

Banking Turn of High-DOF Dynamic Morphing Wing Flight by Shifting Structure Response Using Optimization

Bibek Gupta¹, Yogi Shah¹, Taoran Liu¹, Eric Sihite² and Alireza Ramezani^{1*}

Abstract—The 3D flight control of a flapping wing robot is a very challenging problem. The robot stabilizes and controls its pose through the aerodynamic forces acting on the wing membrane which has complex dynamics and it is difficult to develop a control method to interact with such a complex system. Bats, in particular, are capable of performing highly agile aerial maneuvers such as tight banking and bounding flight solely using their highly flexible wings. In this work, we develop a control method for a bio-inspired bat robot, the Aerobat, using small low-powered actuators to manipulate the flapping gait and the resulting aerodynamic forces. We implemented a controller based on collocation approach to track a desired roll and perform a banking maneuver to be used in a trajectory tracking controller. This controller is implemented in a simulation to show its performance and feasibility.

I. INTRODUCTION

Bats possess highly dynamic morphing wings which are known to be extremely high-dimensional with many active and passive modes [1]. They can take advantage of their flexible wings to dynamically morph the shape of their wings to perform highly acrobatic maneuvers such as tight banking and turning maneuvers [2], [3]. Copying bat dynamic morphing wing can bring fresh perspectives to micro aerial vehicle (MAV) design but is also extremely challenging [4]–[6].

The Northeastern University's *Aerobat* platform, shown in Fig. 1, is a bio-inspired bat robot that is developed to test dynamic morphing wing flight in realistic flight scenarios comparable to bat flights [7]–[9]. This robot features a dynamically morphing wing where the wing can fold and expand during upstroke and downstroke, respectively, which improves flight efficiency. The robot armwing is composed of the linkages we called the “computational structure” which drives the robot's flapping gait. Some of the linkages in the computational structure are more sensitive where a small change in linkage length can greatly change the resulting flapping gait. In the overarching attempts to achieve high-dimensional actuation in flapping-wing MAVs, we utilized several low-powered actuators to manipulate the flapping gait of the robot, which in turn manipulates the generated aerodynamic forces acting on the wing [7], [10]. Using this method, we aim to implement an active closed-loop controller for flight trajectory tracking. The flight dynamic

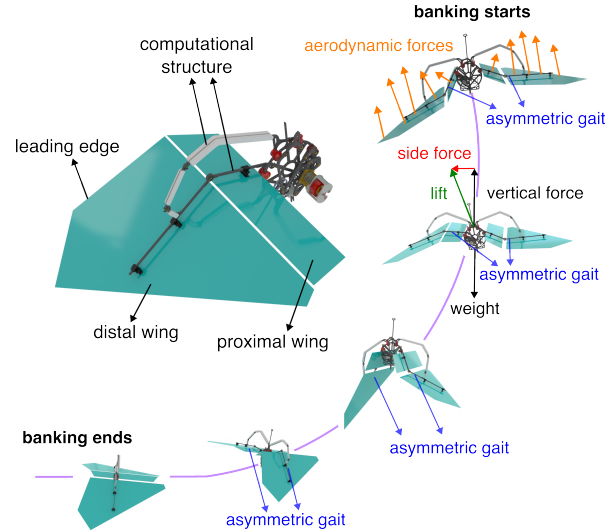


Fig. 1. Cartoon idealization of Aerobat performing sharp banking turns.

model of Aerobat [11] and new contributions in this work are explained in more detail in Section II.

The framework we propose in this work has allowed the fast activation and regulation of many actuated degrees of freedom (DOF). We limit the scope of this work to simulation results to show that this framework is feasible to be used in a closed-loop trajectory-tracking controller. The implementation and testing of this gait regulation will be done in future work. This paper is outlined as follows: we begin with the background, motivations, and new contributions from our Aerobat platform and this work. Then, we followed with actuation methods and 3D trajectory tracking controller design based on collocation approach, their implementation in simulations, results discussion, and finalized with the concluding remarks.

II. FLIGHT DYNAMICS MODEL AND MAIN CONTRIBUTIONS

A. Flight Dynamics Modelling

The flight dynamics of the Aerobat depicted in Fig. 1 can be elucidated through a cascade nonlinear system [11] of the following form:

$$\begin{aligned} \Sigma_{Full} : \begin{cases} \dot{x} = f(x) + g_1(x)u + g_2(x)y_2 \\ y_1 = h_1(x) \end{cases} \\ \Sigma_{Aero} : \begin{cases} \dot{\xi} = A_{\xi}(t)\xi + B_{\xi}(t)y_1 \\ y_2 = C_{\xi}(t)\xi + D_{\xi}(t)y_1 \end{cases} \end{aligned} \quad (1)$$

¹Authors are with the Silicon Synapse Labs, Department of Electrical and Computer Engineering, Northeastern University, Boston, USA. Emails: gupta.bi, a.ramezani@northeastern.edu

²Author is with the Department of Aerospace Engineering, California Institute of Technology, Pasadena, USA. Email: esihite@caltech.edu

*Corresponding author.

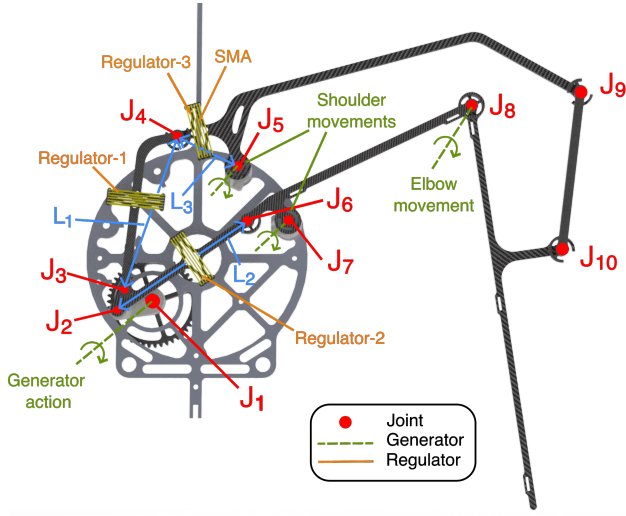


Fig. 2. Shows the computational structure used in Aerobot and the proposed low-power actuators' locations within the structure (regulator). Also shows the relevant joints (shoulder and elbow) that generate the resulting flapping gait (generator).

Here, t , x , and ξ represent time, the state vector, and hidden aerodynamic variables, respectively. In Eq. 1, the state vector x encompasses the position and velocity of both active q_a and passive q_p coordinates. The nonlinear terms denoted by $f(x)$ and $g_1(x)$ are derived from Lagrange equations and encompass inertial, Coriolis, and gravity effects [11]. The state-dependent matrices $g_1(x)$ and $g_2(x)$ map the joint actions $u(x)$ and external force $y_2(x)$ to the state velocity vector $\dot{x}$.

The output y_2 represents the aerodynamic force output, providing information about the instantaneous external forces acting on Aerobot's wings during flapping wing flight. The governing dynamics are expressed in state-space form, characterized by matrices A_ξ , B_ξ , C_ξ , and D_ξ [11]. These terms are derived from the Wagner indicial model and Prandtl lifting line theory, foundational concepts in fluid dynamics [11]. The efficiency of the indicial model lies in two merits:

- First, indicial models can predict unsteady aerodynamic forces-moments efficiently, which is an important feat for optimization-based control of flying systems.
- Second, indicial models can predict wake structures based on horseshoe vortex shedding. This capability facilitates the description of locomotion gaits using wake structures. Such gaits are extensively utilized in biology to characterize bat aerial locomotion [12], [13].

Next, we briefly discuss the significance of the cascade model described by Eq. 1 from the perspective of actuation in small robots such as Aerobot, which cannot accommodate a large number of power actuators.

B. Paper's Main Contributions

The main goal of this study is to execute intricate 3D flight maneuvers using Aerobot, despite its very limited actuation resources. Achieving such maneuvers necessitates the robot's adept manipulation of its fluidic surroundings

(wing plunge motion, elbow flexion-extension movements, feathering, sweeping, etc.), primarily accomplished through gait control. In Eq. 1, the key functions of control inputs include (1) generating gaits, involving swift movements of cluster joints, and (2) regulating these movements, ensuring precise adjustments within the cluster joints. Dynamic morphing in Aerobot imposes stringent demands on the input vector, encompassing requirements such as power density and the challenge of dealing with small space and high-dimensional joint space.

The cascade model given by Eq. 1 implicitly integrates the responsibility of input u within mechanical intelligence and controls (for a brief discussion see Section III). This means that by incorporating the holonomic constraint $y_1 = h_1(x)$, we create a mathematical framework to systematically split the contributions of input $u = [u_{Reg}^\top, u_{Gen}^\top]^\top$ (see [4], [7]).

While there hasn't been a clear strategy in the literature for systematically splitting actuator roles for motion control in small robots unable to accommodate large actuators, motion control strategies, assuming a uniform role for all actuators (i.e., all entries of the input vector u_i recruited in the same manner), have been common in larger systems like manipulators and legged systems, which face fewer design constraints. This study stands out as it delves into motion control for complex robots with articulated bodies, such as Aerobot, by assigning different roles to actuators. This perspective can reduce the dependency on large actuators, which are challenging to accommodate in small systems, while allowing for motion control characteristics such as precision and speed observed in larger systems.

III. GENERATORS AND REGULATORS

As illustrated in Fig. 2, Aerobot's wings feature numerous moving joints. The central question addressed here is how to mobilize these joints effectively. Two approaches exist for generating and manipulating these joint motions.

- Firstly, we can consider separate, isolated actuated joints. While effective for proximal joints, this approach may impose significant inertial loads and potential failure at distal points [14].
- Alternatively, we can employ a constrained mechanical structure, represented by $y_1 = h_1(x)$, which interconnects joint motions for dynamic motion control. For manipulation of joint motion, the response of the constrained mechanical structure can be adjusted using small, low-power actuators embedded within it.

We propose the utilization of the second approach. The design of Aerobot, as depicted in Fig. 2, is based on this second option. For more details about Aerobot's hardware, readers are referred to [11].

In this study, we aim to showcase how adjusting the positions of regulators with a fast and computationally efficient controller can facilitate the manipulation of correlated joint motions $y_1 = h_1(x)$ in Aerobot, consequently enabling force tracking. We tackle this tracking challenge by employing a

collocation optimization method that enables rapid approximation of the computational structure dynamics. It's imperative to highlight the significance of speed and efficiency in control calculations, especially in dynamic morphing wing flight scenarios where body joint motions occur rapidly, and there is limited space to accommodate powerful computers.

IV. EFFICIENT 3D PATH TRACKING CONTROLS BASED ON COLLOCATION APPROACH

To solve this flight control problem, i.e., Aerobat's posture y_1 is recruited to regulate fluid-structure forces-moments y_2 to track a 3D path, we consider the following cost function given by

$$J = \sum_i^N (z_i - z_{ref,i})^\top C (z_i - z_{ref,i}), \quad z = [\theta_r, \theta_p, \omega^\top]^\top, \quad (2)$$

where θ_r and θ_p represent the robot's roll and pitch angles relative to the inertial frame, respectively, while ω represents the robot's angular velocities, z is the optimization state, z_{ref} is the state reference for z , and C is a diagonal cost weighting matrix. The cost function J is governed by a system of n nonlinear equations representing the computational structure dynamics driven by low-power actuators, as depicted in Fig. 2.

To further elucidate, following the principle of virtual work [15], the response from the computational structure can be determined by

$$\begin{bmatrix} \ddot{y}_{1,1} \\ \ddot{y}_{1,1} \\ \vdots \\ \ddot{y}_{1,n} \\ \ddot{y}_{1,n} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots \\ \vdots & \ddots & \\ a_{2n,1} & & a_{2n,2n} \end{bmatrix} \begin{bmatrix} y_{1,1} \\ \dot{y}_{1,1} \\ \vdots \\ y_{1,n} \\ \dot{y}_{1,n} \end{bmatrix} + \begin{bmatrix} b_{1,1} & b_{1,2} & \dots \\ \vdots & \ddots & \\ b_{2n,1} & & b_{2n,m} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_m \end{bmatrix} \quad (3)$$

where $y_{1,j}$, $j = 1, \dots, n$ denotes the movement from each element of the computational structure, $a_{j,k}$ and $b_{j,k}$ are determined by the physical properties, and ω_j , $j = 1, \dots, m$ is the regulator's input.

By inspecting Eq. 3, it can be observed that the contribution of the input term u based on mode generation and regulation can be separately considered through the design of $a_{j,k}$ (i.e., structure configuration and material properties) and $b_{j,k}$ (regulator or low-power actuator placement), as discussed in [7].

We perform temporal (i.e., t_i , $i = 1, \dots, n$, $0 \leq t_i \leq t_f$) discretization of Eq. 3 to obtain the following system of equations

$$\ddot{Y}_i(t_i) = A_i Y_i(t_i) + B_i \Omega_i(t_i), \quad i = 1, \dots, n, \quad 0 \leq t_i \leq t_f \quad (4)$$

where $Y_i = [y_{1,1}^\top, \dots, y_{1,n}^\top]^\top$ embodies n spatial values of the computational structure response at i -th discrete time (i.e., posture at time t_i). And, $\Omega_i = [\omega_1, \dots, \omega_m]$ embodies

m regulators actions at i -th discrete time. A_i and B_i are the matrices shown in Eq. 3 with their entries.

We stack all of the postures and low-power inputs from the regulators from each i -th sample time, i.e., Y_i and Ω_i , in the vectors $Y = [Y_1^\top(t_1), \dots, Y_n^\top(t_n)]^\top$ and $\Omega = [\Omega_1^\top(t_1), \dots, \Omega_m^\top(t_n)]^\top$.

We consider $2n$ boundary conditions at the boundaries of n structure elements (2 equations at each boundary) to enforce the continuity of the computational structure, given by

$$r_i(Y(0), Y(t_f), t_f) = 0, \quad i = 1, \dots, 2n \quad (5)$$

Since we have m regulators, we consider m inequality constraints given by

$$g_i(Y(t_i), \Omega(t_i), t_i) \geq 0, \quad i = 1, \dots, m, \quad 0 \leq t_i \leq t_f \quad (6)$$

to limit the actuation stroke from the low-power actuators.

To approximate nonlinear dynamics from the computational structure, we employ a method based on polynomial interpolations. This method extremely simplifies the computation efforts.

A. Polynomial Approximation of the Discrete States and Regulator Inputs

Consider the n time intervals during a gait cycle of the dynamic morphing system, as defined previously and given by

$$0 = t_1 < t_2 < \dots < t_n = t_f \quad (7)$$

We stack the states and regulator inputs $Y = [Y_1^\top(t_1), \dots, Y_n^\top(t_n)]^\top$ and $\Omega = [\Omega_1^\top(t_1), \dots, \Omega_m^\top(t_n)]^\top$ from the computational structure at these discrete times into a single vector denoted by $\mathcal{Y}$ and form a decision parameter vector that the optimizer finds at once. Additionally, we append the final discrete time t_f as the last entry of $\mathcal{Y}$ so that gaitcycle time too is determined by the optimizer.

$$\mathcal{Y} = [Y_1^\top(t_1), \dots, Y_n^\top(t_n), \Omega_1^\top(t_1), \dots, \Omega_m^\top(t_n), t_f]^\top \quad (8)$$

We approximate the regulator's action at time $t_i \leq t < t_{i+1}$ as the linear interpolation function $\tilde{\Omega}(t)$ between $\Omega_i(t_i)$ and $\Omega_{i+1}(t_{i+1})$ given by

$$\tilde{\Omega}(t) = \Omega_i(t_i) + \frac{t - t_i}{t_{i+1} - t_i} (\Omega_{i+1}(t_{i+1}) - \Omega_i(t_i)) \quad (9)$$

We interpolate the computational structure states $Y_i(t_i)$ and $Y_{i+1}(t_{i+1})$ as well. However, we use a nonlinear cubic interpolation, which is continuously differentiable with $\tilde{Y}(s) = \mathbf{F}(Y(s), \Omega(s), s)$ at $s = t_i$ and $s = t_{i+1}$, where $\mathbf{F}$ is formed from the dynamics defined in Eq. (1).

To obtain $\tilde{Y}(t)$, we formulate the following system of

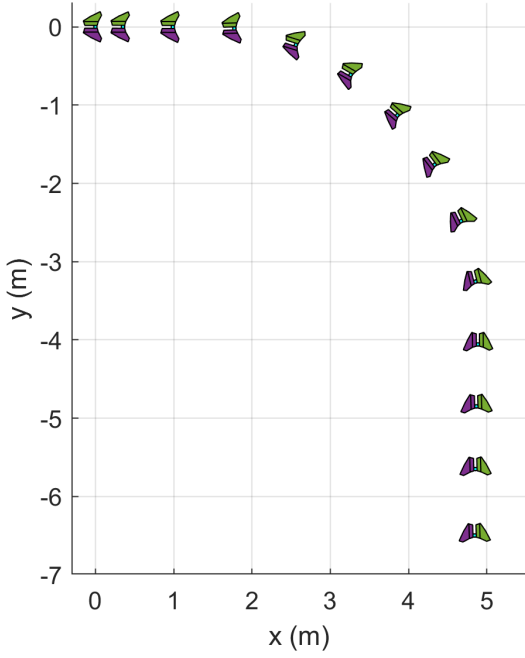


Fig. 3. Top view snapshots of the simulation showing the trajectory of the banking turn.

equations:

$$\begin{aligned} \tilde{Y}(t) &= \sum_{k=0}^3 c_k^j \left(\frac{t-t_j}{h_j} \right)^k, \quad t_j \leq t < t_{j+1}, \\ c_0^j &= Y(t_j), \\ c_1^j &= h_j \mathbf{F}_j, \\ c_2^j &= -3Y(t_j) - 2h_j \mathbf{F}_j + 3Y(t_{j+1}) - h_j \mathbf{F}_{j+1}, \\ c_3^j &= 2Y(t_j) + h_j \mathbf{F}_j - 2Y(t_{j+1}) + h_j \mathbf{F}_{j+1}, \end{aligned} \quad (10)$$

where $\mathbf{F}_j := \mathbf{F}(Y(t_j), \Omega(t_j), t_j)$, $h_j := t_{j+1} - t_j$.

The interpolation function $\tilde{Y}$ utilized for Y needs to fulfill the computational structure's derivative requirements at discrete points and at the midpoint of sample times. By examining Eq. 10, it is evident that the derivative terms at the boundaries t_i and t_{i+1} are satisfied. Hence, the only remaining constraints in the nonlinear programming problem are the collocation constraints at the midpoint of $t_i - t_{i+1}$ time intervals, the inequality constraints at t_i , and the constraints at t_1 and t_f , all of which are included in the optimization process.

Given that the computational structure is spatially discrete and incurs significant costs associated with its curse of dimensionality, this collocation scheme reduces the number of parameters for interpolation polynomials, thereby enhancing computational performance. We address this optimization problem using MATLAB's `fmincon` function.

V. RESULTS

The modeling and simulation of the non-linear dynamical system, as described by the equations presented in Eq. 1, were conducted using MATLAB. The system of equations was solved numerically employing the fourth-order Runge-Kutta method, a well-regarded technique for its accuracy and

computational efficiency. A collocation-based optimization controller, incorporating the complete nonlinear dynamic equation of motion within its predictive framework, was utilized. To enhance computational efficiency, a 5-step prediction horizon was adopted. The simulation was conducted with a time step of 0.0001 seconds, while the collocation controller's update time step was set at 0.005 seconds (200 Hz).

In the simulation, the motor—functioning as a generator—produced flapping gaits at a constant rate of 10 Hz. The regulators adjusted the lengths of links 1, 2, and 3, modifying the flapping gaits to achieve the desired roll and pitch references. Specifically, the pitch reference was maintained at a constant negative 10 degrees throughout the simulation. Meanwhile, the roll reference was initially set to 0 degrees for the first second, adjusted to 15 degrees for the subsequent 2.8 seconds to facilitate the banking turn, and then reverted to 0 degrees for the remainder of the simulation time.

The results elucidated in Fig. 4 and Fig. 5 offer a detailed examination of the system's dynamic states and the regulation of link lengths, alongside the precision of Euler angle tracking. Figure 3, in particular, showcases the collocation controller's proficiency in adhering to predefined roll and pitch angles. It is observed that the robot's pitch fluctuates around a constant -10 degrees with an amplitude in the range of 8 to 10 degrees. This consistent oscillation is caused by the wing's flapping motion.

Moreover, the roll reference tracking is maintained with a high degree of accuracy, facilitating a significant yaw angle alteration that results in a smooth banking turn, as depicted in fig. 3. Over the full trajectory span of 5 seconds, the robot successfully ascended to a height of 1.2 meters and traversed a distance of 4.8 meters in the x-direction and 6 meters in the y-direction. This demonstrates the system's integrated control strategy's capability to achieve complex maneuvers, such as coordinated turns, with remarkable spatial efficiency.

The success of the aerial maneuvers can be attributed to the precise regulation of link lengths within the robot's wing structure, which is essential for modifying the flapping patterns. Figure 5 clearly illustrates the variations in length between the robot's right and left wings. These adjustments result in changes to the shoulder and elbow angles across the wings, consequently leading to asymmetric gaits.

Such asymmetry is crucial for the robot's ability to execute complex maneuvers, like banking turns, by allowing differential aerodynamic forces to be generated on either side of the robot. Figure 6 vividly captures the robot engaging in these asymmetric flapping gaits during such maneuvers. Conversely, when the trajectory necessitates straight-line motion, the robot reverts to symmetrical flapping gaits, demonstrating its adaptable wing movement for effective navigation.

Figure 7 presents the simulated vorticity in the vicinity of the wingtip, capturing the dynamic interaction between the Aerobat and the surrounding air throughout distinct stages of a flapping cycle. Notably, the vorticity is significantly higher during the downstroke, indicative of stronger aerodynamic

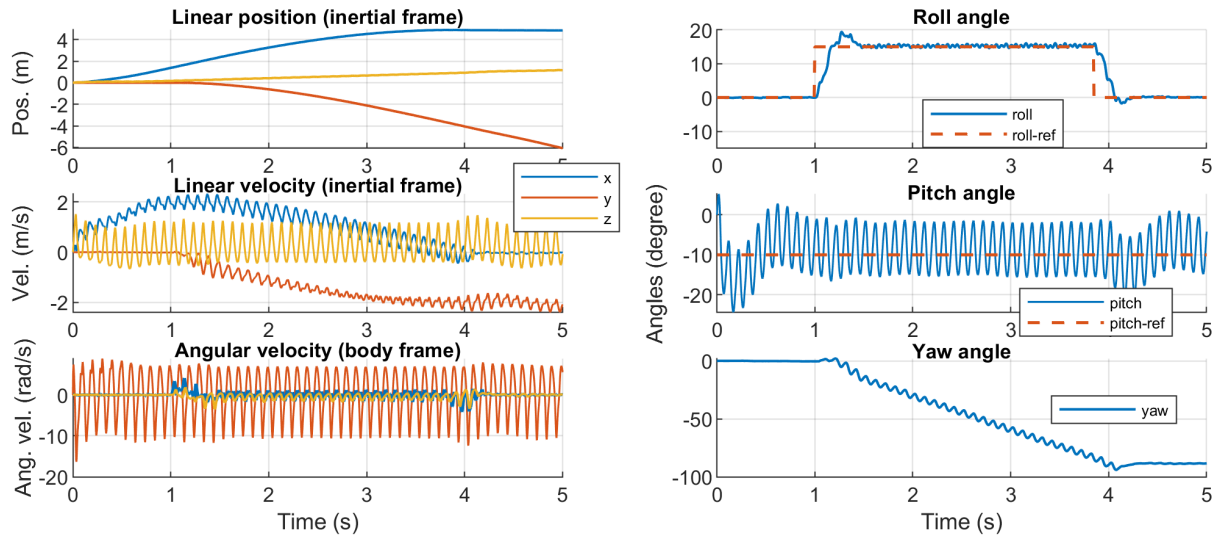


Fig. 4. Plots of the robot's position and orientation states versus time in the simulation. The robot has successfully tracked the desired roll to perform the banking maneuver and generated a steady change in the robot's heading.

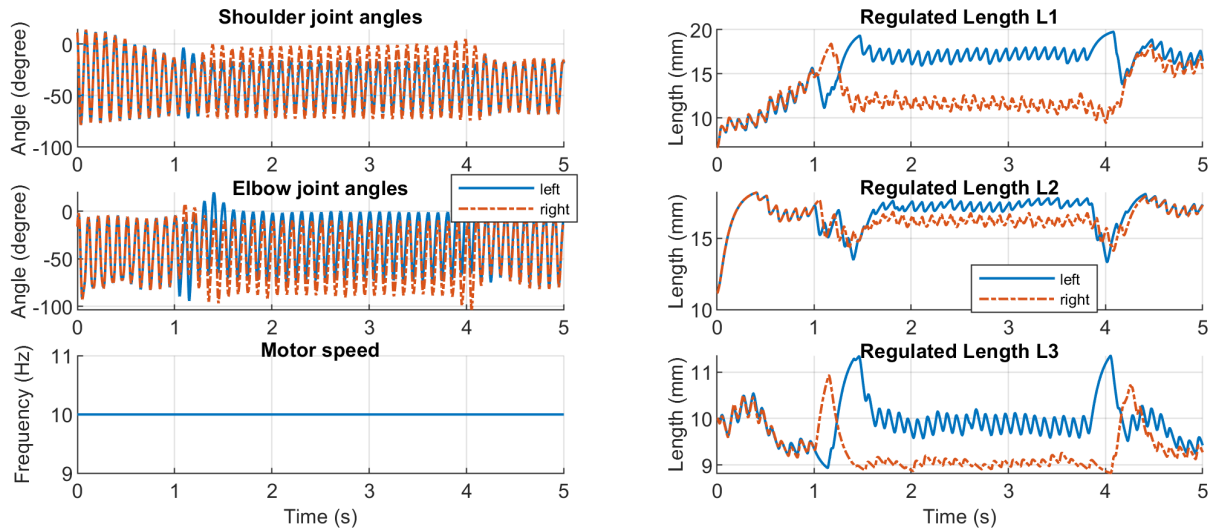


Fig. 5. Plots of the robot's generator trajectories (shoulder and elbow joint angles) and regulator lengths versus time in the simulation. The plots show how the asymmetric flapping gait and regulator actions between the left and right wings that allow the robot to perform the banking maneuver.

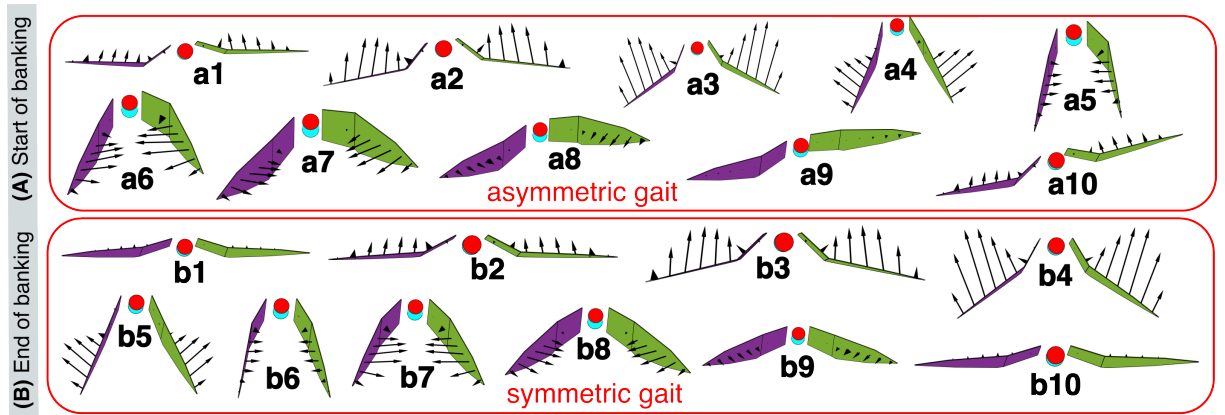


Fig. 6. Simulation snapshots of the frontal view of the robot and the generated aerodynamic forces are shown applied on the wings. The gait is asymmetric during the banking maneuver, causing the robot to bank and turn.

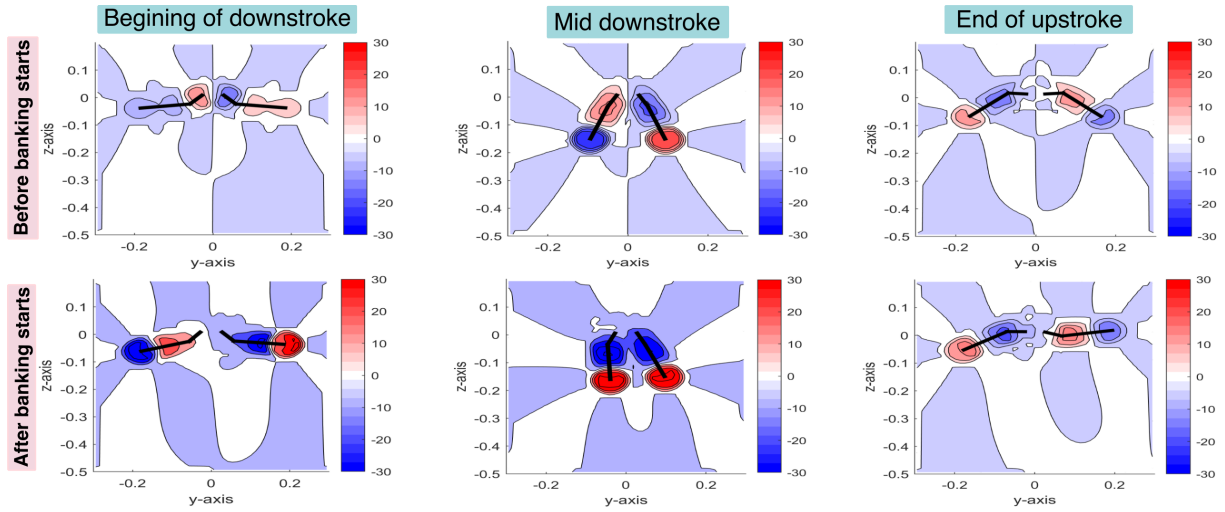


Fig. 7. Phase map plots showing the magnitude of the vortices induced by the flapping gait in rad/s, where the vorticity levels near the wing tip show higher vorticity during downstrokes and lower during upstrokes. Black lines represent Aerobat's wing in frontal view.

forces at play, while it diminishes when the wings fold during the upstroke, reflecting a decrease in aerodynamic activity. Before banking, the vorticity distribution is relatively symmetrical, signifying a steady flight. However, as banking commences, an asymmetrical vorticity pattern emerges, with marked differences between the two sides of the Aerobat. This asymmetry is crucial for the initiation and maintenance of the banking maneuver, altering the lift and drag forces on the wings to achieve the desired roll angle for the turn.

VI. CONCLUDING REMARKS

In this paper, we present our control framework to perform a closed-loop trajectory tracking of a flapping wing robot, the Aerobat. The robot utilizes three regulators on each wing and a single motor to drive and generate the flapping gait for stabilizing the robot's orientation in the air. The application of the collocation control framework was shown in the simulation where we successfully tracked the desired body roll and pitch angles mid-flight to perform the bank turning maneuver. This approach can further be extended to track a predefined flight path for a more practical application such as exploration and reconnaissance. For future work, we will develop and implement the regulator into our existing Aerobat platform and eventually perform untethered experiments to show the feasibility of this controller framework.

REFERENCES

- [1] D. K. Riskin, D. J. Willis, J. Iriarte-Díaz, *et al.*, "Quantifying the complexity of bat wing kinematics," *Journal of Theoretical Biology*, vol. 254, no. 3, pp. 604–615, Oct. 2008.
- [2] D. K. Riskin, A. Bergou, K. S. Breuer, and S. M. Swartz, "Upstroke wing flexion and the inertial cost of bat flight," *Proceedings. Biological Sciences*, vol. 279, no. 1740, pp. 2945–2950, Aug. 2012.
- [3] J. Iriarte-Díaz, D. K. Riskin, D. J. Willis, K. S. Breuer, and S. M. Swartz, "Whole-body kinematics of a fruit bat reveal the influence of wing inertia on body accelerations," *Journal of Experimental Biology*, vol. 214, no. 9, pp. 1546–1553, May 2011.
- [4] A. Ramezani, S.-J. Chung, and S. Hutchinson, "A biomimetic robotic platform to study flight specializations of bats," *Science Robotics*, vol. 2, no. 3, eal2505, Feb. 2017.
- [5] A. Ramezani and E. Sihite, *Aerobat, A Bioinspired Drone to Test High-DOF Actuation and Embodied Aerial Locomotion*, arXiv:2212.05361 [cs, eess], Dec. 2022. DOI: 10.48550/arXiv.2212.05361. [Online]. Available: <http://arxiv.org/abs/2212.05361> (visited on 05/17/2023).
- [6] E. Sihite, X. Hu, B. Li, A. Salagame, P. Ghanem, and A. Ramezani, *Bang-Bang Control Of A Tail-less Morphing Wing Flight*, arXiv:2205.06395 [cs, eess], May 2022. DOI: 10.48550/arXiv.2205.06395. [Online]. Available: <http://arxiv.org/abs/2205.06395> (visited on 05/17/2023).
- [7] E. Sihite, A. Salagame, P. Ghanem, and A. Ramezani, "Actuation and Flight Control of High-DOF Dynamic Morphing Wing Flight by Shifting Structure Response," in *Conference on Decision and Control (CDC)*, Singapore, Dec. 2023.
- [8] A. Dhole, B. Gupta, A. Salagame, *et al.*, *Hovering Control of Flapping Wings in Tandem with Multi-Rotors*, arXiv:2308.00183 [cs, eess], Jul. 2023. DOI: 10.48550/arXiv.2308.00183. [Online]. Available: <http://arxiv.org/abs/2308.00183> (visited on 10/19/2023).
- [9] E. Sihite, P. Kelly, and A. Ramezani, "Computational Structure Design of a Bio-Inspired Armwing Mechanism," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5929–5936, Oct. 2020.
- [10] E. Sihite, A. Darabi, P. Dangol, A. Lessieur, and A. Ramezani, "An Integrated Mechanical Intelligence and Control Approach Towards Flight Control of Aerobat," in *2021 American Control Conference (ACC)*, May 2021, pp. 84–91.
- [11] E. Sihite, P. Ghanem, A. Salagame, and A. Ramezani, "Unsteady aerodynamic modeling of Aerobat using lifting line theory and Wagner's function," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 10 493–10 500.
- [12] T. Y. Hubel, D. K. Riskin, S. M. Swartz, and K. S. Breuer, "Wake structure and wing kinematics: The flight of the lesser dog-faced fruit bat, *Cynopterus brachyotis*," *Journal of Experimental Biology*, vol. 213, no. 20, pp. 3427–3440, Oct. 2010.
- [13] B. Parslew and W. J. Crowther, "Theoretical modelling of wakes from retractable flapping wings in forward flight," *PeerJ*, vol. 1, e105, Jul. 2013.
- [14] E. Sihite and A. Ramezani, "Enforcing nonholonomic constraints in Aerobat, a roosting flapping wing model," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Jeju Island, Korea (South): IEEE, Dec. 2020, pp. 5321–5327.
- [15] L. Landau and E. Lifshitz, *Mechanics: Volume 1*. Elsevier Science, 1982, ISBN: 9780080503479. [Online]. Available: <https://books.google.com/books?id=bE-9tUH2J2wC>.