2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Actuation and Flight Control of High-DOF Dynamic Morphing Wing Flight by Shifting Structure Response

Eric Sihite[1], Adarsh Salagame[2], Paul Ghanem[2], and Alireza Ramezani[2*]

***Abstract*— Bat's dynamically morphing wings are highly versatile with many active and passive modes which allows them to display highly dexterous flight maneuvers. We take inspiration from bat wings and attempt to mimic their high degrees of freedom and flexibility in our small bat robot with dynamically morphing wings called the Aerobat. This small robot uses linkages, or computational structure, to animate the robot's flapping gait. In this work, we present the theoretical framework of using small low-energy actuators, called the primers, to adjust highly sensitive linkages length for changing the robot's flapping gait and use it to control the robot's orientation. This method is applied in a dynamic simulation to show its feasibility.**

I. Introduction

Bats' dynamic morphing wings are known to be extremely high-dimensional, involving the synchronous movements of many active and passive coordinates, joint clusters, in a gaitcycle. These animals apply their unique array of specializations to dynamically morph the shape of their wings to enhance their agility and energy efficiency. Copying bat dynamic morphing wing can bring fresh perspectives to micro aerial vehicle (MAV) design.

For instance, bats employ the combination of inertial dynamics and aerodynamics manipulations to showcase extremely agile maneuvers. Unlike rotary- and fixed-wing systems wherein aerodynamic surfaces (e.g., ailerons, rudders, propellers, etc.) come with the sole role of aerodynamic force adjustments, the articulated wings in bats possess more sophisticated roles [1]. Or, it is known that bats can perform zero-angular-momentum turns by making differential adjustments (e.g., collapsing armwings) in the inertial forces led by their wings. Bats can apply a similar mechanism to perform sharp banking turns [2], [3].

However, unfortunately, copying bat dynamic morphing wing flight is a significant ordeal. Due to the challenges associated with hardware design and control, existing bioinspired MAV designs completely overlook bat dynamic morphing capabilities. As a result, much attention has been paid to simpler forms of animal aerial locomotion, such as those from insects. While the mathematical models of insect-inspired robots of varying size and complexity are relatively well developed, models of airborne, fluidic-based vertebrate locomotion, their control, and high-dimensional actuation remain largely open to date. The mainstream school of thought

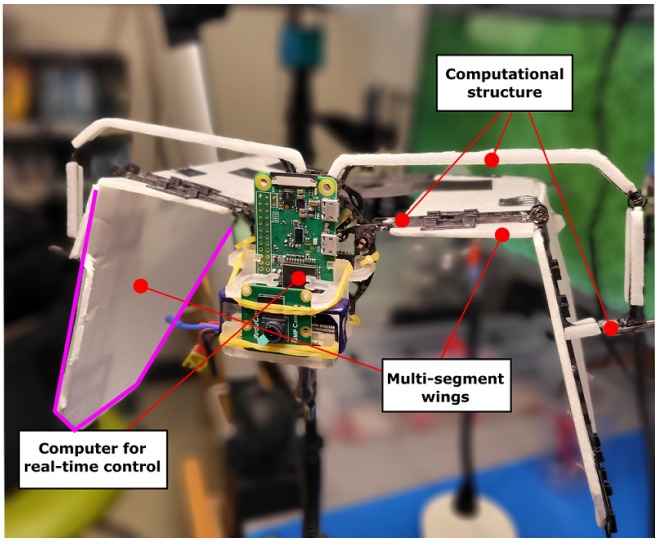

Fig. 1. Shows Northeastern's Aerobat platform, designed to study articulated flapping wing flight and high-dimensional actuation in small micro aerial vehicles. Each wing has two segments and each segment dynamically translates with respect to the other segment.

inspired by insect flight has conceptualized wing as a massless, rigid structure, which is nearly planar and translates – as a whole or in two-three rigid parts – through space [4]–[6]. In this view, wings possess no inertial effect, yield two-time-scale dynamics [7], permit quasi-static external force descriptions [8], and produce a tractable dynamical system. Unfortunately, these paradigms fail to provide insight into airborne, vertebrate locomotion and lack the ingredients of a more complete and biologically meaningful model.

So, the overarching objective of our efforts is to present a systematic method for high-dimensional actuation in MAVs based on the computational structure design and the optimal placement of low-power actuators, *'primers'*, within computational structures [9]–[11]. Note that computational structures (also called mechanical intelligence or computational morphology) are mechanical structures that deliver computational resources. The main contribution of this work is to demonstrate the feasibility of embodied aerial locomotion through simulations. Embodied locomotion is a notion that, despite its endorsement by the robotics community it has remained unexplored in morphing MAV design.

By employing embodiment to achieve high-dimensional actuation in MAVs, we specifically aim to generate several active body joints with low-power actuators. Then, we utilize these low-power actuators to stabilize the flight dynamics

[1] The author is with the Department of Aerospace, California Institute of Technology, Pasadena, CA-91125, USA. (e-mail: esihite@caltech.edu).
[2] The author is with the SiliconSynapse Laboratory, Department of Electrical and Computer Engineering, Northeastern University, Boston, MA-02119, USA. (e-mail: salagame.a, ghanem.p, a.ramezani@northeastern.edu).
* Corresponding author's e-mail: a.ramezani@northeastern.edu.

through closed-loop feedback actively. Northeastern's *Aerobat* platform, shown in Fig. 1, has allowed us to test dynamic morphing wing flight in realistic flight scenarios comparable to bat flights. Because of prohibitive design restrictions such as limited payload and power budget, applying classical joint motion control based on sensing, processing, and actuation is infeasible in Aerobat.

The framework we propose in this work has allowed the fast activation and regulation of many actuated degrees of freedom (DOF). We aim to inspect the feasibility of gait regulation in Aerobat on untethered flights. For now, we limit the scope of this work to simulation results. This work is organized as follows: a section outlining the dynamic and aerodynamic model used in the control design and simulation, followed by a brief section outlining the actuation method, then the control section detailing the collocation scheme, simulation results, discussion, and concluding remarks.

II. Dynamical Model

The flight dynamics of Aerobat shown in Fig 1 can be described with a nonlinear system [12] of the following form:

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

where t, x, and ξ denote time, the state vector, and hidden aerodynamic variables, respectively. In Eq. 1, the state vector x embodies the position and velocity of the active q_a and passive q_p coordinates. The nonlinear terms given by $f(x)$ and $g_1(x)$ are obtained from Lagrange equations and embody inertial, Coriolis and gravity terms. 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 $\dot{x}$, respectively.

The aerodynamic force output denoted by y_2 gives the instantaneous external forces. The governing dynamics are given by the state-space form made of A_ξ, B_ξ, C_ξ, and D_ξ matrices [12]. These terms are obtained based on Wagner indicial model and Prandtl lifting line theory reported in the fluid dynamics textbook. The benefit of this indicial model is that it allows to compute the wake structures based on horseshoe vortex shedding efficiently. As a result, it allows using wake structures to describe locomotion gaits. Wake-structure-based gaits are widely used in biology to describe bat aerial locomotion [13], [14].

Gait generation and regulation are two main contributions of control inputs in Eq. 1. Dynamic morphing enforces tight requirements (such as power density and curse of dimensionality) on the input vector. By considering the holonomic constraint $y_1 = h_1(x)$ we leave room for ourselves to be able to dichotomize the contributions from input u [15]. Meaning, we can systematically determine which actuator generates and which one regulates the gait. So far there has been no clear strategy in the literature regarding how these contributions can be systematically assigned to the actuators in a locomotion system. Bats majorly inspire our actuation view [16]. In their membraned flight apparatus, bats possess specialized power and steering muscles to generate and regulate gaits. Joint motion control, by assuming a similar role for all actuators, has been widely utilized in large systems such as manipulators and legged systems, which possess less prohibitive design restrictions. However, joint motion control based on dividing actuators' roles has remained unexplored.

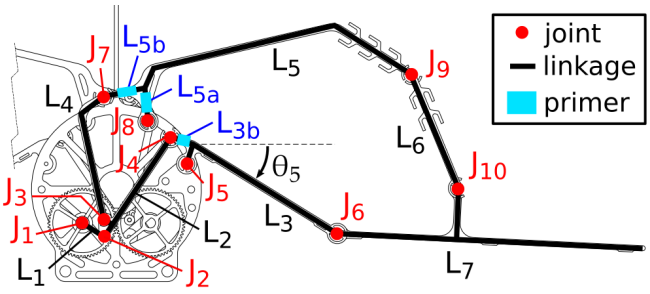

Fig. 2. Shows the computational structure used in Aerobat and the assumed low-power actuators' locations within the structure. In the simulation results presented in this paper, the primers are located at the linkage L_3 and L_5, denoted by the segment lengths L_{3a}, L_{5a}, and L_{5b}.

III. Brief Overview of Actuation

As shown in Fig. 2, the wing joints dynamically change during each gaitcycle. There are two options to generate and manipulate the joint motions. First, we can assume separate isolated actuated joints. While this option works for proximal joints, it can lead to large inertial loads and failure at this distal points. Second, a constrained mechanical structure (i.e., $y_1 = h_1(x)$) that interconnects joint motions can be assumed for dynamic joint motions. For joint motion manipulations, the response of the constrained mechanical structure can be shifted using small low-power actuators embedded within the constrained mechanical structure.

The above paragraph in simple words describes the underlying principles of actuation in dynamic morphing wing flight. To do this, we design closed-kinematic chains using optimization [9]. These kinematic chains are called computational structures because they carry computational resources by correlating joint motions. In other words, the sensing, actuation, and computation effort needed to be taken in classical closed-loop motion of joints in the wing is replaced with the roles from these mechanical structures.

The forced response of these computational structures to a periodic excitation can be shifted using small adjustments in the physical properties of the structure. The physical property that we consider is linkage length. We identify locations within the computational structure such that with minimum actuation power, large changes in the response from the computational structure are observed [10], [11].

Next, the overall control design idea is based on activating these low-power actuators in order to manipulate correlated joint motions $y_1 = h_1(x)$ and consequently employ that

for force tracking. We resolve this tracking problem using a collection method that allows approximation of the computational structure dynamics rapidly. Note that the speed of control calculations is important as dynamic morphing wing flight involves fast body joint motions.

IV. Control

Several control design frameworks can be employed based on our view to actuate dynamic morphing wing flight. The approach we consider in this paper is based on the collocation technique.

Our objective is to find the commands for the low-power actuator such that the desired force $y_{2,d}$ needed to push the rates of changes in the Euler angles (indication of flight stability) to zero are tracked. Note that tail-less flapping flight is open-loop unstable [17]. We consider the following cost function given by

$$J = \sum_{i}^{N} \left(c_1 \|R(x_i) - I\|^2 + c_2 \|\omega(x_i)\|^2 \right) \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm, $R(x)$ is the robot's rotation matrix relative to the inertial frame, $\omega(x)$ is the robot's angular velocities, c_1 and c_2 are the cost weighting. The cost function J is subject to a system of n nonlinear equations that embody the computational structure dynamics driven by low-power actuators shown in Fig. 2. Following the principle of virtual work, the response from the computational structure is given by

$$\begin{bmatrix} \dot{y}_{1,1} \\ \ddot{y}_{1,1} \\ \vdots \\ \dot{y}_{1,n} \\ \ddot{y}_{1,n} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots \\ \vdots & \ddots & \\ a_{n1} & & a_{nn} \end{bmatrix} \begin{bmatrix} y_{1,1} \\ \dot{y}_{1,1} \\ \vdots \\ y_{1,n} \\ \dot{y}_{1,n} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots \\ \vdots & \ddots & \\ b_{n1} & & b_{nn} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_n \end{bmatrix} \tag{3}$$

where $y_{1,i}$ denotes the response from each element of the computational structure. By inspecting Eq. 3, it can be seen that the input term u contribution based on mode generation and regulation can be separately considered through the design of a_{ij} (structure configuration and material properties) and b_{ij} (low-power actuator placement). We perform temporal discretization of Eq. 3 to obtain the following system of equations

$$\dot{Y}_i(t) = A_i Y_i(t) + B_i \Omega_i(t), \quad i = 1, \dots, n, \quad 0 \le t \le t_f \tag{4}$$

where Y_i embodies all of the spatial values of the computational structure response at i-th discrete time (i.e., posture). And, Ω_i embodies all of the low-power-actuator actions at i-th discrete time. A_i and B_i are the matrices shown in Eq. 3 with their entries. We consider 2n boundary conditions given by

$$r_i \left(Y(0), Y\left(t_f\right), t_f \right) = 0, \quad i = 1, \dots, k \le 2n \tag{5}$$

and m inequality constraints given by

$$g_i(Y(t), \Omega(t), t) \ge 0, \quad i = 1, \dots, m, \quad 0 \le t \le t_f \tag{6}$$

to limit the actuation stroke from the low-power actuators. We stack all of the states and low-power inputs from the computational structures in the vectors $Y = \left[Y_1^\top(t), \dots, Y_l^\top(t) \right]^\top$ and $\Omega = \left[\Omega_1^\top(t), \dots, \Omega_l^\top(t) \right]^\top$. To approximate nonlinear dynamics from the computational structure, we employ a method based on polynomial interpolations. This method extremely simplifies the computation efforts.

Consider the n time intervals during a gaitcycle of the dynamic morphing systems given by

$$0 = t_1 < t_2 < \ldots < t_n = t_f \tag{7}$$

where t_i denotes discrete times. Then, we stack the states Y_i and low-power Ω_i input from the computational structure at these discrete times in a single vector denoted by $\mathcal{Y}$. Also, we add final discrete time t_f as the last entry of $\mathcal{Y}$,

$$\mathcal{Y} = \left(\Omega\left(t_1\right), \dots, \Omega\left(t_n\right), Y\left(t_1\right), \dots, Y\left(t_n\right), t_f \right) \tag{8}$$

We take the low-power input to be as the linear interpolation function between $\Omega(t_i)$ and $\Omega(t_{i+1})$ for $t_i \le t < t_{i+1}$.

$$\Omega_{\text{int}}(t) = \Omega\left(t_i\right) + \frac{t - t_i}{t_{i+1} - t_i} \left(\Omega\left(t_{i+1}\right) - \Omega\left(t_i\right) \right) \tag{9}$$

We interpolate the computational structure states $Y(t_i)$ and $Y(t_{i+1})$ too. However, we take a nonlinear cubic interpolation which is continuously differentiable with $\dot{Y}_{\text{int}}(s) = f(Y(s), \Omega(s), s)$ at $s = t_i$ and $s = t_{i+1}$. To do this, we write the following system of equations:

$$\begin{aligned} Y_{\text{int}}(t) &= \sum_{k=0}^{3} c_k^j \left(\frac{t - t_j}{h_j} \right)^k, \quad t_j \le t < t_{j+1}, \\ c_0^j &= Y\left(t_j\right), \\ c_1^j &= h_j f_j, \\ c_2^j &= -3Y\left(t_j\right) - 2h_j f_j + 3Y\left(t_{j+1}\right) - h_j f_{j+1}, \\ c_3^j &= 2Y\left(t_j\right) + h_j f_j - 2Y\left(t_{j+1}\right) + h_j f_{j+1}, \\ \text{where } f_j &:= f\left(Y\left(t_j\right), \Omega\left(t_j\right), t_j\right), \quad h_j := t_{j+1} - t_j. \end{aligned} \tag{10}$$

The interpolation function used for Y must satisfy the derivative of the computational structure at the discrete points and at the middle of sample times, that is, $t_{c,i}$. By inspecting Eq. 10, it can be seen that the derivative terms at the boundaries t_i and t_{i+1} are satisfied. Therefore, the only remaining constraints in the nonlinear programming constitute the collocation constraints at the middle of $t_i - t_{i+1}$ time interval, the inequality constraints at t_i, and the constraints at t_1 and t_f. These constraints are given by:

$$\begin{aligned} f\left(Y_{\text{int}}\left(t_{c,i}\right), \Omega_{\text{int}}\left(t_{c,i}\right), t_{c,i}\right) - \dot{Y}_{\text{int}}\left(t_{c,i}\right) &= 0 \\ g\left(Y_{\text{int}}\left(t_i\right), \Omega_{\text{int}}\left(t_i\right), t_i\right) &\ge 0 \\ r\left(Y_{\text{int}}\left(t_1\right), Y_{\text{int}}\left(t_N\right), t_N\right) &= 0 \end{aligned} \tag{11}$$

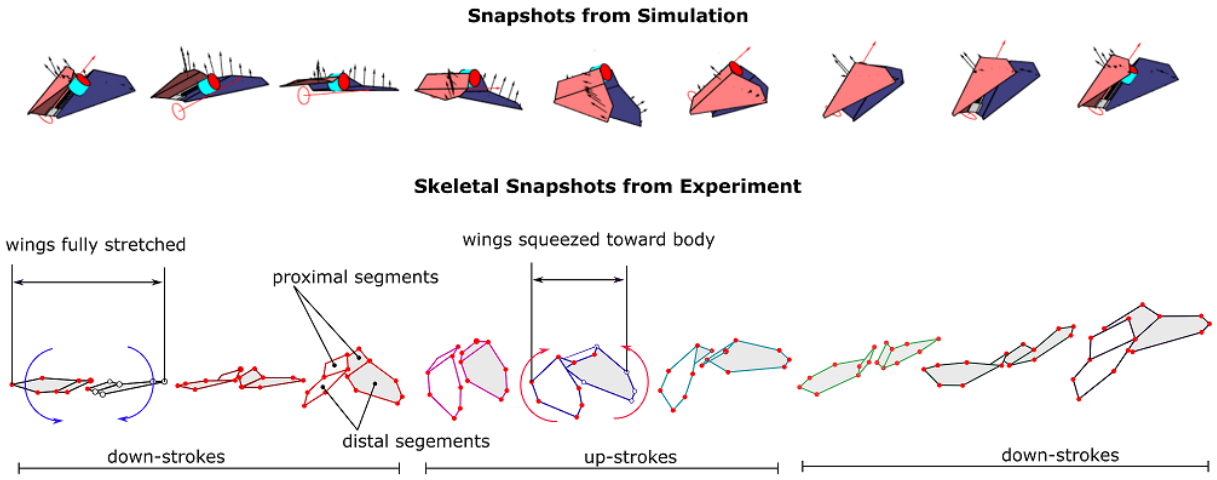

Fig. 3. Snapshots of experimental bounding flight and its comparison to simulated snapshots from Eq. 1.

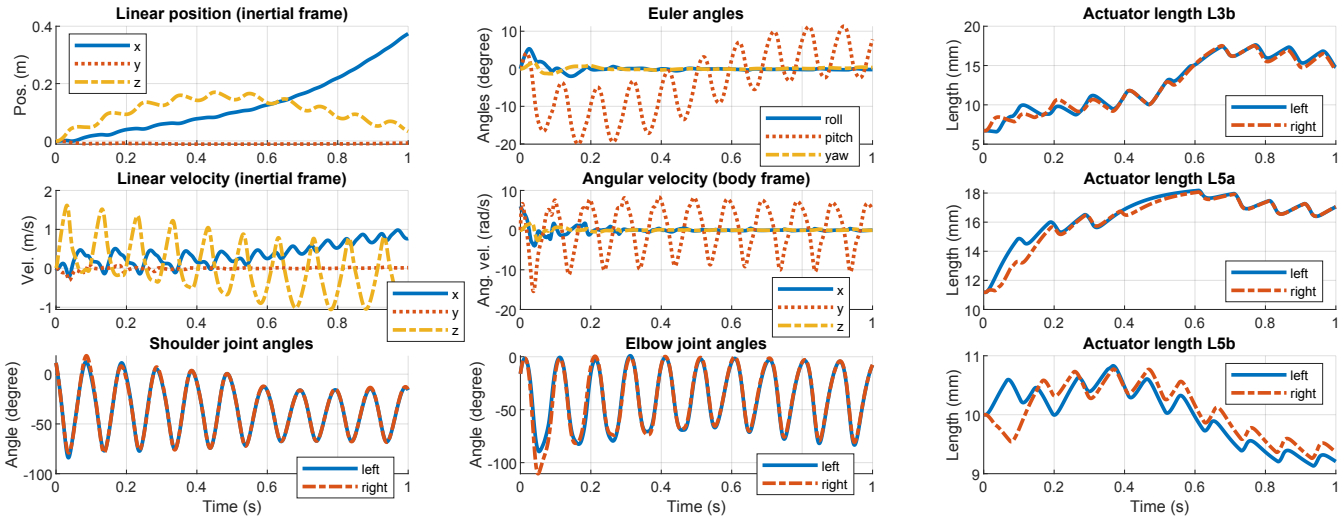

Fig. 4. Simulated response of the small actuators rejecting the initial roll rate disturbance of 6 rad/s. The small actuators adjusted the length of the linkage segment to alter the flapping gait, resulting in an asymmetric forces that generate the necessary wrench to stabilize the initial disturbance. This disturbance was rejected within 2 or 3 gait cycles which is relatively quick considering that the robot was flapping at a frequency of 10 Hz.

Given that the computational structure is spatially discrete with large costs associated with its curse of dimensionality, this collocation scheme results in a smaller number of parameters for interpolation polynomials which enhance the computation performance. We resolve this optimization problem using MATLAB fmincon function.

V. Results

The simulation was performed in Matlab using the 4th-order Runge-Kutta method to march the system equation of motion (1) forward in time. We used the simulation time step of 0.0001s and the collocation controller update time step of 0.005s. The controller used the full nonlinear dynamic equation of motion in its prediction steps, so we used 5 step prediction horizon to reduce the computational time. The robot was initialized with 6 rad/s initial roll rate, which will be stabilized using the low-power actuators and the control framework described in (11). The wing was set to flap at a constant rate of 10 Hz, and we only used the small wing actuator to adjust the wing's flapping gait to reject the roll disturbance. There are also other external inputs to the system, such as the aerodynamic damping acting on the body, which helps the system to dampen the oscillations as the wing flaps.

Figures 3, 4 and 5 show the simulated input and output response of the orientation stabilization, which was done by adjusting the computational structure using the low-power actuators. The simulation shows that the initial roll rate disturbance was successfully rejected and stabilized in approximately 2 seconds. The discrepancy between the actuation of the left and right wings generated the required wrench to reject this initial disturbance. This discrepancy can be seen in Fig. 5, showing an asymmetric flapping gait within the first few gait cycles to reject the initial roll rate disturbance.

The simulated pitch was highly oscillatory and we had

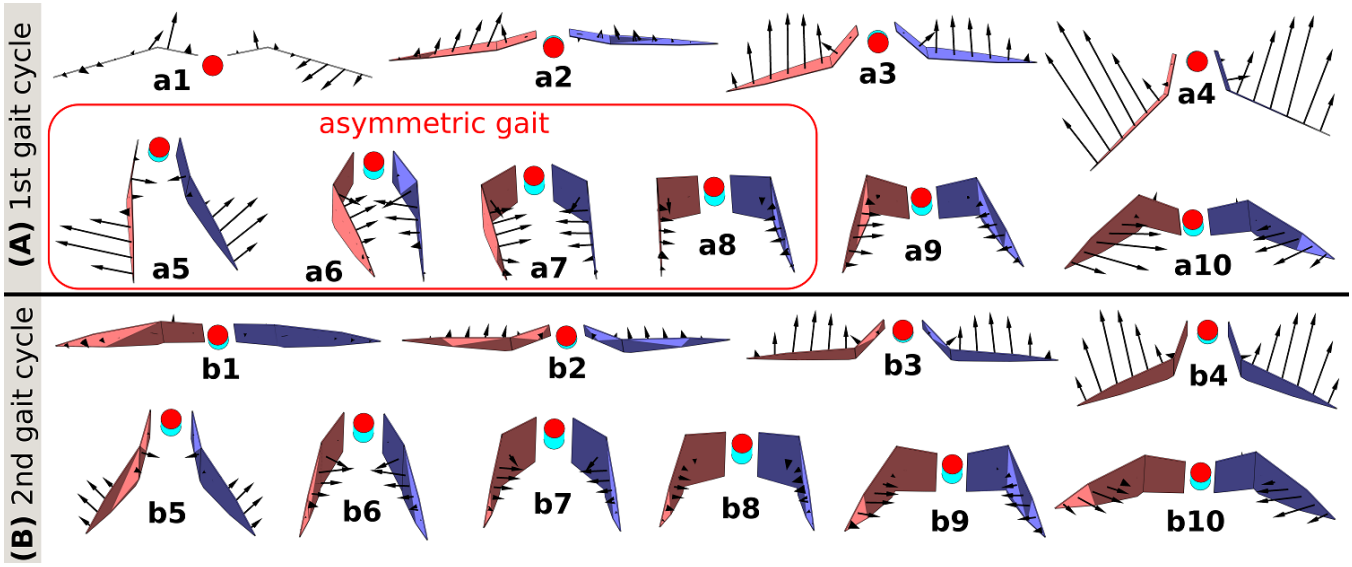

Fig. 5. The front-view illustrations of the robot states and aerodynamic forces in the simulation shown in Fig. 4 during its first two gait cycles. During the first gait cycle, the actuators altered the wing flapping gait to generate the asymmetric gait shown in (a5) to (a8) and generated a rolling moment to reject the initial disturbance. The gait then returned to a mostly symmetric gait in the following gait cycles.

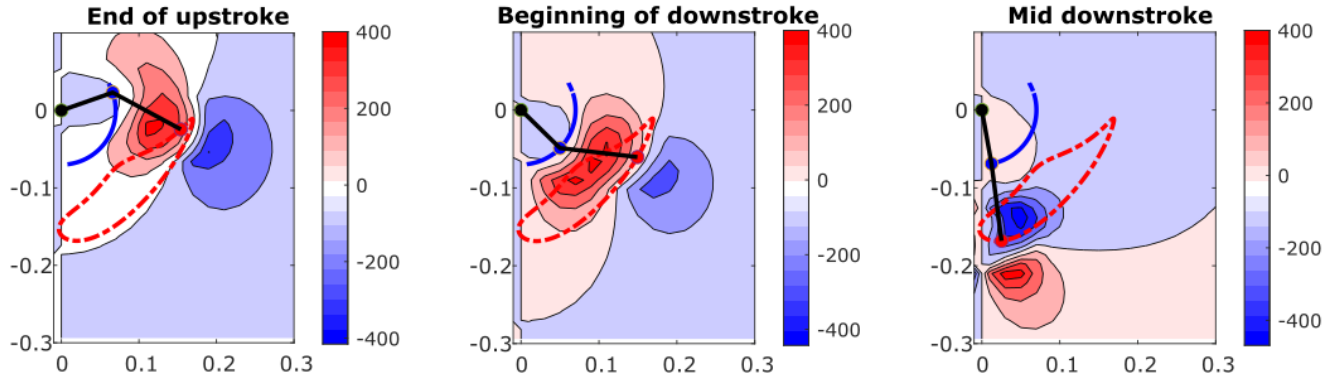

Fig. 6. Shows the simulated cross-sectional vorticity near the wing tip at various timings within a gait cycle. The vorticity is larger during the downstroke and much smaller when the wing is folded during the upstroke. The solid black lines denote the segmented wing of Aerobat in the frontal plane (front view) of flight. Dashed red lines denote the wing tip trajectories.

little control over the pitch dynamics using the actuation method proposed in Section III. This actuation, in addition to the planar kinematic linkages design used in the robot, can only adjust the flapping gait in the robot's $x-z$ plane. This means that we cannot change the center of pressure in the front-rear directions, which effectively makes us incapable of adjusting the pitching moment. As shown in Fig. 4, the wing flapping caused the robot's pitch to oscillate with an amplitude of approximately 10 to 15 degrees.

Figure 6 shows the simulated cross-sectional vorticity near the wing tip at various timings within a gait cycle. The vorticity is larger during the downstroke and much smaller when the wing is folded during the upstroke. The vorticity near wing tips is obtained from Eq. 3 and it describes Aerobat-environment interactions as dynamic morphing wing flight takes place.

VI. Concluding Remarks

Actuation problems can be very challenging in small robots, particularly MAVs. In MAV design, prohibitive design restrictions do not allow the incorporation of many large actuators in the systems. This limitation can pose challenges regarding generating dynamic joint motions in MAVs. We considered Northeastern's Aerobat platform design which was created to inspect dynamic morphing wing flight. Aerobat possesses computational structures with low-power actuators in its design. We employed these low-power actuators to design an optimization-based control policy stabilizing Aerobat's unstable flight dynamics. We considered the cascade model of Aerobat from our previous works. This cascade mode embodies the dynamic of computational structures in Aerobat and aerodynamics interactions. Using polynomial interpolations, we employed a collocation-based method to approximate Aerobat's computational structure

dynamics. This collocation approach allowed us rapidly calculate the inputs required to track the desired aerodynamic forces for flight stabilization in simulation. For our future work, we will implement this method in our bat robot shown in Fig. 1, and test it in a wind tunnel or outdoor experiments.

REFERENCES

[1] D. K. Riskin, J. W. Bahlman, T. Y. Hubel, J. M. Ratcliffe, T. H. Kunz, and S. M. Swartz, "Bats go head-under-heels: The biomechanics of landing on a ceiling.," *The Journal of experimental biology*, vol. 212, no. Pt, pp. 945–953, 2009.

[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-Diaz, 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] E. Farrell Helbling and R. J. Wood, "A Review of Propulsion, Power, and Control Architectures for Insect-Scale Flapping-Wing Vehicles," *Applied Mechanics Reviews*, vol. 70, no. 1, Jan. 2018.

[5] J. Zhang, B. Cheng, and X. Deng, "Instantaneous wing kinematics tracking and force control of a high-frequency flapping wing insect MAV," *Journal of Micro-Bio Robotics*, vol. 11, no. 1, pp. 67–84, Jun. 2016.

[6] M. Karpelson, Gu-Yeon Wei, and R. J. Wood, "A review of actuation and power electronics options for flapping-wing robotic insects," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 779–786.

[7] F. Bullo, R. M. Murray, and A. Sarti, "Control on the Sphere and Reduced Attitude Stabilization," *IFAC Proceedings Volumes*, 3rd IFAC Symposium on Nonlinear Control Systems Design 1995, Tahoe City, CA, USA, 25-28 June 1995, vol. 28, no. 14, pp. 495–501, Jun. 1995.

[8] D. Lentink and M. H. Dickinson, "Biofluiddynamic scaling of flapping, spinning and translating fins and wings," *Journal of Experimental Biology*, vol. 212, no. 16, pp. 2691–2704, Aug. 2009.

[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 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.

[11] E. Sihite, A. Darabi, P. Dangol, A. Lessieur, and A. Ramezani, "An Integrated Mechanical Intelligence and Control Approach Towards Flight Control of Aerobat," *arXiv:2103.16566 [cs, eess]*, Mar. 2021.

[12] E. Sihite, P. Ghanem, A. Salagame, and A. Ramezani, "Unsteady aerodynamic modeling of aerobat using lifting line theory and wagner's function," *arXiv preprint arXiv:2207.12353*, 2022.

[13] 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.

[14] B. Parslew and W. J. Crowther, "Theoretical modelling of wakes from retractable flapping wings in forward flight," *PeerJ*, vol. 1, e105, Jul. 2013.

[15] A. Ramezani, S.-J. Chung, and S. Hutchinson, "A biomimetic robotic platform to study flight specializations of bats," *Science Robotics*, vol. 2, no. 3, eaal2505, Feb. 2017.

[16] J. Iriarte-Diaz and S. M. Swartz, "Kinematics of slow turn maneuvering in the fruit bat Cynopterus brachyotis.," *The Journal of experimental biology*, vol. 211, no. Pt, pp. 3478–3489, 2008.

[17] M. Sun, J. Wang, and Y. Xiong, "Dynamic flight stability of hovering insects," *Acta Mechanica Sinica*, vol. 23, no. 3, pp. 231–246, 2007.