

Mechanism Design and Control of a Winged Hovering Robot with Flapping Angle Constraint

Hamid Vejdani*, LaRance Haji, Vernon Fernandez, Badih Jawad

Bio-Inspired Robotics and Dynamics (BIRD) lab
Mechanical, Robotics, and Industrial Engineering Department
Lawrence Technological University
Southfield, Michigan
Email: hvejdani@ltu.edu

In this paper, we first presented a four-bar linkage mechanism for actuating the wings in a flapping wing flying robot. After that, given the additional constraints imposed by the four-bar linkage, we parameterized the wing kinematics to provide sufficient control authority for stabilizing the system during 3D hovering. The four-bar linkage allows the motors to spin continuously in one direction while generating flapping motion on the wings. However, this mechanism constrains the flapping angle range which is a common control parameter in controlling such systems. To address this problem, we divided each wingbeat cycle into four variable-time segments which is an extension to previous work on split-cycle modulation using wing bias but allows the use of a constant flapping amplitude constraint for the wing kinematic. Finally, we developed an optimization framework to control the system for fast recovery while guaranteeing the stability. The results showed that the proposed control parameters are capable of creating symmetric and asymmetric motions between the two wings and therefore, can stabilize the hovering system with minimal actuation and flapping angle amplitude constraint.

Keywords: flapping wing robot, hovering, four-bar linkage mechanism

1 Introduction

Flapping wing micro aerial vehicles (MAVs) have received significant attention due to the outstanding performance of their biological counterparts (insects, bats and birds) [1, 2]. However, flapping flight is inherently unstable, and thus one of the key challenges in designing these robots is the trade-off between control authority, weight and system complexity. In nature, highly-articulated wings in aerial fliers such as bats provide redundant possibilities in choosing control inputs. However, a robot with too many actuators would lead to increased weight and structural complexity. Therefore, recognizing the dominant wing motions in natural

fliers (like flapping motion), and exploiting that motion to its utmost potential during a flight mode is critical in designing efficient MAVs. Moreover, the flapping (forward and backward) motion on the wings observed in natural fliers is not preferred to be directly generated by electric motors and requires proper mechanical mechanisms to transform the unidirectional motions from the motors into the desired motion on the wings. In this work we address these issues by using a four-bar linkage mechanism for a flapping wing system and then proposing a new wing kinematic parametrization to solve the restraining problem imposed by the four-bar linkage mechanism.

Although lightweight and powerful actuators are becoming more affordable, Wood *et al.* [3] and Tu *et al.* [2] have pointed out that the actuator can comprise 50 - 60% of the total vehicle weight. Recently, He *et al.* [4] introduced a bird-scale flapping wing robot that used 3 actuators, one used to flap the wings and two more to actuate the tail. Although they showed successful forward flight, stable operation proved to be challenging, due to the lack of airflow over the tail. They actuated the flapping of both wings by a four-bar linkage with one motor and a tail controlled by two servo motors that by itself took 20% of the weight. Although this robot can accomplish forward flight where airflow allows the tail to control the vehicle, due to lack of forward speed, stabilizing it for hovering (or low forward velocities) would be a challenge. Ramezani *et al.* [1] designed a tailless bat-scale flapping wing robot equipped with 5 actuators and demonstrated remarkable capabilities including banking and turning maneuvers. However, partly due to the number of actuators and hence the weight of the robot, the authors didn't present hovering flight. Doman and Oppenheimer *et al.* [5, 6] stabilized an insect-scale hovering system by introducing asymmetric duration between downstroke and upstroke while keeping the total period constant. Using a more rapid or slower upstroke, they showed it to be effective in controlling the body except for pitch DoF for which they used a bob-weight. Later, Oppenheimer *et al.* [6] extended their previous parameterization, adding wing bias

*Address all correspondence to this author.

to the wingbeat function to eliminate the need of the bob-weight in the pitch control - effectively demonstrating that the vehicle could be controlled using as few as two actuators. However, in their formulation, for the wing forcing functions to match across cycles, the flapping amplitude of each cycle must vary and be used as a control parameter. In a practical manner, this can only be implemented through the use of a motor undergoing constant braking in order to reverse the direction of spinning, wasting kinetic energy as heat. This approach has been used in designing the hummingbird-scaled robot in [2]. Tijmons *et al.* [7], designed a mechanical mechanism for their DelFly II to enhance the controllability of the two wings and made it tailless. However, the number of actuators did not reduce significantly while the mechanical mechanism added extra weight.

In this paper, we first introduce a four-bar linkage kinematic design to transform a unidirectional movement of the motor to a flapping motion on the wing. Then, we propose a new wing kinematic function by parameterizing a wingbeat cycle and then explore the possibility of controlling a flapping wing robot with the minimum number of actuators. Notably different than [6], the flapping amplitude does not need to vary, which enables the usage of the four-bar linkage to achieve the flapping motion.

2 Mechanism Design

We design a proper four-bar linkage mechanism to transform the unidirectional rotation of the motors into appropriately desired flapping motion on the wing. Ideally, the kinematic goal is to generate a flapping motion on the wing that generates balanced aerodynamic forces and torques after each wingbeat cycle with a constant motor velocity. To stabilize such a system against unknown perturbations, the motor velocity changes around that nominal speed. Therefore, the wing's upstroke and downstroke motions should be as symmetric as possible not only with respect to each other, but also with respect to their own mid-strokes. To achieve this goal, the distribution of wing angular velocity over the downstroke and upstroke angles ($\dot{\phi}_i - \phi_i$) are designed to be as symmetric as possible. For the kinematic formulations we consider figure 1 as the schematic of the mechanism. The γ angle can be calculated based on the geometry of the mechanism as: $\gamma = \tan^{-1}(\frac{L_1 \cos \beta}{L_4 - L_1 \sin \beta})$. Then the wing gear angle (ϕ_i) can be obtained:

$$\phi_i = -\sin^{-1} \frac{-L_2^2 + W^2 + L_3^2}{2WL_3} + \gamma \quad (1)$$

where $W = \sqrt{L_1^2 + L_4^2 - 2L_4L_1 \sin \beta}$. To find the wing angular velocity we need to differentiate equation 1 with respect to time in which we will need $\dot{\gamma}$ and $\dot{W}$ from the following equations:

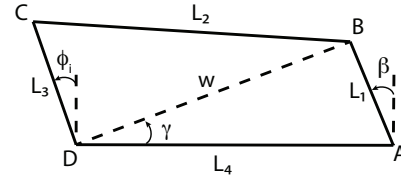


Fig. 1. Four-bar linkage schematic.

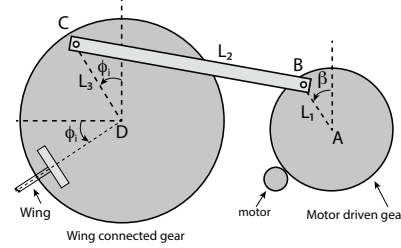


Fig. 2. Wing to motor connection mechanism.

$$\dot{\gamma} = \frac{\cos^2 \gamma (L_1^2 - L_1 L_4 \sin \beta)}{(L_4 - L_1 \sin \beta)^2} \dot{\beta} = G_1(\beta) \dot{\beta}$$

$$\dot{W} = \frac{-L_4 L_1 \cos \beta}{\sqrt{L_1^2 + L_4^2 - 2L_4 L_1 \sin \beta}} \dot{\beta} = G_2(\beta) \dot{\beta}$$

Using these equations, the wing angular velocity can be found as:

$$\dot{\phi}_i = \frac{-4W^2 L_3 \dot{W} + 2L_3 \dot{W} (-L_2^2 + L_3^2 + W^2)}{4W^2 L_3^2 \cos(\gamma - \phi_i)} + \dot{\gamma} = G(\beta) \dot{\beta} \quad (2)$$

After formulating the kinematic equations, an optimization program is constructed to find the optimal dimensions for the links (L_1, L_2, L_3, L_4). This optimization framework minimizes the upstroke and downstroke asymmetry in $(\dot{\phi} - \phi)$ graph (Fig. 3) with $J_{design} = \int (\dot{\phi}_{US}(\phi) + \dot{\phi}_{DS}(\phi))^2 d\phi$ as the objective function. Here, the subscripts *US* and *DS* denote the upstroke and downstroke respectively. The results showed that the angular velocity of the wings turned out symmetric for downstroke but is not symmetric for the upstroke motion when subjected to a constant motor angular velocity ($\dot{\beta} = \text{const.}$). However, $\dot{\beta}$ can be controlled by using equation 2 to form the wing angular velocity ($\dot{\phi}_i$) profile into the desired shape.

To confirm the beginning and end of the wings' upstrokes and downstrokes, we set up two light contact sensing plates at the two ends of the flapping angle range (figure 4-ii & iv). These plates send closed circuit signals once the pin on the wing-connected gear touches them. This mechanism allows the controller to know when the wing reaches the flapping range.

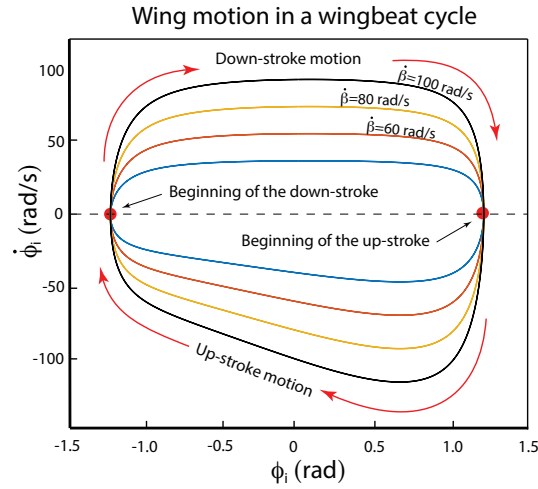


Fig. 3. Wing angular velocity vs wing angle ($\dot{\phi} - \phi$) during a wingbeat cycle for different motor driven gear velocity ($\dot{\beta}$).

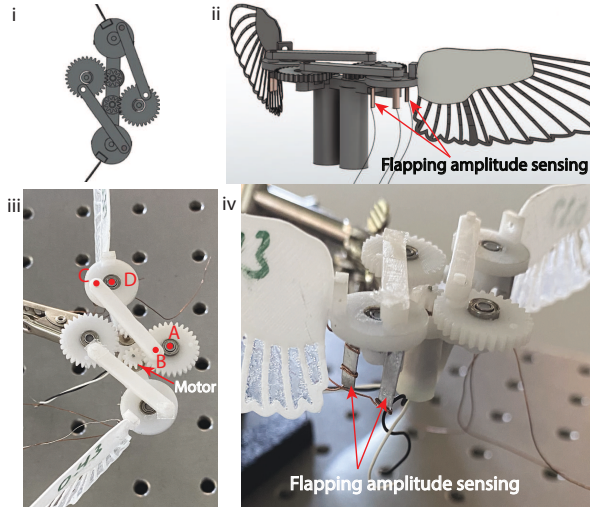


Fig. 4. The CAD model of the four-bar linkage (i) and a perspective view of the model showing the contact plates at the two ends of the flapping angle for flapping amplitude sensing (ii), the fabricated four-bar linkage mechanism (iii), and a perspective of the fabricated robot showing the flapping amplitude sensing mechanism (iv).

3 Modeling

We extend the two-dimensional modeling framework developed by Vejdani *et al.* [8, 9] to fully three-dimensional hovering. The 3D model consists of three connected rigid bodies representing the body and two wings. The body has three translational and three rotational degrees of freedom ($\mathbf{q}_b = [x, y, z, \phi, \theta, \psi]^T$) in a global coordinate system $\mathbf{G}$ (figure 5). The wings' motion are described with respect to the body in a body-fixed coordinate system $\mathbf{B}$. The left and right wings can flap independently, parameterized by angles ϕ_L and ϕ_R . Note that the wings' rotation angle, θ_L and θ_R , are assumed to be passive with constant values during upstroke and downstroke independently ($\|\theta_L\| = \|\theta_R\| = 45^\circ$) [2, 10, 11].

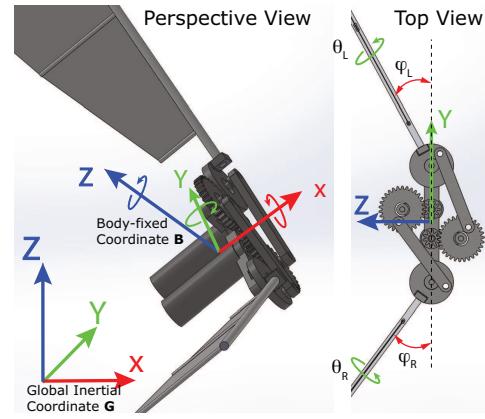


Fig. 5. The global and body-fixed coordinate systems and illustration of the degrees of freedom. The body has three translational (x, y and z) and three rotational (yaw ψ - roll ϕ - pitch θ) degrees of freedom. Each wing has 1 DoF as the flapping angle (ϕ_L and ϕ_R) and a rotation angle (θ_L and θ_R) along the wing.

Incorporating these eight DoFs into a Lagrangian formulation, we construct the equations of motion in the following form:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{f}_{aero}, \quad (3)$$

where $\mathbf{D}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ are the mass matrix, centrifugal matrix and gravitational vector respectively. The motor torques that actuate the flapping motion of the wings are in $\boldsymbol{\tau}$, while $\mathbf{f}_{aero}$ represents the external aerodynamic forces determined based on the movements of the body and the relative motion of the wings with respect to the body [8].

3.1 Aerodynamic Modeling

We use the quasi-steady assumption and the blade element theory to model the aerodynamic effects [1, 11, 12]. Specifically, the left and right wings are divided into $N = 10$ equal-width segments, with each segment approximating that of a 2D airfoil to generate a local lift and drag force vector on each element. These lift and drag (ΔL and ΔD) are located on the quarter-chord position of each individual segment and calculated as: $\Delta L = 1/2\rho C_l U_G^2 c \Delta r$ and $\Delta D = 1/2\rho C_d U_G^2 c \Delta r$ where $\rho, c, \Delta r$ and U_G are the air density, chord length, width, and airspeed of each segment in the global coordinate system $\mathbf{G}$. C_l and C_d are the lift and drag coefficients assumed to be $C_l = 1.75 \sin(2\alpha)$ and $C_d = 1.75(1 - \cos(2\alpha))$ [13]. Here, α is the effective angle of attack seen by each wing segment and, besides geometric angle of attack (that only depends on the body pitch angle and wing rotation at each instant), varies along the wingspan due to the different wing segment velocities.

4 Analysis

A hovering limit cycle is a periodic solution of the equations of motion (Eq. 3) in which the states repeat themselves

after one wingbeat period ($[\mathbf{q}, \dot{\mathbf{q}}]_t = [\mathbf{q}, \dot{\mathbf{q}}]_{t+T}$) where t is an arbitrary time instant and T is the wingbeat period [9]. To find the limit cycles, we use the multiple shooting method [14]. Therefore, the wingbeat period is divided into multiple time intervals, and to ensure the continuity of the response at the time intervals, we construct an optimization problem with the hovering and continuity conditions as the constraints. The objective function in this optimization determines which characteristic is desired for the limit cycle (e.g. minimum power, minimum wingbeat frequency, etc).

After finding the limit cycle, we determine its stability by using the concept of a Poincare map. This method converts a limit cycle into an equilibrium point on a surface called Poincare section [15]. Here, we choose the Poincare section at the beginning of the downstroke where we have the contact plates to directly sense the presence of the wing. The intersection of this surface and the limit cycle represents the states of the system at the beginning of the downstroke. Finally, the map that gives the next states, $\mathbf{x}_{k+1}$, on the Poincare section from the current state, $\mathbf{x}_k$, and the control parameters, $\mathbf{u}_k$, is called Poincare map:

$$\mathbf{x}_{k+1} = \mathcal{P}(\mathbf{x}_k, \mathbf{u}_k), \quad (4)$$

where $\mathcal{P}(\cdot)$ is the Poincare map, and $\mathbf{u}_k$ carries the control parameters at the k -th intersection, and will be introduced in next section. The equilibrium point of this map, $\mathbf{x}^*$, satisfies $\mathbf{x}^* = \mathcal{P}(\mathbf{x}^*, \mathbf{u}^*)$ where $\mathbf{u}^*$ carries the control parameters that along with $\mathbf{x}^*$ result in the limit cycle.

The dynamical properties of the system are assumed as in Table 1:

Table 1. Dynamical properties

Parameter	Description	Value	Unit
m_b	body mass	8.7×10^{-3}	kg
I_x, I_y, I_z	[roll,pitch,yaw] inertia	$[20, 5, 20] \times 10^{-6}$	kg-m ²
s	wing length	7.5	cm
c	wing chord	3.0	cm
ϕ_0	flapping amp.	70	degree
ρ	air density	1.2	kg/m ³

5 Control

In this section we explain our choice of control parameters and present the optimization framework to stabilize the system.

5.1 Wing Parameterization

To be able to recover from any type of perturbation, the wing velocity profile is parameterized such that recovery

forces and torques can be generated in any direction given the constant flapping angle amplitude constraint imposed by the four-bar mechanism. Figure 6 shows the forces and torques that can be generated by two wings moving symmetrically and asymmetrically. In this figure, the curved arrows represent the wing velocities and straight arrows show the generated aerodynamic forces with the thicker lines representing higher values than the thinner ones. Figure 6-i shows the movement and forces of the limit cycle (no perturbation). When the two wings move symmetrically, it can generate recovering forces needed for perturbations in x -, z - and θ (pitch angle) directions shown in Fig. 6-iv and 6-vi. The drag during downstroke and upstroke can regulate the x -motion and an increase to wingbeat frequency can control the z -motion. Recovering aerodynamic forces for other degrees of freedom (lateral motion y , roll ϕ and yaw ψ angles) can be generated by the asymmetric motions of the two wings.

To parameterize the wing kinematics, we use four time intervals for each wing to capture the possible symmetric and asymmetric motions of the two wings (figure 7). Additionally, we define the wing velocity profile to be continuous (C^0) and vary linearly between these time intervals. This assumption allows the wing flapping angle to be continuous and differentiable (C^1). The flapping cycle starts with the beginning of downstroke, and we divide the flapping cycle into four variable-time segments, t_{ij} , where $j = 1, 2, 3, 4$, shows the segment number and $i = L, R$ denotes the left or right wing. Thus, t_{L1} refers to the left wing's first time segment (which falls during the *downstroke*), and t_{R3} is the right wing's third time segment (which falls during the *upstroke*). The instantaneous flapping angle for each wing is denoted by ϕ_L or ϕ_R (ϕ_i for abbreviation), with its velocity indicated by $\dot{\phi}_i$.

The wing kinematic constraints enforce the magnitude of the flapping angle amplitude (constant ϕ_0) and synchronous timing of the beginning and end of the wings' downstroke and upstroke. First, we require that the start time for the downstroke and upstroke of both wings is synchronized, and since $t_{R1} + t_{R2}$ and $t_{L1} + t_{L2}$ constitute the downstroke duration while $t_{R3} + t_{R4}$ and $t_{L3} + t_{L4}$ represent the upstroke duration for the two wings (Fig. 7) therefore:

$$t_{R1} + t_{R2} = t_{L1} + t_{L2} \quad \& \quad t_{R3} + t_{R4} = t_{L3} + t_{L4}$$

Then, the magnitude of the flapping angle amplitude for each stroke (the area under the curve in figure 7 B) is enforced by:

$$2\phi_0 = 0.5(t_{i1} + t_{i2})\dot{\phi}_{i0} \quad \& \quad 2\phi_0 = 0.5(t_{i3} + t_{i4})\dot{\phi}_{i1}$$

These two equations determine the wings' velocity profile based on their time segments. Finally, the wing should come back to its original state after exactly one cycle, therefore: $(t_{i1} + t_{i2})\dot{\phi}_{i0} + (t_{i3} + t_{i4})\dot{\phi}_{i1} = 0$. Note that the period of the flapping motion ($T = t_{i1} + t_{i2} + t_{i3} + t_{i4}$) may change when controller is active.

Since ϕ_0 (half of the wing flapping angle range) is a constant imposed by the four-bar mechanism, it is easy to

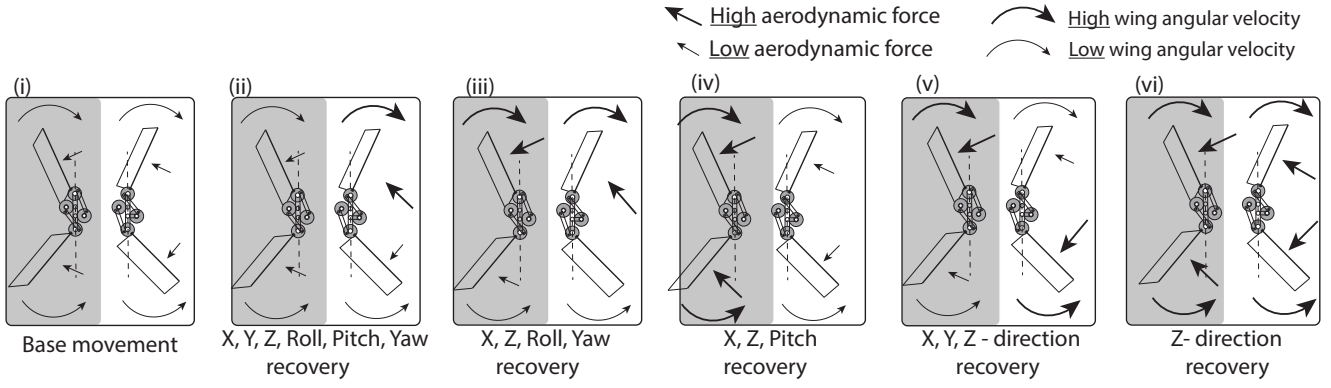


Fig. 6. Wing flapping motion scenarios during downstroke. Each box illustrates one downstroke kinematic scenario with the shade side of each box showing the first half of downstroke and plain side representing the second half of downstroke. The curved arrows show the wing angular velocity, and the straight arrows show the resultant aerodynamic forces. The thicker arrows (curved or straight) represent higher values than the thinner ones. The balancing forces and torques for the degrees of freedom are presented above each box.

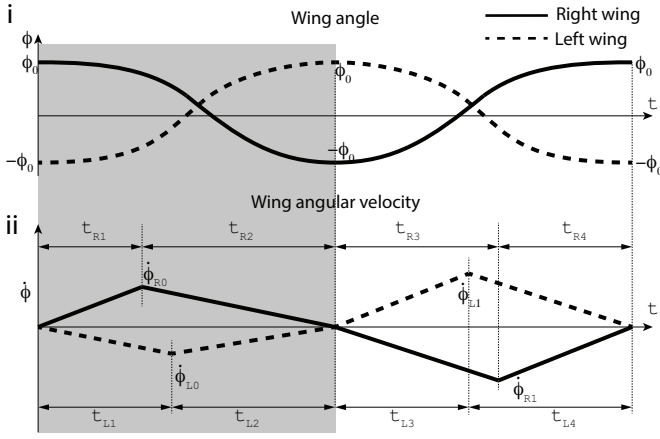


Fig. 7. Wing kinematic parameterization. Wing flapping angles (i) and velocities (ii) during a wingbeat cycle. Four variable-time intervals divide each wing's cycle. The flapping angle range is retained constant (as dictated by the four-bar mechanism) for upstroke and downstroke (gray shaded region) ($\phi_0 = 70^\circ$).

show that the whole wing kinematic can be uniquely defined by $\mathbf{u} = \{t_{R1}, t_{R2}, t_{R3}, t_{R4}, t_{L1}, t_{L3}\}^T$ where t_{L2}, t_{L4} can be calculated as a result of the above constraints. We use these parameters as control inputs to stabilize the system. Intuitively, by manipulating the duration for each time segment, the air-speed experienced by wing segment changes, and hence the instantaneous forces will be different. As seen in figure 6(iv), if a slightly larger thrust or lift is desired, the controller would choose to elongate the time duration for both wings towards downstroke, when the orientation of the segments is in favor of positive lift and thrust. In another situation, when a nose-down perturbation is applied, a recovering moment is achieved by properly arranging bias inside each downstroke for both wings t_{i1} and t_{i2} .

5.2 Optimization Framework

Now that we have parameterized the wing kinematics and defined the control input vector as $\mathbf{u} =$

$\{t_{R1}, t_{R2}, t_{R3}, t_{R4}, t_{L1}, t_{L3}\}^T$, we can design a feedback law to stabilize the system while receiving the desired performance for recovery. We aim the fast recovery as our desired performance and therefore, we use the greatest eigenvalue of the closed-loop Monodromy matrix as the objective function ($J = \max(\|\lambda_i\|)$) to be minimized to recover the system as fast as possible. For this purpose, we construct an optimization problem with this objective function and use the discrete algebraic Riccati equation [16] as a constraint inside the optimization to guarantee the stability (equation 5). The outcome of this optimization problem is the gain matrix $\mathbf{K}$ that maps the states of the system ($\mathbf{x}_k^T$) at the beginning of each downstroke to the required time intervals ($\{t_{R1}, t_{R2}, t_{R3}, t_{R4}, t_{L1}, t_{L3}\}^T$) during that wingbeat. Then, these required time intervals along with the kinematic constraints presented in the previous section (Sec. 5.1) uniquely determine the required wing velocity profile to stabilize the system.

$$\text{Minimize } J = \max(\|\lambda_i\|) \\ \{q_1 \dots q_{12}\}, \{r_1 \dots r_6\}$$

subject to:

$$\left| \frac{\partial \mathcal{P}}{\partial \mathbf{x}} - \frac{\partial \mathcal{P}}{\partial \mathbf{u}} \mathbf{K} - \lambda_i \mathbf{I} \right| = 0,$$

$$\mathbf{K} = -(\text{diag}(\{r_1 \dots r_6\} + [\frac{\partial \mathcal{P}}{\partial \mathbf{u}}]^T [\mathbf{S}] [\frac{\partial \mathcal{P}}{\partial \mathbf{u}}])^{-1} [\frac{\partial \mathcal{P}}{\partial \mathbf{u}}]^T [\mathbf{S}] [\frac{\partial \mathcal{P}}{\partial \mathbf{x}}]),$$

$$\mathbf{S} = \text{Ric}(\frac{\partial \mathcal{P}}{\partial \mathbf{x}}, \frac{\partial \mathcal{P}}{\partial \mathbf{u}}, \text{diag}(\{q_1 \dots q_{12}\}^T), \text{diag}(\{r_1 \dots r_6\}^T)),$$

(5)

Where $|\cdot|$ is the determinant operator, $\mathcal{P}$ is the Poincare map from equation 4 and the operator $\text{Ric}(\cdot)$ returns a symmetric matrix that satisfies the discrete algebraic Riccati equation [16].

6 Results and Discussion

In this section we present the simulation results to stabilize the 3D system in table 1. Figure 8 shows the normalized control input vector (time intervals) for an initial nose-down pitch perturbation. To recover from this initial perturbation, the time intervals u_k start to change at the beginning of the first wingbeat and need to update only once per cycle. This sensing update rate is aligned with observations from natural fliers' like fruit flies [17] and releasing the need for continuous monitoring of the states in robotic systems [18]. This update instant is detected by the flapping amplitude sensing plates. All the time intervals approach their nominal values for the unperturbed periodic cycle after about ten wingbeats which is comparable to [19] but here with less actuation need. The total wingbeat period, which is the summation of the time intervals for each wing, are shown as a dashed black line. The maximum change of the wingbeat period with respect to the reference value (T_{ref}) is about 4%.

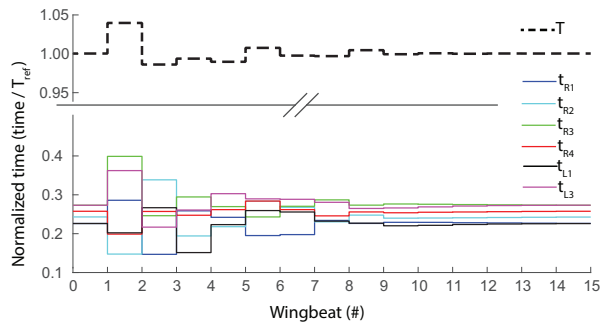


Fig. 8. The controlled time intervals during recovery. At the beginning of each wingbeat cycle, the time intervals are updated and remain constant during the wingbeat. The vertical axis is normalized based on the wingbeat period of the limit cycle (T_{ref}) and the dashed line (T) shows the current wingbeat period.

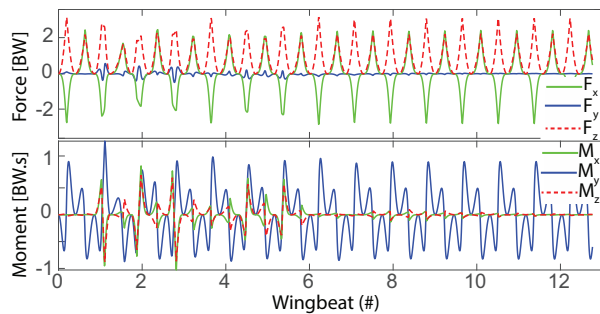


Fig. 9. The normalized aerodynamic forces and torques to recover the 3D hovering model. The forces are normalized by the body weight (BW) and the aerodynamic moments are normalized by ($BW \cdot s$), where s is the wing length.

Next, the generated aerodynamic forces by the proposed controller are shown in Fig. 9. The forces and torques are

normalized with respect to the body weight (BW) and wing length (s). Due to the coupling nature of the degrees of freedom, all of the in- and out-of-plane forces and torques are used to recover the system. This coupling behavior is due to the minimal actuation in this system and unlike [20], that showed low level of cross coupling between pitch and roll moments, the controller engages the torques in other directions for recovery.

7 Conclusions

In this paper, we presented the kinematic design of a four-bar linkage mechanism for a 3D hovering flapping wing robot. This mechanism allows the motors to move continuously in one direction while generating flapping motion on the wings. However, this mechanism restrains the wing flapping amplitude and therefore applies higher level of underactuation to the system. To stabilize such a system, we proposed a new wing kinematic parameterization. The results have shown that the proposed wing parameterization provides adequate control authority to stabilize the 3D hovering system. Additionally, the results showed that an update rate of once per wingbeat is enough to stabilize the system, rather than monitoring the system continuously. These findings expand our understanding of the control authority that a flapping wing system possesses, and offer a way to articulate and control flapping wing robots that can lead to lighter, more cost- and energy efficient vehicles.

acknowledgment

This work is supported by the National Science Foundation grant 1931122.

References

- [1] Alireza Ramezani, Soon-Jo Chung, and Seth Hutchinson. A biomimetic robotic platform to study flight specializations of bats. *Science Robotics*, 2(3):eaal2505, 2017.
- [2] Zhan Tu, Fan Fei, Jian Zhang, and Xinyan Deng. An At-Scale Tailless Flapping-Wing Hummingbird and Experimental Validation. 36(5):1–15, 2020.
- [3] Robert J. Wood. The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Transactions on Robotics*, 24(2):341–347, 2008.
- [4] Guangping He, Tingting Su, Taoming Jia, Lei Zhao, and Quanliang Zhao. Dynamics Analysis and Control of a Bird Scale Underactuated Flapping-Wing Vehicle. *IEEE Transactions on Control Systems Technology*, 28(4):1233–1242, 2020.
- [5] David B. Doman, Michael W. Oppenheimer, and David O. Sigthorsson. Wingbeat shape modulation for flapping-wing micro-air-vehicle control during hover. *Journal of Guidance, Control, and Dynamics*, 33(3):724–739, 2010.
- [6] Michael W. Oppenheimer, David B. Doman, and David O. Sigthorsson. Dynamics and control of a

- biomimetic vehicle using biased wingbeat forcing functions. *Journal of Guidance, Control, and Dynamics*, 34(1):204–217, 2011.
- [7] Sjoerd Tijmons, Matj Karásek, and Guido De Croon. Attitude control system for a lightweight flapping wing MAV. *Bioinspiration and Biomimetics*, 13(5), 2018.
- [8] Hamid Vejdani, David Boerma, Sharon M Swartz, and Kenneth S Breuer. Guidelines for the Design and Control of Bio-inspired Hovering Robots. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4160–4166, 2017.
- [9] Hamid R. Vejdani, David B. Boerma, Sharon M. Swartz, and Kenneth S. Breuer. The dynamics of hovering flight in hummingbirds, insects and bats with implications for aerial robotics. *Bioinspiration and Biomimetics*, 14(1), 2019.
- [10] Jian Zhang, Fan Fei, Zhan Tu, and Xinyan Deng. Design optimization and system integration of robotic hummingbird. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5422–5428, 2017.
- [11] Jian Zhang, Zhan Tu, Fan Fei, and Xinyan Deng. Geometric flight control of a hovering robotic hummingbird. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5415–5421, 2017.
- [12] Ben Parslew. Predicting power-optimal kinematics of avian wings. *Journal of the Royal Society Interface*, 12(102), 2015.
- [13] Mostafa R. A. Nabawy and William J. Crowther. On the quasi-steady aerodynamics of normal hovering flight part i: the induced power factor. *Journal of The Royal Society Interface*, 11(93):20131196, 2014.
- [14] J Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, second edition, 2010.
- [15] Ali H. Nayfeh and Balakumar Balachandran. Applied Nonlinear Dynamics. *Applied Nonlinear Dynamics*, 1995.
- [16] K. Ogata. *Discrete-time Control Systems*. Prentice Hall, 1995.
- [17] Song Chang and Z Jane Wang. Predicting fruit flies sensing rate with insect flight simulations. *Proceedings of the National Academy of Sciences*, 111(31):11246–11251, 2014.
- [18] Alireza Ramezani, Xichen Shi, Soon Jo Chung, and Seth Hutchinson. Lagrangian modeling and flight control of articulated-winged bat robot. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:2867–2874, 2015.
- [19] Hamid R Vejdani. Dynamics and stability of bat-scale flapping wing hovering robot. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 1106–1111. IEEE, 2019.
- [20] Matej Karasek, Alexandre Hua, Yanghai Nan, Mohamed Lalami, and André Preumont. Pitch and roll control mechanism for a hovering flapping wing mav. *International Journal of Micro Air Vehicles*, 6(4):253–