s (tail span), A_b (area of body plate), C_{L1} , C_{D1} , C_{D0} (aerodynamic coefficients), c_{DV}^+ (tail function mapping parameter), and a_{coup} (factor for coupling q_{PS} and q_{DV}). The wing chord and tail span are both estimates because the wing and tail are part of a contiguous surface, and it is a bit arbitrary as to where they should be separated. Consequently they are included as decision variables in the optimization. We group all of these parameters into the vector \mathcal{P} .

The model of B2 is to be used for trajectory optimization and control, so we desire it to have excellent long-term prediction capabilities with minimal long-term simulation errors. Therefore we consider a multistep prediction formulation. Given the same initial condition and actuator commands as each experimental flight result, ideally our simulated model will closely match those states trajectories. Therefore, we set up our objective function as

$$\mathcal{J}(\mathcal{P}) = \sum_{i=1}^{N_f} \sum_{k=0}^{N-1} \|\bar{\mathbf{x}}_{\text{sim}}^i(t_k) - \bar{\mathbf{x}}_{\text{exp}}^i(t_k)\|^2 \quad (8)$$

where $\bar{\mathbf{x}} = [q_y \ p_x \ p_z \ \dot{q}_y \ \dot{p}_x \ \dot{p}_z]^\top$. Here we are only considering the error between the body CoM orientation, position, and their velocities as a metric for evaluating tracking performance of our model because the actuated coordinates should be roughly equivalent.

We use forward Euler integration with a step size of $dt = 0.001$ s as

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \mathbf{f}(t_k)dt \quad (9)$$

to generate the simulated state trajectory $\{\mathbf{x}_{\text{sim}}^i(t_k), k = \{0, \dots, N-1\}\}$ for trial i . The initial state of the forward simulation is set to $\mathbf{x}_{\text{sim}}^i(t_0) = \mathbf{x}_{\text{exp}}^i(t_0)$, i.e. the initial state of the experimental flight result. The references for the actuated states q_{FL} , q_{PS} , q_{DV} , and their derivatives are likewise set to be those recorded in data. For the flights in which the tail actuator changed between multiple positions, we add a small delay of 0.1 s to q_{DV}^r when simulating the flights to account for the delay of the servo motors. We add the constraint on the aerodynamic coefficients $C_{D0} + C_{D1} \geq 0$ to prevent drag term from adding thrust instead of drag. If C_D is negative, drag would be adding thrust to the model.

We used the interior-point algorithm of MATLAB's *fmincon* to solve the proposed optimization problem. The original model parameters from [12] were used as the initial guess to the optimization.

VI. RESULTS

We evaluate the performance of long-term prediction by comparing the original model to the optimized model with the

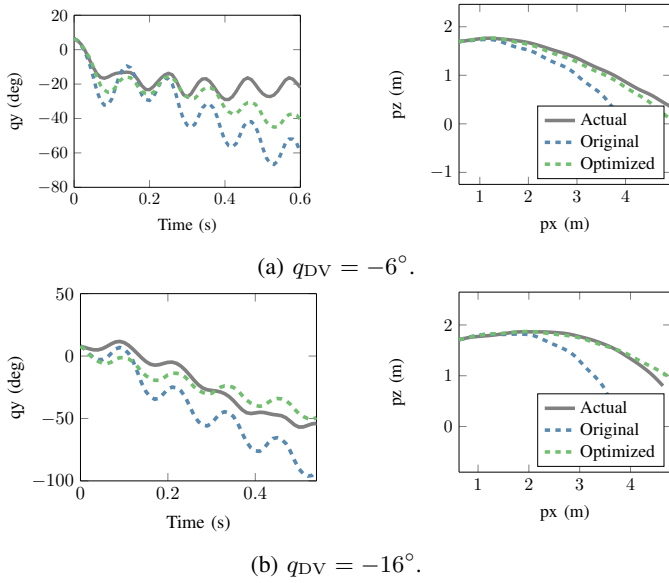


Fig. 5: Parameter estimation results showing pitch angle q_y and body CoM position p_z vs p_x for $q_{DV} < 0$ of flights from the test data set. Gray is actual flight data, dotted blue is the original model prediction of the model after forward simulation, and dotted green is the optimized prediction of the model.

Parameter	Initial	Optimized	Description
w_c	0.24	0.172	wing chord (m)
t_s	0.12	0.089	tail span (m)
C_{L1}	1.58	1.332	lift coefficient amplitude
C_{D1}	-1.55	-1.639	drag coefficient amplitude
C_{D0}	1.65	1.713	drag coefficient offset
A_b	0.00	0.012	area of body plate (m ²)
c_{DV}^+	0.00	0.088	q_{DV} offset parameter
a_{coup}	0.00	-0.592	q_{DV} - q_{PS} coupling offset

TABLE I: Comparison of the initial guesses (original model parameters) and final values of the decision variables of the parameter estimation.

added terms. We separate the free-flight data into a training data set and a testing data set. We select 13 training flight tests that cover the range of q_{DV} positions of the data set, and we run parameter estimation with these to train the model. The subsequent analyses and plots use only the remaining 30 flights of the test data set. We select a handful of different flight experiments to represent different scenarios of the tail position for $q_{DV} < 0$ (Figure 5), $q_{DV} \geq 0$ (Figure 6), and q_{DV} initially constant, pitching down, and then pitching up (Figure 7). Each of these figures shows the pitch angle q_y over the full trajectory and the body's flight path given by p_z vs. p_x . Figure 7 also shows the prediction of the velocities. The spikes in \dot{p}_x are a result of noise after finite differencing the position data. The plots for $q_{DV} < 0$ and large positive q_{DV} show how the original model overestimated the effects of the tail's influence on the dynamics and the new and optimized model's effectiveness in prediction. The improvement in prediction of

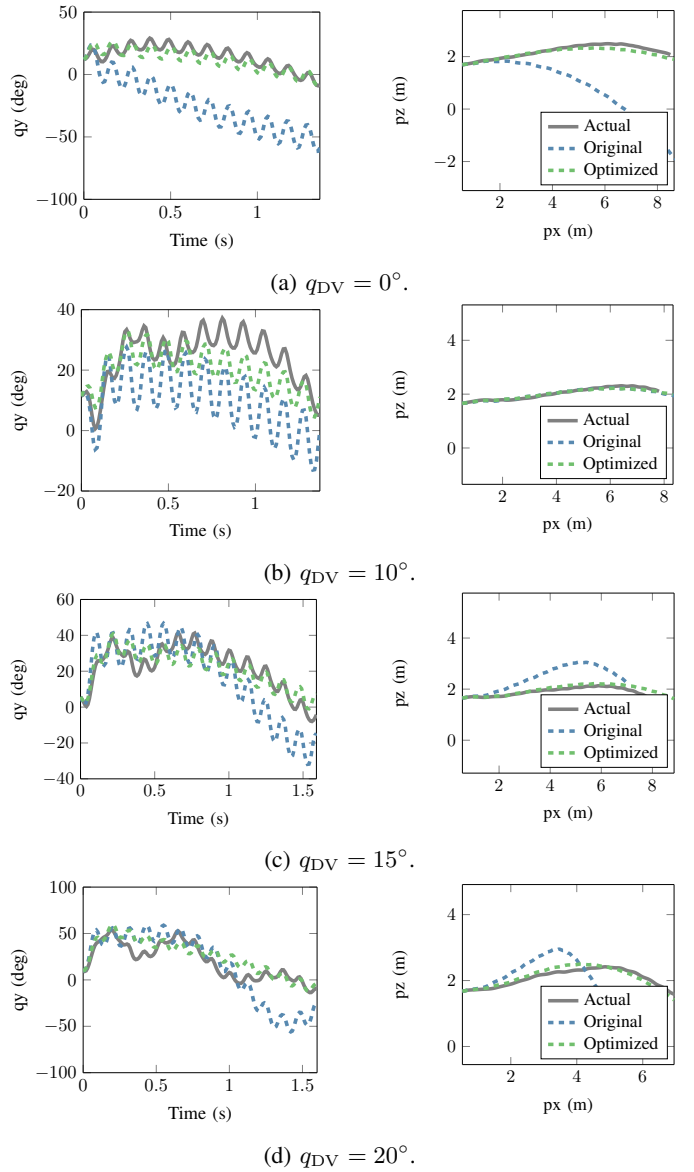


Fig. 6: Parameter estimation results showing pitch angle q_y and body CoM position p_z vs p_x for $q_{DV} \geq 0$ of flights from the test data set.

$q_{DV} = 0$ is partly due to the added bias term from Equation (6) whose parameter was optimized. The evolution of the pitch angle and altitude are remarkably accurate for most cases.

While the prediction has a high level of accuracy, flights with larger q_{DV} positions are more difficult to predict. The pitch angle estimation at the end of the trajectory in Figure 7 and those with large q_{DV} in Figure 6 deviate somewhat from the actual. This is likely due to the higher angle of attack of the system. However, altitude prediction remains accurate, the average pitch estimation is satisfactory for this application, and we still see a great improvement over the previous model.

The optimized values for the decision variables in the optimization are shown in Table I. The changes of the parameters are difficult to determine because they represent responses to

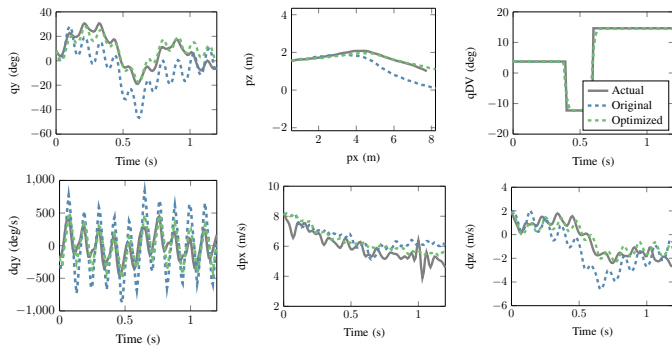


Fig. 7: Parameter estimation results showing states and velocities for a flight from test data with $q_{DV} = 3^\circ$, then $q_{DV} = -12.5^\circ$, and finally $q_{DV} = 15^\circ$.

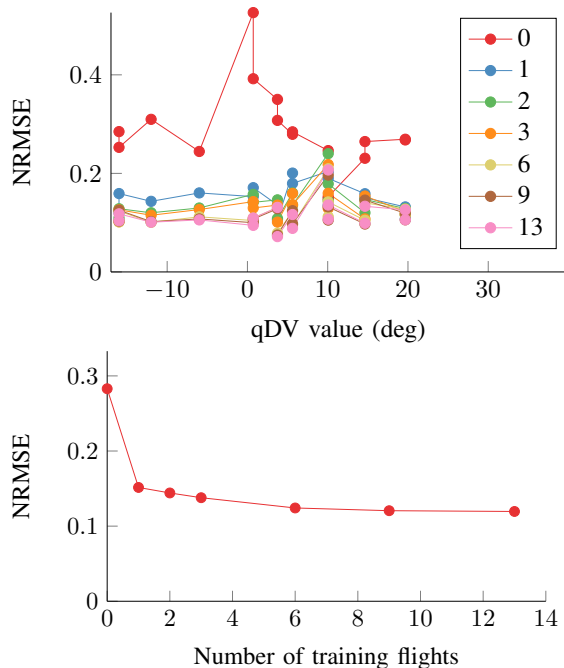


Fig. 8: NRMSE for predictions of flights in the test data set. The top plot shows the error for flights with fixed q_{DV} positions and the bottom is the mean of the NRMSE for predicted flights vs. the number of flights used for training the model. The legend specifies the varying number of training flights used in the optimization.

unmodeled dynamics of the system. However, a few seem to have physical explanations. The wing chord length and tail span have both decreased significantly, and this effectively decreases both wing and tail area. Area is directly proportional to the aerodynamic force on the surface, so consequently the force is likewise reduced. The improvement from reducing wing area is reflected in Figure 5, Figure 6, and Figure 7 because the pitch angle oscillation amplitude is reduced and now matches significantly better to the flight results. The reduced tail area is also apparent in the plots because the original model overestimated the effect of the tail for increasing and

decreasing pitch and altitude. The addition of the coupling term and the tail function both compensate for the rigidity assumptions, and it can be seen in Table I that they have nontrivial values.

Without the wing-tail coupling term in Equation (5), even after optimizing the parameters the model would overestimate the tail's influence for large positive q_{DV} and gain too much altitude. The effect of the coupling can be explained as follows. When the tail is tilted up to a large angle, the wings have a negatively biased tilt because of the wing-tail coupling. The tail produces a net torque on the system from the aerodynamic force on it and reorients the body to pitch up. If the wings had a neutral tilt, the angle of attack of the wings would be positive and would produce more lift and force the robot to a higher altitude. However, given the negative tilt, this reorientation gives the wings a neutral angle of attack, producing less lift in this case and less altitude gain. This is clear from the plots for $q_{DV} = 15^\circ$ and $q_{DV} = 20^\circ$ in Figure 6.

Additionally, it is important to consider the amount of data needed to train the model. We run parameter estimation with varying numbers of training flights, and we compute the normalized root-mean-square error (NRMSE) for each flight of the test data set for the unactuated states \bar{x} . Figure 8 compares the NRMSE against the number of flight experiments used for training. This plot demonstrates several things. First, there is a clear trend of improvement for every predicted flight test, and most significantly for trials at $q_{DV} = 0^\circ$. Second, even after training with just a single 1-second flight test, there are dramatic improvements in performance, demonstrating that the model structure requires very few data points to be trained with and that this method generalizes to flight conditions of q_{DV} not used in training. After training with 3 trials, the relative change in the decision variables becomes significantly smaller indicating that the threshold is around this number. However, this number is subject to change depending on the intended application.

VII. CONCLUSION

In this study, we have developed a data-driven methodology to create an accurate model of a bat robot by using the previous physics-based structure of an existing model [12], adding several additional terms to the structure, and optimizing the parameters using a series of load cell experiments and flight tests. Our long-term prediction results demonstrate the accuracy of the model. One limitation of this method is that new data may be required to retrain the model when modifications are made to the physical system. However, we have shown that only a few flight experiments are necessary to achieve excellent long-term prediction because of the setup of the model structure and the additional terms added, and training with even a single flight test significantly improves prediction.

ACKNOWLEDGMENTS

The flight experiments were performed in IRL at UIUC.

REFERENCES

- [1] S. Armanini, J. Caetano, G. De Croon, C. De Visser, and M. Mulder. Quasi-steady aerodynamic model of clap-and-fling flapping MAV and validation using free-flight data. *Bioinspiration & Biomimetics*, 11(4):046002, 2016.
- [2] S. F. Armanini, M. Karásek, and C. C. de Visser. Global LPV model identification of flapping-wing dynamics using flight data. In *AIAA Modeling and Simulation Technologies Conference*, page 2156, 2018.
- [3] J. V. Caetano, C. De Visser, G. De Croon, B. Remes, C. De Wagter, J. Verboom, and M. Mulder. Linear aerodynamic model identification of a flapping wing MAV based on flight test data. *International Journal of Micro Air Vehicles*, 5(4):273–286, 2013.
- [4] P. Chirarattananon and R. J. Wood. Identification of flight aerodynamics for flapping-wing microrobots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1389–1396. IEEE, 2013.
- [5] G. De Croon, K. De Clercq, R. Ruijsink, B. Remes, and C. De Wagter. Design, aerodynamics, and vision-based control of the DelFly. *International Journal of Micro Air Vehicles*, 1(2):71–97, 2009.
- [6] X. Deng, L. Schenato, W. C. Wu, and S. S. Sastry. Flapping flight for biomimetic robotic insects: Part I—system modeling. *IEEE Transactions on Robotics*, 22(4):776–788, 2006.
- [7] J. Grauer, E. Ulrich, J. E. Hubbard, D. Pines, and J. S. Humbert. Testing and system identification of an ornithopter in longitudinal flight. *Journal of Aircraft*, 48(2):660–667, 2011.
- [8] J. A. Grauer and J. E. Hubbard Jr. Multibody model of an ornithopter. *Journal of Guidance, Control, and Dynamics*, 32(5):1675–1679, 2009.
- [9] W. Hoberg and R. Tedrake. System identification of post stall aerodynamics for UAV perching. In *AIAA Infotech Aerospace Conference*, page 1930, 2009.
- [10] J. Hoff, A. Ramezani, S.-J. Chung, and S. Hutchinson. Synergistic design of a bio-inspired micro aerial vehicle with articulated wings. In *Robotics: Science and Systems (RSS)*, 2016.
- [11] J. Hoff, A. Ramezani, S.-J. Chung, and S. Hutchinson. Optimizing the structure and movement of a robotic bat with biological kinematic synergies. *The International Journal of Robotics Research*, 37(10):1233–1252, 2018.
- [12] J. Hoff, U. Syed, A. Ramezani, and S. Hutchinson. Trajectory planning for a bat-like flapping wing robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [13] M. Keennon, K. Klingebiel, and H. Won. Development of the nano hummingbird: A tailless flapping wing micro air vehicle. In *50th AIAA Aerospace Science Meeting*, page 588, 2012.
- [14] J. Lee, S. Ryu, T. Kim, W. Kim, and H. J. Kim. Learning-based path tracking control of a flapping-wing micro air vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7096–7102. IEEE, 2018.
- [15] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340(6132):603–607, 2013.
- [16] K. Peng, F. Lin, and B. M. Chen. Modeling and control analysis of a flapping-wing micro aerial vehicle. In *IEEE International Conference on Control and Automation (ICCA)*, pages 295–300. IEEE, 2017.
- [17] A. T. Pfeiffer, J.-S. Lee, J.-H. Han, and H. Baier. Ornithopter flight simulation based on flexible multi-body dynamics. *Journal of Bionic Engineering*, 7(1):102–111, 2010.
- [18] A. Ramezani, X. Shi, S.-J. Chung, and S. Hutchinson. Lagrangian modeling and flight control of articulated-winged bat robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2867–2874. IEEE, 2015.
- [19] A. Ramezani, X. Shi, S.-J. Chung, and S. Hutchinson. Bat Bot (B2), a biologically inspired flying machine. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3219–3226. IEEE, 2016.
- [20] A. Ramezani, X. Shi, S.-J. Chung, and S. Hutchinson. Nonlinear flight controller synthesis of a bat-inspired micro aerial vehicle. In *AIAA Guidance, Navigation, and Control Conference*, page 1376, 2016.
- [21] A. Ramezani, S.-J. Chung, and S. Hutchinson. A biomimetic robotic platform to study flight specializations of bats. *Science Robotics*, 2(3):eaal2505, 2017.
- [22] R. Ritz and R. D’Andrea. A global controller for flying wing tailsitter vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2731–2738. IEEE, 2017.
- [23] C. Rose and R. S. Fearing. Comparison of ornithopter wind tunnel force measurements with free flight. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1816–1821. IEEE, 2014.
- [24] C. J. Rose, P. Mahmoudieh, and R. S. Fearing. Modeling and control of an ornithopter for diving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 957–964, 2016.
- [25] Z. J. Wang. Dissecting insect flight. *Annual Review of Fluid Mechanics*, 37:183–210, 2005.
- [26] Z. J. Wang, J. M. Birch, and M. H. Dickinson. Unsteady forces and flows in low Reynolds number hovering flight: two-dimensional computations vs robotic wing experiments. *Journal of Experimental Biology*, 207(3):449–460, 2004.
- [27] J. Zhang, F. Fei, Z. Tu, and X. Deng. Design optimization and system integration of robotic hummingbird. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5422–5428. IEEE, 2017.