

# Trajectory planning for a bat-like flapping wing robot

Jonathan Hoff<sup>1</sup>, Usman Syed<sup>1</sup>, Alireza Ramezani<sup>2</sup>, and Seth Hutchinson<sup>1,3</sup>

**Abstract**—Planning flight trajectories is important for practical application of flying systems. This topic has been well studied for fixed and rotary winged aerial vehicles, but far fewer works have explored it for flapping systems. Bat Bot (B2) is a bio-inspired flying robot that mimics bat flight, and it possesses the ability to follow a designed trajectory with its on-board electronics and sensing. However, B2's periodic flapping and its complex aerodynamics present major challenges in modeling and planning feasible flight paths. In this paper, we present a generalized approach that uses a model with direct collocation methods to plan dynamically feasible flight maneuvers. The model is made to be both accurate through collection of load cell force data for parameter selection and computationally inexpensive such that it can be used efficiently in a nonlinear solver. We compute the trajectory of launching B2 to a desired altitude and a banked turn maneuver, and we validate our methods with experimental flight results of tracking the launch trajectory with a PD controller.

## I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have numerous applications in society from package delivery to quality inspection of construction sites. These tasks require safe and accurate trajectory planning in order to operate in settings shared by humans. While UAVs are effective, their rotors are dangerous to humans in shared environments. Biologically inspired fliers offer an excellent alternative because they are considered safe: their flapping wings are not dangerous to people. Attributes of biological fliers such as agility and efficiency have motivated research in flapping-wing systems [1], and numerous works have begun to develop bio-inspired flapping fliers [2]–[6]. Bats in particular possess these qualities, and recent works have developed and improved Bat Bot (B2) [6]–[8], a bio-inspired flapping wing robot that mimics bat flight. These advances in flapping flight have increased the interests in planning for these systems.

However, while there is a strong body of literature in planning flight for quadcopters and fixed-wing UAVs, there have been fewer works for flapping wing systems. Researchers have planned dive maneuvers in which the wings were held constant during the maneuver [9], [10]. Only a few works in flapping flight have generated feasible trajectories for the flapping system to follow [11], [12]. Additionally, B2 is a unique platform that is larger than most flapping-wing micro

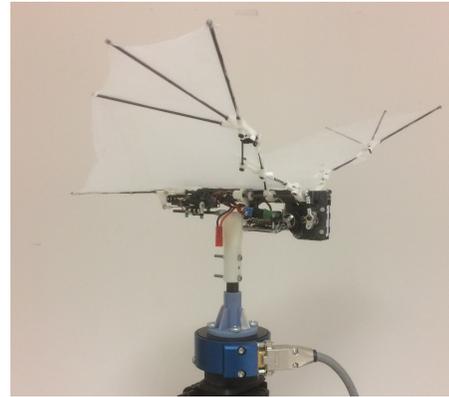


Fig. 1: B2 on load cell sensor experimental setup.

aerial vehicles (FWMAV) but smaller than most ornithopters with a flexible membrane and morphing wings. This type of system has not yet been considered for trajectory planning.

In this paper, we address the challenge of creating a strategy for planning autonomous flight for B2 that generalizes to multiple types of maneuvers. The contribution of this work is twofold. First, we have developed a hybrid first principles and data-driven model that captures the dynamics of B2 and is computationally tractable such that it can be used for trajectory planning. This approach utilizes load cell data to select model parameters to improve modeling accuracy. Second, we generate dynamically-feasible flight paths for the robot using this model with direct collocation methods. We have tested our approach by conducting closed-loop flight tests on our experimental platform B2 using its on-board computing and sensing for a launch maneuver. B2 tracks a generated trajectory of launch from rest to a desired altitude with a proportional-derivative (PD) controller. We have also extended the model to three dimensions and planned a banked turn in simulation. Our research helps fill the gap of trajectory planning in the literature as well as present a general methodology for modeling and planning with B2.

The contents of this paper are ordered as follows. Section II provides a thorough description of the dynamic model of B2. The proposed trajectory optimization routine and its simulation results are given in Section III. We present the experimental flight tests in Section IV. Comments and future work are discussed in Section V.

## II. MODELING

Modeling flapping flight is a very challenging problem due to the unsteady time-varying aerodynamic forces produced by the wings. B2 is an especially complex system

\*This research was supported by NSF Grant 1427111.

<sup>1</sup>Authors are with the Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (UIUC), Urbana, IL 61801, USA. jehoff2@illinois.edu

<sup>2</sup>Authors are with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA.

<sup>3</sup>Authors are with the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30308, USA.

because of the thin, flexible membrane that deforms over a wingbeat cycle. Given these complexities, we propose a hybrid first-principles and data-driven model for representing B2 that is both computationally tractable for a trajectory planning routine and accurate in predicting B2's behavior. This simplified longitudinal model displayed in Figure 2 is similar to [7], with the following changes: the massless wing assumption is relaxed, wing pronation is added to model passive wing twisting, and the aerodynamic force coefficients are modified. While most works in flapping flight assume massless wings, B2's wings have non-negligible mass, with each weighing approximately 7 g out of a total of approximately 100 g. We consider the multi-body system of four rigid links: three flat plates for the right and left wings and hindlimbs, and one link for the body. We supplement the model by estimating model parameters from load cell force data to improve model accuracy.

### A. Lagrangian modeling

The flapping angle between the body  $xy$  plane and each wing is denoted  $q_{FL}$ , and it is actuated by the torque  $u_{FL}$ . When the wings flap, they passively pronate, i.e. rotate about the spanwise axis, because they are attached to the shoulders at the front of the wing. This is responsible for the forward thrust generation of the robot. We model this with the angle  $q_{PS}$  that is actuated by the torque  $u_{PS}$  such that it maintains a periodic trajectory. The tail is approximated also as a flat plate that pitches up and down, and this pitching angle relative to the body is measured as  $q_{DV}$ . These joints are assumed to have no damping or stiffness. The wings each have mass  $m_w = 7$  g and the tail has mass  $m_t = 2$  g, and they are included in the Lagrangian formulation. The underactuated coordinates are the pitch  $q_y$  and the body center of mass (CoM) position  $(p_x, 0, p_z)$ . We combine these configuration variables and inputs as

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

We use the Euler-Lagrange convention to derive

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

the equations of motion. Vector  $\Gamma(\mathbf{q}, \dot{\mathbf{q}})$  is the aerodynamic forces mapped to the configuration space. It will be defined in the following section. Matrix  $B = [\mathbf{0}_{3 \times 3} \quad \mathbf{I}_{3 \times 3}]^\top$  maps the inputs to directly actuate these Degrees of Freedom (DoF).

### B. Aerodynamics

Figure 2 shows the aerodynamic forces acting on the right wing ( $\mathbf{F}_r$ ), left wing ( $\mathbf{F}_l$ ), and tail ( $\mathbf{F}_t$ ). These are written with respect to the inertial frame. When B2 flaps its wings, the wingtip travels much faster than points close to the shoulder joint. Therefore, we integrate the chordwise strips of the lift  $\|\mathbf{dF}_{L,l}\|$  and drag  $\|\mathbf{dF}_{D,l}\|$  aerodynamic forces

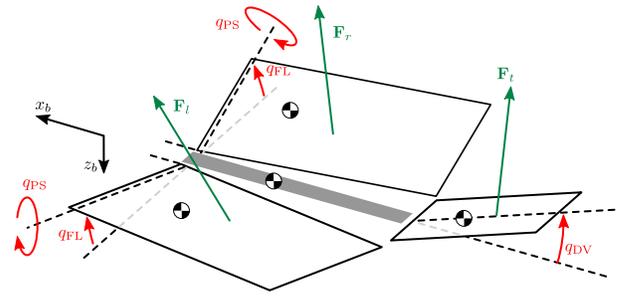


Fig. 2: Simplified 2D model of B2 dynamics and aerodynamics.

over the wingspan  $w_s = 0.22$  m of the left wing as

$$\begin{aligned} \|\mathbf{F}_{L,l}\| &= \int_0^{w_s} \|\mathbf{dF}_{L,l}\| = \frac{4}{3} \cdot \frac{1}{2} C_L(\alpha_l) \rho \|\mathbf{v}_{qc,l}\|^2 w_c w_s \\ \|\mathbf{F}_{D,l}\| &= \int_0^{w_s} \|\mathbf{dF}_{D,l}\| = \frac{4}{3} \cdot \frac{1}{2} C_D(\alpha_l) \rho \|\mathbf{v}_{qc,l}\|^2 w_c w_s \end{aligned} \quad (3)$$

where  $\rho = 1.1839$  kg/m<sup>3</sup> is the density of air,  $w_c = 0.24$  m is the wing chord length, and  $\mathbf{v}_{qc,l}$  is the velocity located at a quarter chord behind the leading edge and mid span of the left wing. The lift and drag coefficients  $C_L(\alpha_l)$  and  $C_D(\alpha_l)$  are functions of only the angle of attack  $\alpha_l$  of the left wing. The angle of attack is  $\alpha_l = \text{atan2}(v_{qc,l,z}, v_{qc,l,x})$ , where  $v_{qc,l,x}$  and  $v_{qc,l,z}$  are the  $x$  and  $z$  components of the velocity in the left wing frame. The  $\frac{4}{3}$  factor results from the integration and accounts for the higher velocity of the wing tip. This factor has been used for other models of bat flight with rectangular plates [13]. The forces on the right wing are calculated equivalently. The tail force  $\mathbf{F}_t$  is not integrated because there is no flapping motion. Researchers have relied on quasi-steady aerodynamic models for FWMAVs and ornithopters because of their simplicity and accuracy [2], [7], [14], [15]. These works follow the structure of Dickinson's quasi-steady model: the lift  $C_L$  and drag  $C_D$  coefficients are simple algebraic expressions of only angle of attack  $\alpha$ . We select this model for our lift and drag coefficients because of its simple representation and its effectiveness in previous works.

The aerodynamic forces are written with respect to the workspace. We use the principle of virtual work to transform these workspace forces into configuration space:

$$\Gamma(\mathbf{q}, \dot{\mathbf{q}}) = \frac{\partial \mathbf{p}_{qc,r}^\top}{\partial \mathbf{q}} \mathbf{F}_r + \frac{\partial \mathbf{p}_{qc,l}^\top}{\partial \mathbf{q}} \mathbf{F}_l + \frac{\partial \mathbf{p}_{qc,t}^\top}{\partial \mathbf{q}} \mathbf{F}_t. \quad (4)$$

From thin airfoil theory, we assume that the forces act at the points one quarter chord length behind the leading edge of the right wing  $\mathbf{p}_{qc,r}$ , left wing  $\mathbf{p}_{qc,l}$ , and tail  $\mathbf{p}_{qc,t}$ . Vector  $\Gamma(\mathbf{q}, \dot{\mathbf{q}})$  represents the generalized aerodynamic forces and torques on the configuration variables.

### C. Parameter tuning

The amplitude of the pronation angle is a challenging parameter to estimate because the pronating motion is a passive one in B2. It occurs because the forelimb is connected at

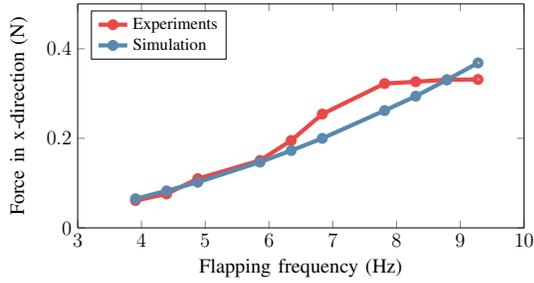


Fig. 3: Comparison of average net  $x$ -force (thrust generation) of physical experiments on a load cell (red) and those simulated with the proposed aerodynamic model (blue).

the front of the wing, and the wing is flexible. In order to estimate an appropriate amplitude for this angle, we recorded force data with an analog six-axis JR3 force-torque sensor (model #30E12A4). This sensor has a resolution of 0.005 N. Figure 1 shows B2 secured to the sensor. A fixed voltage of 8.4 V powered a speed controller and brushless DC (BLDC) motor driving the flapping motion. The load cell signals were recorded with a dSPACE CLP1104 I/O box and were saved onto a computer with a DS1104 R&D Controller Board. The sampling rate was 1000 Hz.

We recorded data at various flapping frequencies with no wind speed. The raw data consisted of the net force in the  $x$  direction (forward in the inertial frame). The recorded data were post-processed by subtracting the nominal force readings (no flapping) to remove the force of gravity and sensor biases, and they were filtered with a 6<sup>th</sup>-order Butterworth low-pass filter with a cutoff frequency of 50 Hz to remove high frequency noise.

The processed data were collected and displayed in Figure 3 to show the average net  $x$ -force recorded at different flapping frequencies. It can be observed that the pronating motion of the wings causes an average positive force in the  $x$ -direction, and it is responsible for the thrust that B2 produces. Furthermore, it is a function of the flapping frequency, as the thrust increases with higher frequencies. Using this data, we estimated the pronation angle amplitude such that our model very accurately approximates the thrust generation of B2, and it can be seen in Figure 3 that this is accurate for the full range of flapping frequencies.

### III. TRAJECTORY OPTIMIZATION

Trajectory optimization is a method that finds a state trajectory  $\mathbf{x}(t)$  and control input trajectory  $\mathbf{u}(t)$  that satisfy the system dynamics and minimize an objective function. This problem can be solved with either direct or indirect methods. Direct methods first discretize the states and inputs and solve the nonlinear programming problem, while indirect methods first attempt to satisfy the optimal control necessary conditions, and then discretize the solution. In this paper, we use direct collocation because the region of convergence often is smaller for indirect methods, indirect methods require explicitly deriving the necessary conditions which are

difficult for this problem [16], and direct collocation has had wide success in underactuated robots [17], [18].

Direct collocation discretizes the optimal control problem at  $N$  knot points to generate a finite dimensional nonlinear programming problem [19]. We select the Hermite-Simpson discretization scheme in which the state trajectories are represented as piecewise cubic Hermite splines and the input trajectories are piecewise linear splines. This discretization offers a higher level of accuracy than lower order methods such as trapezoidal for the same number of knot points, though the computation time is increased.

The system dynamics  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  are enforced at the midpoints between the knots, where  $\mathbf{x} = [\mathbf{q} \quad \dot{\mathbf{q}}]^T$ . The state at the midpoint between two knots is interpolated as

$$\bar{\mathbf{x}}_{k+\frac{1}{2}} = \frac{1}{2}(\mathbf{x}_k + \mathbf{x}_{k+1}) + \frac{h_k}{8}(\mathbf{f}_k - \mathbf{f}_{k+1}) \quad (5)$$

where  $\mathbf{x}_k = \mathbf{x}(t_k)$  and  $\mathbf{f}_k = \mathbf{f}(t_k, \mathbf{x}_k, \mathbf{u}_k)$ . The term  $h_k = (t_{k+1} - t_k)$  is the time difference between knot points  $k$  and  $k+1$ . The control input at this midpoint is linearly interpolated as  $\bar{\mathbf{u}}_{k+\frac{1}{2}} = (\mathbf{u}_{k+1} + \mathbf{u}_k)/2$ . We calculate the state derivative at the midpoint as  $\dot{\mathbf{f}}_{k+\frac{1}{2}} = \mathbf{f}(t_{k+\frac{1}{2}}, \bar{\mathbf{x}}_{k+\frac{1}{2}}, \bar{\mathbf{u}}_{k+\frac{1}{2}})$ . The state derivative at the midpoint is interpolated as

$$\dot{\mathbf{x}}_{k+\frac{1}{2}} = -\frac{3}{2h_k}(\mathbf{x}_k - \mathbf{x}_{k+1}) - \frac{1}{4}(\mathbf{f}_k + \mathbf{f}_{k+1}). \quad (6)$$

We compare the difference between these two computations by evaluating the defect vector

$$\zeta_k = \mathbf{f}_{k+\frac{1}{2}} - \dot{\mathbf{x}}_{k+\frac{1}{2}}, \quad (7)$$

i.e. the error between the actual dynamics and the polynomial approximation between the knot points. The equations  $\zeta_k = \mathbf{0}$  for all  $k \in \{0, 1, \dots, N-2\}$  form the set of equality constraints that enforce the polynomials to conform to the system dynamics. When this defect vector is close to zero, the cubic polynomials are accurately representing the system dynamics. The optimization is now a sparse finite-dimensional nonlinear programming problem that can be solved efficiently.

#### A. Partial feedback linearization

B2 flaps its wings continuously during a flight, and thus we can impose constraints on the dynamics such that  $q_{FL}$  and  $q_{PS}$  track periodic trajectories. We can use partial feedback linearization in order to enforce these conditions. We solve for  $\ddot{\mathbf{q}}$  in (2), and we separate the equations into the actuated ( $q_{FL}, q_{PS}, q_{DV}$ ) and unactuated ( $q_y, p_x, p_z$ ) coordinates as

$$\begin{aligned} \ddot{\mathbf{q}}_a &= \mathbf{f}_a(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_a(\mathbf{q})\mathbf{u} \\ \ddot{\mathbf{q}}_u &= \mathbf{f}_u(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_u(\mathbf{q})\mathbf{u}. \end{aligned} \quad (8)$$

Because this system is affine in control, we can simplify this expression by redefining the input terms  $u_{FL}$ ,  $u_{PS}$  and  $u_{DV}$ . The control action  $\mathbf{u} = \mathbf{g}_a^{-1}(\mathbf{q})(\boldsymbol{\nu} - \mathbf{f}_a(\mathbf{q}, \dot{\mathbf{q}}))$  simplifies the actuated dynamics to  $\ddot{\mathbf{q}}_a = \boldsymbol{\nu}$ . The new control term  $\boldsymbol{\nu} = [\nu_{FL} \quad \nu_{PS} \quad \nu_{DV}]^T$  allows direct shaping of the actuated coordinates. We force the flapping angle to follow a sinusoidal flapping trajectory  $q_{FL}^r(t) = a_{FL} \sin(\omega_{FL}t + b_{FL}) + c_{FL}$  to effect periodic flapping of the model. Similarly,

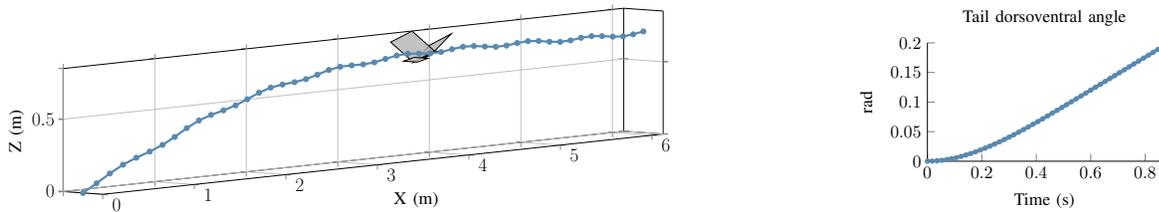


Fig. 4: Body CoM position  $(p_x, 0, p_z)$  and tail angle  $q_{DV}$  trajectories computed by optimization for a launch maneuver.

the pronation angle must follow the reference  $q_{PS}^r(t) = a_{PS} \sin(\omega_{FL}t + b_{PS}) + c_{PS}$  to mimic the passive pronations of each wingbeat. Note that the frequency  $\omega_{FL}$  for pronation is the same as that of flapping, as pronation is coupled to the flapping motion. Sinusoidal functions have been used in past work for the flapping and pronation angles [20]. We have run B2 simulations with the feedback linearization policy and a PD controller for  $q_{DV}$  and have shown stable flight.

An advantage of this approach is that the states  $q_{FL}$ ,  $q_{PS}$ ,  $\dot{q}_{FL}$ ,  $\dot{q}_{PS}$  and the corresponding inputs  $u_{FL}$  and  $u_{PS}$  can be removed from the decision variables from the optimization routine defined in the following section because we can assume perfect tracking of the states to their desired references. Therefore, we do not require enforcing the dynamic constraints for these variables, and thus the problem has fewer decision variables and constraints. These simplifications reduce the configuration variable vector to  $\mathbf{q} = [q_y \ p_x \ p_z \ q_{DV}]^T$  and the control to  $\boldsymbol{\nu} = \nu_{DV}$ .

### B. Launch trajectory to a specified altitude

One important maneuver to plan for B2 is launching from rest to reach a desired altitude. This entails selection of a feasible launch angle and the control input of the hindlimbs during this period. In formulating this as a trajectory optimization problem, we utilize the longitudinal model from Section II. We formulate the trajectory optimization problem such that B2 is launched from a given initial launch velocity  $v_0 = \sqrt{\dot{p}_{z_0}^2 + \dot{p}_{x_0}^2} = 9 \text{ m/s}$  at initial position  $(p_{x_0}, 0, p_{z_0}) = (0, 0, 0)$  to reach a desired final altitude  $p_{z_f}^d$ . The final vertical velocity  $\dot{p}_{z_f}$  should be close to 0 for stable flight, though not strictly 0 because of the periodic oscillation from flapping. Given these requirements, we write the constrained nonlinear programming problem as

$$\begin{aligned}
 & \underset{\mathbf{x}, \nu_{DV}, t_f}{\text{minimize}} && \mathcal{J}(\nu_{DV}) = \sum_{k=0}^{N-2} \frac{h_k}{2} (\nu_{DV}(t_{k+1})^2 + \nu_{DV}(t_k)^2) \\
 & \text{subject to} && f_1 : \underline{x}_i \leq x_i(t_k) \leq \bar{x}_i, \quad k = 0, \dots, N-1, \quad x_i \in \mathbf{x} \\
 & && f_2 : \underline{\nu}_{DV} \leq \nu_{DV}(t_k) \leq \bar{\nu}_{DV}, \quad k = 0, \dots, N-1 \\
 & && f_3 : 0 \leq t_f \leq t_{\max} \\
 & && f_4 : |x_i(t_f) - x_i^d(t_f)| \leq \epsilon_i, \quad x_i \in \mathbf{x} \\
 & && g_1 : \boldsymbol{\zeta}_k = \mathbf{f}_{k+\frac{1}{2}} - \dot{\mathbf{x}}_{k+\frac{1}{2}} = \mathbf{0}, \quad k = 0, \dots, N-2 \\
 & && g_2 : p_{x_0}, p_{z_0}, \dot{q}_{y_0}, q_{DV_0}, \dot{q}_{DV_0} = 0 \\
 & && \dot{p}_{x_0} = v_0 \cos q_{y_0}, \quad \dot{p}_{z_0} = v_0 \sin q_{y_0}.
 \end{aligned} \tag{9}$$

The cost function  $\mathcal{J}$  is the trapezoidal numerical integration of the control input  $\nu_{DV}$ . Hence, we are minimizing the

acceleration of  $q_{DV}$  to find the trajectory with the least control effort spent moving the hindlimbs. Consequently, it should ignore the fast dynamics induced by flapping and respond only to correct the slower average body dynamics.

The inequality constraints  $f_1$ ,  $f_2$ , and  $f_3$  are bounds on the configuration variables, their derivatives, the control input, and the time length of the trajectory. We select these based on actuator limits of the tail and so that the launch is forward and upwards (positive  $p_x$  and  $p_z$ ). Inequality constraints  $f_4$  enforce the desired final conditions while allowing slackness to the optimizer as specified by small constants  $\epsilon_i$  for  $x_i \in \mathbf{x}$ . The equality constraints of  $g_1$  are the collocation constraints to enforce the dynamics of the system from (7). The initial conditions of the flight are set by constraints  $g_2$ . The launching angle is constrained to be equal to the optimizer's choice of the initial pitch angle of B2. When launching B2, there should be no initial angle of attack of the body, i.e. the direction of launch should be aligned with the pitch orientation.

The proposed optimization problem was solved using MATLAB's constrained optimization algorithm *fmincon* with an interior-point algorithm. The initial guess was generated by selecting approximate initial and final states and linearly interpolating between them to form the state trajectories. The initial guess for the input is set to zero. We found that it wasn't necessary to simulate the dynamics to acquire the periodic behavior of the states in generating the initial guess because the optimization converges without issues using the linearly interpolated guess.

### C. Results

The simulation results of solving the trajectory optimization problem are shown in Figure 4 and Figure 6. As expected, the pitch angle is periodic because of the large aerodynamic and inertial forces generated from the flapping of B2's wings. This is less pronounced in the  $z$  position, but the small oscillations are still present. The hindlimb angle begins at 0 and is gradually tilted up over the course of the trajectory. This final nonzero tail angle is necessary for B2 to reach its final pitch angle and maintain straight flight. The gradual movement is a result of the objective function penalizing large control efforts over the trajectory. The importance of these results is found primarily in the optimizer's selection of the initial launch angle of  $q_{y_0} = 20^\circ$  and the control inputs shaping of the tail  $q_{DV}$  trajectory. A large launch angle will cause B2 to reach a larger altitude

but fail to maintain this, and too small of an angle will result in a slow climb to the desired altitude.

#### D. Banked turn simulation

Banked turning is a crucial maneuver in aerial vehicles because they require this to avoid obstacles and reach destinations. Its role is also significant in biological bat flight [21], and consequently we seek to perform banked turns with B2. We can extend the longitudinal model of B2 to 3D by adding the underactuated DoFs for the roll angle  $q_x$ , the yaw angle  $q_z$ , and the  $y$  position  $p_y$ . We also add the actuated DoFs of the right and left folding positions  $q_{FO,r}$  and  $q_{FO,l}$  and the tail rotation  $q_{RO}$ . The folding-unfolding DoFs  $q_{FO,r}$  and  $q_{FO,l}$  are the lengths of each wing. As the right wing is retracted, the length  $q_{FO,r}$  is reduced, decreasing the effective wing area. The tail rotation angle represents the asymmetric movement of the hindlimbs. One leg moving up and the other moving down can be modeled as a rotation of the tail plate about its chordwise direction. These two DoFs approximate the movement of both legs.

These additional DoFs combined with the longer required flight time of banked turning greatly increases computation time. We can improve the speed of the optimization by providing analytical constraint Jacobian information to the optimizer. The aerodynamic model is an analytical expression, so finding the Jacobian of this symbolically is possible. However, computing this for the full dynamics is difficult because it requires taking the Jacobian of the system dynamics in the form of  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ . Currently,  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  is computed numerically because it requires the inversion of the inertial matrix  $D(\mathbf{q})$ . Symbolic inversion of this  $12 \times 12$  matrix becomes very large, so we draw upon other works in trajectory optimization of quadrupeds in which the defect variables, i.e. the configuration variable accelerations  $\ddot{\mathbf{q}}$ , are added as decision variables [18]. The dynamics can then be enforced implicitly as in (2), avoiding the matrix inversion. We utilize chain rule to compute the Jacobians, specifically for the aerodynamic forces. Additionally, we use IPOPT [22] and supply the sparsity structure of the constraint Jacobian.

Given these speed improvements, we can plan a banked turn maneuver in which B2 must change its yaw by  $90^\circ$  by banking at a nonzero roll angle. As a beginning step, we will solve the feasibility problem by setting the cost  $\mathcal{J} = 0$ . We select the initial roll angle to be  $q_{x_0} = 20^\circ$  in order to initiate the banked turn. We restrict the final yaw position to be  $q_{z_0} = 90^\circ$  so that B2 completes a turn. We use 141 collocation points in the optimization. The optimized trajectory is plotted in Figure 5. The optimizer selects the initial launch angle of  $5^\circ$  and a time duration of 2 s such that B2 banks to the right and completes a full  $90^\circ$  turn.

## IV. EXPERIMENTS

We performed a series of flight tests using only B2's on-board sensors to demonstrate tracking of the optimized launch trajectory (Figure 6). B2 was equipped with a brushless DC (BLDC) motor that drives flapping and two servo motors that independently articulate the angles of the

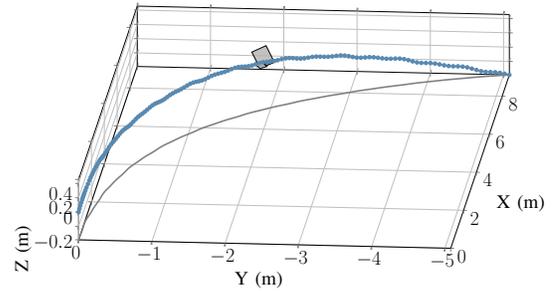


Fig. 5: Body CoM position  $(p_x, p_y, p_z)$  trajectory (blue) computed by the optimization for a banked turn maneuver. Its projection onto the  $xy$  plane (gray) is also shown.

two hindlimbs. A thin silicon membrane was secured to the wings, body, and hindlimbs. B2's on-board computer (STM32f429II, 180 MHz 32-bit Arm CPU) received sensor data from a VN-100 (VectorNav) inertial measurement unit (IMU) and computed control commands to the servo motors actuating the hindlimbs in order to follow the optimized trajectory. The attached video demonstrates a flight result of B2 tracking the optimized trajectory.

The VN-100 is a 10-axis IMU (3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, barometer). This sensor reports roll, pitch, and yaw by fusing the accelerometer, gyroscope, and magnetometer data with an Extended Kalman Filter. The pitch angle can be used directly as an estimate for  $q_y$ . We can estimate altitude by using atmospheric pressure measurements from the barometer with  $h_b = 44330 \left(1 - (p/p_0)^{0.1903}\right)$ , where  $p_0 = 1013.25$  mbar is the pressure at sea level and  $p$  is the current pressure reading from the sensor. We run this a sampling rate of 100 Hz and use a moving average filter to smooth the altitude estimate. These state estimates from the sensor were provided to the controller to compute errors between the actual and desired trajectories. We programmed a PD controller with inputs of pitch  $q_y$  and altitude measurements  $p_z$ . The optimized reference trajectories were programmed in the computer such that B2 can track these trajectories.

We performed the experimental flight tests in the Intelligent Robotics Laboratory (IRL) flight arena at the University of Illinois at Champaign-Urbana. B2 was launched by hand at approximately 9 m/s at the initial pitch angle selected by the optimizer. Throttle was set to maximum to produce a flapping frequency of roughly 10 Hz. At the end of the test, the throttle was shut off to prevent damage to the robot. Upon launch, the controller and time-parameterized trajectories are triggered by sensing the spike in  $x$ -acceleration via the IMU.

Four flight tests were recorded, and their respective pitch and altitude measurements and clips of a flight video are presented in Figure 6. B2 is able to track both the pitch angle and the desired altitude using measurements from its on-board sensors and controlling the hindlimb servo motors. These results demonstrate the effectiveness of the simplified model at predicting B2's complex flight behavior. It should be noted that there are small errors in the initial condition as

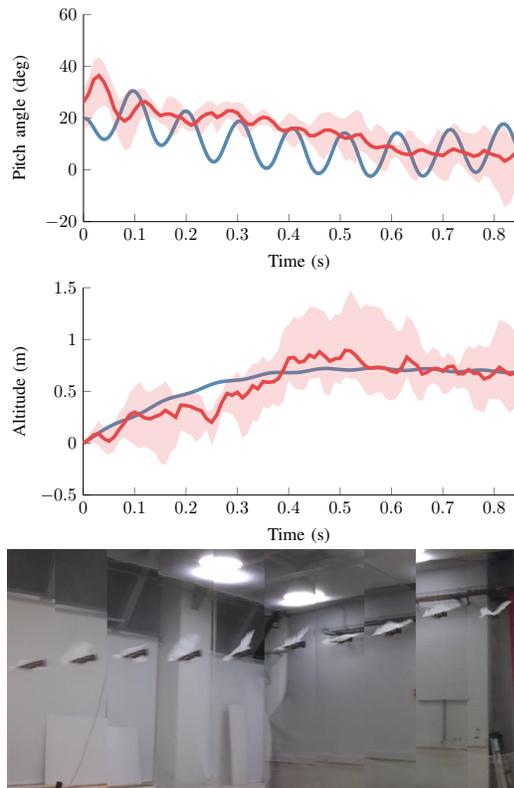


Fig. 6: Pitch angle and altitude tracking of four recorded closed-loop flight tests. The blue line is the optimized trajectory from simulation, the red shading shows the minimum and maximum values of the experimental flight tests, and the red line is the mean of the tests. The frames from the attached flight video are shown.

seen in the initial pitch angle from Figure 6 because B2 was launched by hand. Additionally, the average position of  $q_{FL}$  has a slightly positive bias ( $c_{FL} > 0$ ) because of the effects of air pushing upwards on the wings to generate lift. These explain some of the variance of the altitude measurements, most notably at 0.5 s. There is no initial condition error for altitude because the altitude is zeroed by its initial reading. In future testing, we will develop a launching mechanism for tighter control of the initial launch conditions.

## V. CONCLUSION

In this paper, we have formulated a methodology for planning flight maneuvers of the bio-inspired robotic bat B2. We created a simplified model that is both tractable for computation in nonlinear optimization and accurate from use of experimental load cell data. We then used this model to solve the trajectory optimization problem with direct collocation for a launch maneuver and a banked turn maneuver. The experimental flight tests validate the proposed framework. In future work, we can supplement load cell data with free flight experiments and create a more rigorous parameter estimation routine. We can also experiment more with the objective function and answer some interesting questions

relating to minimum turn curvature, the effects of initial launch conditions, and the role of wing folding.

## REFERENCES

- [1] D. J. Pines and F. Bohorquez, "Challenges facing future micro-air-vehicle development," *Journal of Aircraft*, vol. 43, no. 2, pp. 290–305, 2006.
- [2] 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*, vol. 22, no. 4, pp. 776–788, 2006.
- [3] 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*, vol. 1, no. 2, pp. 71–97, 2009.
- [4] 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*, 2012, p. 588.
- [5] K. Y. Ma, P. Chirattananon, S. B. Fuller, and R. J. Wood, "Controlled flight of a biologically inspired, insect-scale robot," *Science*, vol. 340, no. 6132, pp. 603–607, 2013.
- [6] A. Ramezani, S.-J. Chung, and S. Hutchinson, "A biomimetic robotic platform to study flight specializations of bats," *Science Robotics*, vol. 2, no. 3, p. eaal2505, 2017.
- [7] 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)*. IEEE, 2016, pp. 3219–3226.
- [8] 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*, vol. 37, no. 10, pp. 1233–1252, 2018.
- [9] 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)*, 2016, pp. 957–964.
- [10] L. J. Roberts, H. A. Bruck, and S. Gupta, "Modeling of dive maneuvers for executing autonomous dives with a flapping wing air vehicle," *Journal of Mechanisms and Robotics*, vol. 9, no. 6, p. 061010, 2017.
- [11] J. M. Dietl and E. Garcia, "Ornithopter optimal trajectory control," *Aerospace Science and Technology*, vol. 26, no. 1, pp. 192–199, 2013.
- [12] P. Chirattananon, K. Y. Ma, and R. J. Wood, "Perching with a robotic insect using adaptive tracking control and iterative learning control," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1185–1206, 2016.
- [13] A. J. Bergou, S. M. Swartz, H. Vejdani, D. K. Riskin, L. Reimnitz, G. Taubin, and K. S. Breuer, "Falling with style: bats perform complex aerial rotations by adjusting wing inertia," *PLoS Biology*, vol. 13, no. 11, p. e1002297, 2015.
- [14] J. M. Dietl and E. Garcia, "Stability in ornithopter longitudinal flight dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 1157–1163, 2008.
- [15] T. S. Clawson, S. B. Fuller, R. J. Wood, and S. Ferrari, "A blade element approach to modeling aerodynamic flight of an insect-scale robot," in *American Control Conference (ACC)*. IEEE, 2017, pp. 2843–2849.
- [16] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. Siam, 2010, vol. 19.
- [17] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [18] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1447–1454.
- [19] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.
- [20] J. D. DeLaurier, "An aerodynamic model for flapping-wing flight," *The Aeronautical Journal*, vol. 97, no. 964, pp. 125–130, 1993.
- [21] J. Iriarte-Díaz and S. M. Swartz, "Kinematics of slow turn maneuvering in the fruit bat *Cynopterus brachyotis*," *Journal of Experimental Biology*, vol. 211, no. 21, pp. 3478–3489, 2008.
- [22] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.