Robustness analysis of minimally-actuated flapping wing systems due to aerodynamic modeling uncertainty

Hamid Vejdani[1], Xiaozhou Fan[2] and Kenneth Breuer[3]

Abstract—In this paper, we study the robustness of minimally-actuated 3D flapping wing hovering systems under aerodynamic modeling uncertainties. Our goal is to evaluate and compare the vehicles' robustness under commonly used control scheme and provide guidelines for control design stage. For this purpose, we first develop a 3D flapping wing dynamical model and find a desired hovering flight mode based on the nominal estimate of the aerodynamic characteristics. Then, we parameterize the wing kinematics in a novel way to allow the system to generate recovering aerodynamic forces. Using this parameterization, the control inputs are defined as the variable-time segments of the wingbeat cycle. We use the discrete LQR framework to design three controllers, each focused on a different objective: i) least control input change, ii) fast convergence rate and iii) least state residue. To study the robustness, we apply the controllers to the nonlinear system with aerodynamic uncertainty and through exhaustive simulations, we find the range of uncertainties that lead to stable response. The results show that controller (i) has the largest tolerance for drag and lift coefficient uncertainties. Moreover, this controller is robust on the uncertainties from lift offset phase over a wide range (up to 90 degrees) but has narrower tolerance to drag offset phase changes, especially towards the positive offset angles. Considering inherent uncertainty associated with the quasi-steady models, the results suggest to use a slightly larger than nominal value (upper bound) aerodynamic coefficients for control design to enjoy a greater tolerance (more robustness). This is true for both the magnitudes as well as the offset angles used in lift and drag functions.

I. INTRODUCTION

Flapping wing micro aerial vehicles (MAVs) have received a significant interest due to the outstanding performance of their biological counterparts - insects, bats and birds [1], [2], [3], [4]. However, the aerodynamics of flapping wings at moderate Reynolds numbers ($10k < Re < 100k$), in which bird and bat-scale robots operate, has not been accurately formulated for practical control purposes [5], [6], [7]. Therefore, despite its limitations, the quasi-steady assumption using a blade-element technique has been accepted as a practical tool for aerodynamic modeling on flapping wings [3], [8], [9], [10]. In this paper, we use this aerodynamic modeling approach and present a method to stabilize a 3D flapping wing model of a hovering robot with constant flapping angle range (figure 1). We then explore the effect of the aerodynamic uncertainties on the stability of the system.

[1]Mechanical, Robotics, and Industrial Engineering Department, Lawrence Technological University, Southfield, MI, USA. e-mail: hvejdanin@ltu.edu

[2]Center for Fluid Mechanics, School of Engineering, Brown University, Providence, RI, USA. e-mail: xiaozhou_fan@brown.edu

[3]Center for Fluid Mechanics, School of Engineering, Brown University, Providence, RI, USA. e-mail: kbreuer@brown.edu
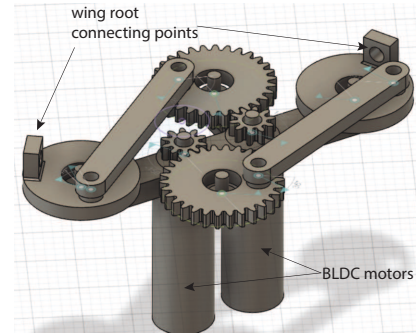
Fig. 1. The CAD model of a designed robot with two motors and constant flapping angle range. The two four-bar mechanisms only allow the wings to move within a constant flapping range.

To study the stability of flapping wing systems, linearizing the dynamics of the full model and evaluating the stability of the resultant linear system is a common approache [11], [12], [13], [14]. Likewise, one can design a stabilizing controller for the linearized model in order to stabilize the original nonlinear system [3], [8], [15], [16], [17]. Deng *et al.* [8] used the averaging theory and discrete LQR to stabilize an insect-scale hovering system. They used the wings' flapping and rotation angles as control inputs and demonstrated stable hovering in simulation. Although they did include the sensor and actuator models, they did not study how uncertainties in aerodynamic modeling might affect the stability of the system. Ramezani *et al.* [3] used a linearized model of their bat-scale flapping wing robot and designed an exponentially stable forward flight. They used blade element theory [18] with the lift and drag located at the quarter chord point of each element to model the aerodynamic effects. Their successful flight experiments showed the usefulness of the quasi-steady modeling of aerodynamic forces for control design. However, they did not discuss the robustness of the controller.

Doman *et al.* [19] proposed using variable velocity profiles for the wing strokes in order to control a 3D hovering system. Since they kept the flapping angle range constant, they used a bob-weight to control the pitching moment. Later, Oppenheimer *et al.* [20] introduced wing stroke bias into the variable velocity profiles and demonstrated full control over a 3D flapping wing in hover. They used blade-element theory and cycle averaging to model the aerodynamic effects, and compared their results to a model using instantaneous aerodynamic forces and moments. They did not discuss the validity of their results for lower flapping frequencies in which cycle averaging technique is less viable [21]. Zhang *et al.* [22]

used quasi-steady modeling for aerodynamics and designed a nonlinear geometric controller to stabilize a hummingbird-scale hovering robot. However, they did not show the robustness of the system due to aerodynamic uncertainty.

Serrani [23] used a different approach and designed a nonlinear robust controller for a 2D flapping wing MAV model - an extension of a single-DOF flapping wing MAV [24]. For the 2D model, the author assumed two control inputs: wingbeat frequency, and stroke plane angle, and used the quasi-steady assumption for aerodynamic modeling. The proposed controller stabilized the model with 15% parameter uncertainty including aerodynamic coefficients. Although the model was under-actuated for a 2D system, controlling it in 3D seems to require additional motors. Moreover, due to the use of the cycle averaging technique in designing the controller, and its robustness for systems with lower flapping frequency is not known. Alkitbi et al. [25] used the same model and presented a robust controller using only the flapping motion of the wings for actuation. They, too, used cycle averaging in designing the controller and deomonstrated robustness in stabilizing the system.

In this paper, we present results on the effect of aerodynamic uncertainty on the stability of a 3D hovering model. Like previous researchers, we employ the quasi-steady assumption for the aerodynamic modeling, and we explore the performance of three stabilizing controllers - all obtained by linearizing the nonlinear system - and quantify the stability of the system as the different parameters in aerodynamic modeling change. The results will demonstrate that the controller designed for control effort efficiency (least control input change) has the largest tolerance for aerodynamic uncertainty. Based on these results, we present recommendations for choosing aerodynamic parameters during the analysis and control design in order to increase the robustness of the system.

II. METHODOLOGY

In this section we first present our dynamic and aerodynamic modeling and describe our analysis method. After that, we explain the control strategy framework and robustness analysis on the system.

A. Dynamic Modeling

We use a Lagrangian framework to model a 3D flying system with two rectangular wings and a rigid body. The left and right wings have independent flapping (ϕ_{lw} and ϕ_{rw}) and rotational (θ_{lw} and θ_{rw}) motions relative to the body (top view in figure 2). We assume the wing flapping angles (ϕ_{lw} and ϕ_{rw}) are controllable but the wing rotational angles (θ_{lw} and θ_{rw}) are passive with $\pm 45°$ angle during upstroke and downstroke [26], [22], [4]. The body has 6 DoFs - three translational (x, y and z as forward, lateral toward left wing and vertical upward respectively) and three rotational as yaw (ψ), roll (ϕ) and pitch (θ) angles. In total the model has 8 degrees of freedom $\mathbf{q} = [\mathbf{q}_b, \theta_{lw}, \theta_{rw}]^T$. The equations of motion can be written as:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau + \mathbf{f}_{aero}, \quad (1)$$
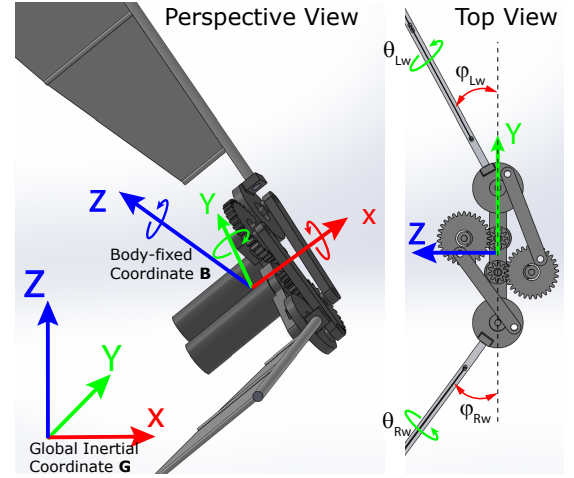

Fig. 2. Schematics of the global and body-fixed coordinate and illustration of degrees of freedom. The body has 6 degrees of freedom (DoFs) in global coordinate **G**: three translational DoF (x,y and z) and three rotational DoF (Euler rotation sequence: yaw ψ - roll ϕ - pitch θ). Each of the wing has 1 DoF as the flapping angle (ϕ_{lw} and ϕ_{rw} for left and right wing, respectively) in **B**. The wing rotation angles (wing pronation) for both left and right wing (θ_{lw} and θ_{rw}) are assumed to be a constant value in each stroke and only reverse sign in stroke reversal ($\|\theta_{lw}\| = \|\theta_{rw}\| = 45°$).

where $\mathbf{D}, \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ are the mass matrix, centrifugal matrix and gravitational vector respectively. The generalized torque, $\tau = [0_{1\times 6}, \tau_{lw,int}, \tau_{rw,int}]^T$ denotes the internal forces between the left and right wings and the body, while $\mathbf{f}_{aero}$ represents the external aerodynamic forces and the ensuing torques felt by both left and right wings [27].

B. Aerodynamic Modeling and Uncertainty

We use the quasi-steady assumption to model the aerodynamic effects [3], [28], [29], [22]. Specifically, the left and right wings are divided into $N = 10$ equal width segments, with each segment approximating that of a 2D airfoil in quasi-steady state to generate a local lift and drag force vector on each element (ΔL and ΔD in figure 3). These lift and drag are located at the quarter-chord position of each individual segment and calculated as:

$$\Delta L = 1/2\rho C_l U_G^2 c\Delta r \quad (2)$$
$$\Delta D = 1/2\rho C_d U_G^2 c\Delta r, \quad (3)$$

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 **G** and C_l and C_d are the lift and drag coefficients assumed in the following format [5], [8], [9], [10], [28]:

$$C_l = C_{l0}\sin(2\alpha + \eta_L) \quad (4)$$
$$C_d = C_{d0}(1 - \cos(2\alpha + \eta_D)). \quad (5)$$

Here α is the effective angle of attack seen by each wing segment and varies along the wingspan due to the different wing segment velocities. It is worth noting that the geometric angle of attack depends on the body pitch angle and wing spanwise rotation at each instant of time. The lift and drag coefficient amplitudes (C_{l0} and C_{d0}) and the corresponding offset phases
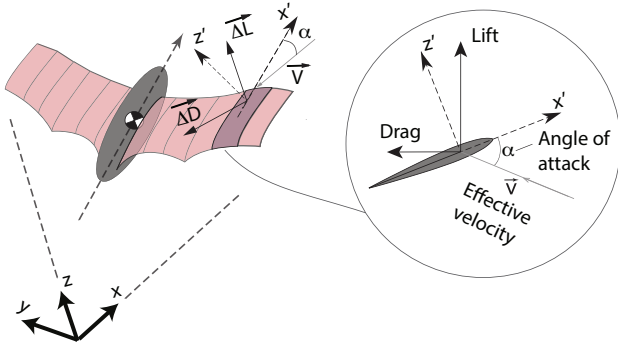
Fig. 3. The schematic of the blade element method on a general wing shape in 3D. The lift and drag are computed at each instant of time and placed on the quarter chord of each segment.
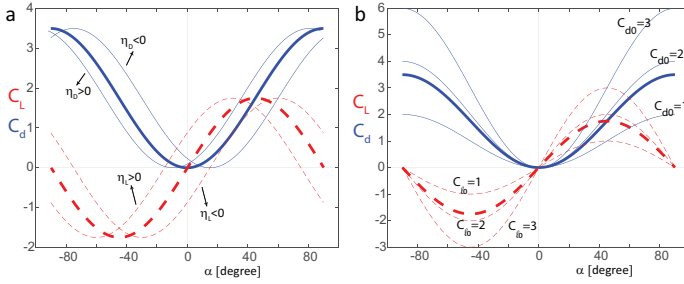

Fig. 4. The lift (dashed red line) and drag (solid blue line) coefficients change with offset phases (a) and amplitudes (b). The bold lines show the base case for the coefficients (nominal values).

(η_L and η_D, physically related to the camber of the airfoil) determine the uncertainties of the aerodynamic modeling. For finding the hovering limit cycle and designing the stabilizing controller, we assume the amplitudes to be $C_{l0} = C_{d0} = 1.75$, and the offset phases to be $\eta_L = \eta_D = 0$ [2], [8], [10], [30]. However, for robustness analysis we allow the amplitudes and offsets to change from 1 to 3 and from $-90°$ to $+90°$ respectively without altering the designed controller. Figure 4 shows the effect of the amplitudes (figure 4-a) and offset phases (figure 4-b) on the lift and drag coefficients.

C. Analysis

We start by parameterizing the wing kinematics and describing how to find hovering limit cycles. After presenting the stability analysis, we explain the controller design. Finally, we present the robustness analysis on the controlled system under aerodynamic uncertainties.

1) Wing Kinematics: We split the wingbeat cycle of each wing into four piecewise-constant acceleration intervals (two for each half stroke) to capture the required symmetric and asymmetric motions of the two wings (figure 5). Assuming the flapping cycle starts with the beginning of downstroke, each wing flapping period is split into four time segments, t_{ij}, where $j = 1, 2, 3, 4$, and $i = L, R$ denotes the left or right wing. Thus, t_{R1} refers to the first time interval of the right wing *downstroke*, and t_{L3} is the first part of the left wing *upstroke*. The wing flapping angle is denoted by ϕ_{ij}

($j = 1, 2, 3, 4$, and $i = L, R$); $\dot{\phi}_{ij}$ and $\ddot{\phi}_{ij}$ are the velocity and acceleration respectively. For each wing, there are 2 constraint equations to enforce - zero velocity at the beginning and end of the downstroke:

$$\ddot{\phi}_{i1}t_{i1} + \ddot{\phi}_{i2}t_{i2} = 0 \tag{6}$$
$$\ddot{\phi}_{i3}t_{i3} + \ddot{\phi}_{i4}t_{i4} = 0 \tag{7}$$

To apply the range of the flapping angle amplitude during each stroke, the following two constraints apply:

$$2\phi_0 = 0.5(t_{i1} + t_{i2})\dot{\phi}_{i0} \tag{8}$$
$$2\phi_0 = 0.5(t_{i3} + t_{i4})\dot{\phi}_{i1} \tag{9}$$

Moreover, we require that the start of the downstroke and upstroke of both wings be synchronized:

$$t_{R1} + t_{R2} = t_{L1} + t_{L2} \tag{10}$$
$$t_{R3} + t_{R4} = t_{L3} + t_{L4} \tag{11}$$

Finally, the wing should come back to its original state after exactly one cycle:

$$(t_{i1} + t_{i2})\dot{\phi}_{i0} + (t_{i3} + t_{i4})\dot{\phi}_{i1} = 0. \tag{12}$$

Since the flapping amplitude, ϕ_0, is a constant, and determined by the wing four-bar linkage design (figure 1), it is easy to show that all $\ddot{\phi}_{ij}$ as well as t_{L2} and t_{L4} can be uniquely defined by $\mathbf{u} = \{t_{R1}, t_{R2}, t_{R3}, t_{R4}, t_{L1}, t_{L3}\}^T$ which we use as the control input vector for our simulation. Intuitively, by manipulating the duration of each time segment, the airspeed experienced by wing segment changes, and hence the instantaneous forces will be different. For example, 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 or upstroke for both wings.

The simulation parameters for the 3D model are given in Table I:

TABLE I
NOMINAL PROPERTIES OF A 3D HOVERING MODEL

Parameter	Description	Value	Unit
m_b	body mass	4.68×10^{-3}	kg
I_x, I_y, I_z	[roll, pitch, yaw] rotational inertia	$[2.5, 0.5, 2.5] \times 10^{-6}$	kg-m^2
s	wing length	52.5	mm
c	wing chord	13	mm
ϕ_0	flapping angle ampl.	80	degree
ρ	air density	1.2	kg/m^3
C_{l0}, C_{d0}	lift, drag ampl.	1.75	N/A
η_L, η_D	lift, drag offset phase	0	degree

2) Hovering Limit Cycles: We consider hovering periodic orbits as a solution of the equations of motion in which the states repeat themselves after one wingbeat. Mathematically, it is formulated as:

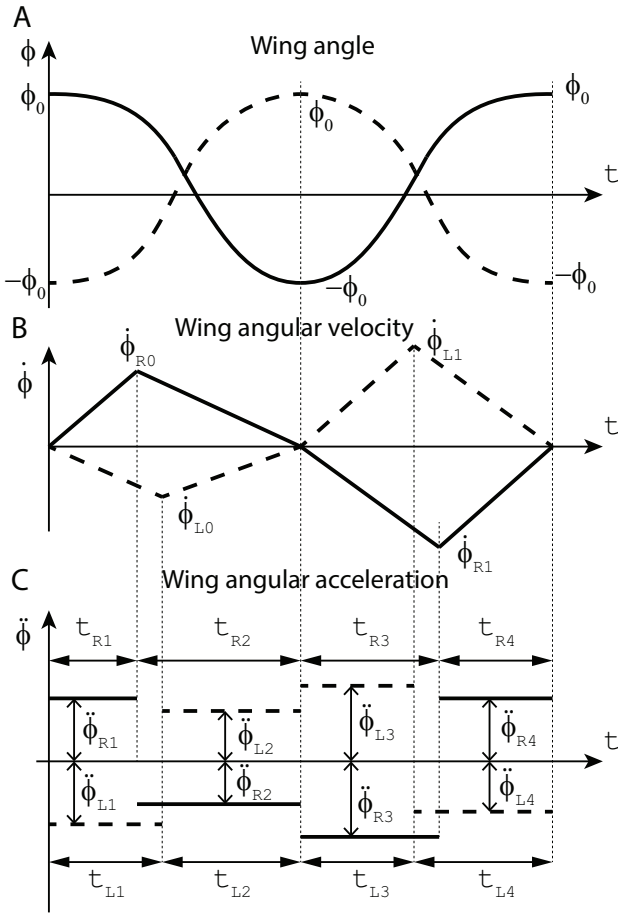$$[\mathbf{q}, \dot{\mathbf{q}}]_t = [\mathbf{q}, \dot{\mathbf{q}}]_{t+T} \tag{13}$$

Fig. 5. Wing flapping angels (A), velocities (B) and accelerations (C) for a 3D hovering system during a wingbeat. The accelerations are constant during each time interval. Each wing has 4 time intervals and 4 accelerations. The flapping angle range is assumed to be constant for upstroke and downstroke ($\phi_0 = 80°$).

where t is an arbitrary time instant and T is the wingbeat period. Due to the instability and sensitivity of these limit cycles, we use the multiple shooting method [31]. In this method, the wingbeat period is divided into multiple time intervals and the equations of motion are solved for each time interval separately. 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). This limit cycle is used as the reference trajectory about which to linearize the dynamics of the nonlinear system for controller design.

3) Stability Analysis: After finding the desired limit cycle, we determine its stability by discretizing the continuous system at the beginning of the downstroke, which we define as our Poincare section. Therefore, the hovering limit cycle is converted to an equilibrium point on this Poincare section. The stability of the discretized system is determined based on the magnitude of the eigenvalues of the Monodromy matrix

[32]. The intersection of the Poincare section and the hovering limit cycle, a.k.a the equilibrium point of the Poincare map, represents the states of the hovering system at the beginning of the downstroke. The intersection of the Poincare section and the neighboring trajectories around the limit cycle is determined by the Poincare map which gives the next state, $\mathbf{x}_{k+1}$, from the current state, $\mathbf{x}_k$, and the control values, $\mathbf{u}_k$:

$$\mathbf{x}_{k+1} = \mathcal{P}(\mathbf{x}_k, \mathbf{u}_k), \tag{14}$$

where $\mathcal{P}(\cdot)$ is the Poincare map. $\mathbf{u}_k$ carries the control values at the k-th intersection, and will be introduced in next section. The equilibrium point, $\mathbf{x}^*$, returns itself such that: $\mathbf{x}^* = \mathcal{P}(\mathbf{x}^*, \mathbf{u}^*)$, where $\mathbf{u}^*$ carries the control values that, along with $\mathbf{x}^*$, yields the desired/reference limit cycle.

The stability of this map can be determined by linearizing around its equilibrium point, $(\mathbf{x}^*, \mathbf{u}^*)$, as follows:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{15}$$

where

$$\mathbf{A} = \left.\frac{\partial \mathcal{P}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*, \mathbf{u}=\mathbf{u}^*} \quad \text{and} \quad \mathbf{B} = \left.\frac{\partial \mathcal{P}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\mathbf{x}^*, \mathbf{u}=\mathbf{u}^*} \tag{16}$$

The eigenvalues of the Jacobian matrix, $\mathbf{A}$, indicate the stability of the system. If all the eigenvalues are located inside of the unit circle, the system is stable. Otherwise we need a feedback controller (next section) to stabilize the system by updating the control input values ($\mathbf{u}_k$) at the Poincare section (beginning of the downstroke).

D. Control

The feedback controller is designed to stabilize the linear system (Eq. 15) but will apply to the original nonlinear system (Eq. 1). We assume a feedback law in the form of $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k$ to update the control inputs only at the beginning of the downstroke (once per wingbeat cycle). For this purpose, we use the LQR technique for discrete systems to determine the gain matrix, $\mathbf{K}$. This requires minimizing the objective function:

$$J = \Sigma(\mathbf{x}_k^T \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^T \mathbf{R}\mathbf{u}_k), \tag{17}$$

where the positive semi-definite matrix $\mathbf{Q}$ and positive definite matrix $\mathbf{R}$ carry weights that trade off between the control effort and regulation performance. To minimize the cost function, J, and find the stabilizing control input, $\mathbf{u}_k$, the discrete Riccati equation is solved. After that, $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k$ can be calculated at the beginning of each downstroke. We consider three scenarios for the $\mathbf{Q}$ and $\mathbf{R}$ matrices: i) a controller with focus on the least control input change, ii) a controller with highest convergence rate, and iii) a controller with focus on all the state change equally (large values for all entries in $\mathbf{Q}$ with respect to $\mathbf{R}$). To find the gain matrix, $\mathbf{K}$, for controller (i) we assign high values to the diagonal entries in $\mathbf{R}$ matrix to penalize control effort with respect to state values. For controller (ii) we use the following optimization to minimize the largest eigenvalue of the closed-loop system by tuning the diagonal entries of $\mathbf{Q}$ and $\mathbf{R}$ matrices:
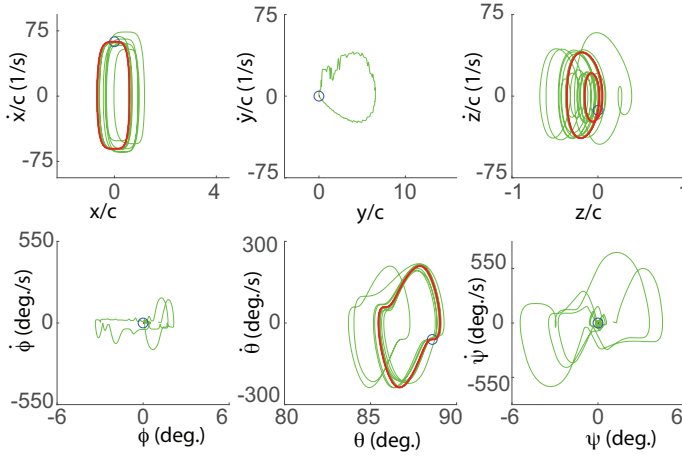
Fig. 6. The phase portrait of the controlled 3D system. The equilibrium limit cycle is shown in red, while the green lines are the trajectories during the fast recovery (controller (ii)). Due to the coupled nature of the dynamics, the lateral translation and the roll and yaw angles exit the equilibrium state during recovery but return after about 10 wingbeats.

Fig. 7. The recovery of the controllers under initial pitch perturbation. The $x-$ and $z-$movements are normalized by the wing chord. The unperturbed trajectory is shown with a dashed red line. The response using the quick-recovery controller(controller (ii)) is shown with a green line, and converges within 10 wingbeats; The minimal-change controller (i), shown with a blue line) recovers more slowly.

$$\underset{Q_{ii}, R_{ii}}{Minimize} \quad \max(\|\lambda_i\|), \qquad \text{subject to:}$$

$$|\mathbf{A} - \mathbf{BK} - \lambda_i \mathbf{I}| \quad = \quad 0;$$
$$\mathbf{K} \quad = \quad \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S};$$
$$\mathbf{S} \quad = \quad Ric(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R});$$
$$Q_{ii} \succeq 0, \qquad R_{jj} \succ 0.$$
$$i = 1 \ldots 12; \qquad j = 1 \ldots 6,$$

where $| \cdot |$ is the determinant operator and the operator $Ric(\cdot)$ returns a symmetric matrix that satisfies the Riccati equation. In this controller, the diagonal entries of $\mathbf{R}$ and $\mathbf{Q}$ matrices can be any value satisfying the constraints in the optimization program.

E. Robustness analysis

After designing the controller and finding the gain matrix, $\mathbf{K}$, we apply the aerodynamic uncertainties (as defined in section II-B) to the nonlinear model (Eq. 1). We run the simulation for at least 200 flapping cycles and evaluate the convergence of the states on the Poincare section. If this sequence converges, we consider that level of uncertainty in the tolerance domain of the controller.

III. RESULTS AND DISCUSSION

We start by showing the phase-portrait of the hovering orbit (red line) and the response of the fast-converging controller (controller (ii) - green line) in figure 6. Figure 7 compares controller (i) and (ii). Again, the equilibrium cycle is shown by a red line. In this figure, a $2.5°$ initial perturbation is applied to the equilibrium pitch angle, which in addition to being a state perturbation, also affects the angle of attack. Controller (ii), designed for quick recovery, can return the states to the reference orbit within 10 wingbeats (green line) while controller (i), designed for the least change in control
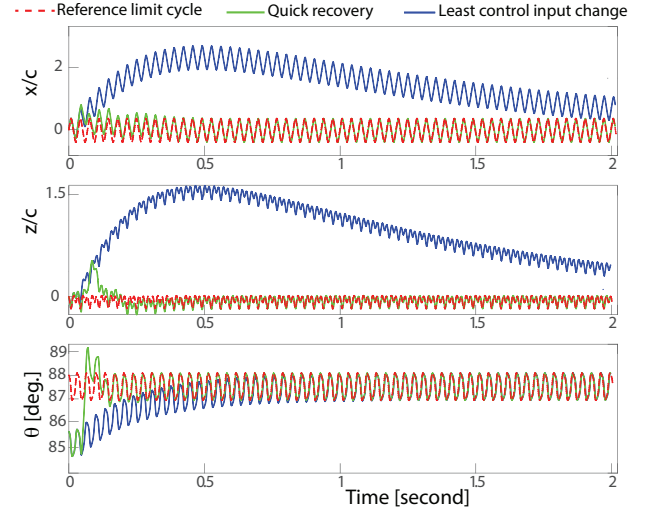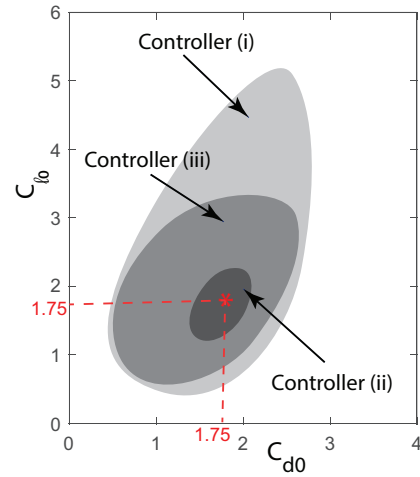

Fig. 8. The regions show the tolerance domains for the three controllers. The red star corresponds to the nominal values of lift and drag coefficient amplitudes for which the controllers designed ($C_{d0} = C_{l0} = 1.75$). Controller (i) with the slowest recovery rate has the largest tolerance domain (light gray), while the fast recovery controller (ii), shows the most sensitivity to the aerodynamic uncertainties (darkest gray).

inputs, recovers the system over a much longer time ((blue line). The convergence using controller (iii) lies in-between these two results.

Figure 8 shows the robustness of the controllers due to the C_{d0} and C_{l0} uncertainties while keeping the phase uncertainties equal to zero ($\eta_D = \eta_L = 0$). The controller designed for the least change in control inputs, controller (i), provides the largest robustness while the controller designed for quick recovery, controller (ii), can only handle small changes in C_{d0} and C_{l0} around the values for which the controllers are designed ($C_{d0} = C_{l0} = 1.75$ - red star in Fig 8).
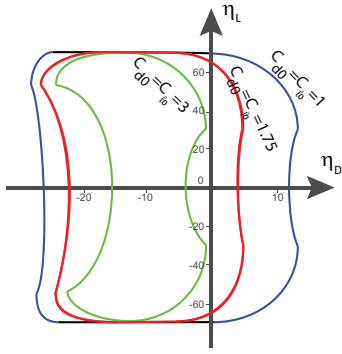
674

Fig. 9. The enclosed regions show the tolerance of the system on the lift and drag offset phase uncertainties as the true values of coefficient amplitudes vary as shown. The system gained larger tolerance for the offset phases when the true values of the coefficient amplitudes are smaller than the nominal values. The tolerance of the controller is asymmetric with respect to drag offset phase and sensitive to the positive values.
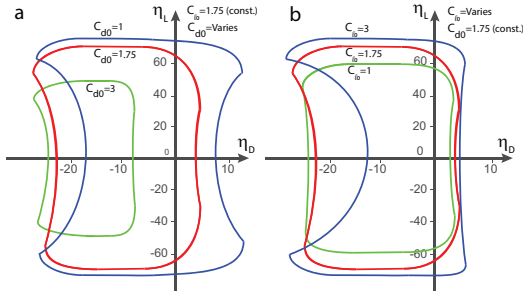

Fig. 10. The robustness of the system with respect to the offset angles (η_D and η_L) when: (a) the lift coefficient amplitude ($C_{l0} = 1.75$) is constant (low uncertainty on lift) and drag coefficient amplitude has high uncertainty, (b) the drag coefficient amplitude is constant ($C_{d0} = 1.75$) and higher uncertainty exists on lift coefficient amplitude.

Figure 9 shows the effect of uncertainties on the phase offsets (η_D and η_L) in the aerodynamics modeling while the actual C_{d0} and C_{l0} are varied together ($C_{d0} = C_{l0}$). The results show that the controller has a larger tolerance to phase offset uncertainties if the actual values of C_{d0} and C_{l0} are smaller than the values for which the controller is designed (blue line vs. green line). Thus, to increase the tolerance domain (the area enclosed), this result suggests using a slightly larger than nominal value (upper bound) for lift and drag coefficients during the controller design stage. Moreover, the domains in this figure show that the tolerance of the controller to the drag offset phase (η_D) is significantly smaller than the lift offset phase (η_L). This is even more pronounced for positive drag phase offsets. As the C_{d0} and C_{l0} increase together, the tolerance domain shrinks, especially for the drag offset uncertainty (green boundary). The asymmetry of the controller tolerance for the drag offset phase (η_D) suggests to use an initial positive drag offset phase during the controller design, and since the lift offset phase is symmetric for positive and negative uncertainties, there is no need to do the same for lift offset phase.

Figure 10 shows the tolerance domains of the controller

when one of the coefficient amplitudes (C_{d0} or C_{l0}) is subjected to higher uncertainty than the other one. When there is more uncertainty in drag coefficient amplitude (C_{d0}), figure 10-a shows that the tolerance domain shrinks toward negative drag offset phase for larger drag coefficients, but remains robust and symmetric for lift offset phase. If the actual drag coefficient amplitude is bigger than the value for which the controller is designed (like $C_{d0} = 1.75$ here), the controller may not show any tolerance for positive drag offset uncertainty. This behavior suggests using an upper bound for the drag coefficient amplitude when designing the controller. This is compatible with the previous case in which there are uncertainty in lift and drag coefficient amplitudes (Fig. 4). It suggests to consider a slightly greater phase offset for the drag during the control design. Figure 10-b shows the tolerance domain of the controller with more uncertainty in the lift coefficient amplitude. Here, the results show that the tolerance of the system remains symmetric and robust for lift offset phase and nearly constant for the *positive* drag offset phase as the lift uncertainty changes. For lift coefficient amplitude (C_{l0}) higher than the value for which the controller is designed ($C_{l0} = 1.75$ here), the tolerance expands in lift offset phase but shrinks rapidly for *negative* drag offset phase. This behavior confirms using an upper bound for lift coefficient amplitude but also suggests to use a lower bound for drag offset phase during the control design, if the uncertainty of increased lift coefficient amplitude is high.

IV. CONCLUSIONS

In this paper, we studied the robustness of a flapping wing hovering system under aerodynamic parameter uncertainties. Three linearized controllers based on the full dynamics of the system were considered, each representing different control objectives. The results show that the controller focused on minimizing the change of our proposed control inputs has the slowest convergence rate, but the largest tolerance for aerodynamic modelling uncertainties. Moreover, for control design purposes, the results suggest that using the upper bound values for lift and drag coefficient amplitudes for the control design stage will have the largest tolerance domain. Also, the tolerance of the controller is asymmetric for the drag offset phase with low tolerance on positive values. Therefore, to maximize robustness, one should use a positive drag offset phase while simulating the aerodynamic forces during analysis and control design stage.

REFERENCES

[1] Robert J. Wood. The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Transactions on Robotics*, 24(2):341–347, 2008.
[2] J P Whitney and R J Wood. Conceptual design of flapping-wing micro air vehicles. *Bioinspiration & Biomimetics*, 7(3):036001, apr 2012.

[3] Alireza Ramezani, Soon-Jo Chung, and Seth Hutchinson. A biomimetic robotic platform to study flight specializations of bats. *Science Robotics*, 2(3):eaal2505, 2017.

[4] Zhan Tu, Fan Fei, Jian , and Xinyan Deng. An At-Scale Tailless Flapping-Wing Hummingbird and Experimental Validation. 36(5):1–15, 2020.

[5] Jialei Song, Haoxiang Luo, and Tyson L. Hedrick. Performance of a quasi-steady model for hovering hummingbirds, 2015.

[6] Haithem E. Taha, Muhammad R. Hajj, and Ali H. Nayfeh. Flight dynamics and control of flapping-wing MAVs: A review. *Nonlinear Dynamics*, 70(2):907–939, 2012.

[7] S A Ansari, R Żbikowski, and K Knowles. Non-linear unsteady aerodynamic model for insect-like flapping wings in the hover. part 1: Methodology and analysis. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 220(2):61–83, 2006.

[8] Xinyan Deng, Luca Schenato, and S. Shankar Sastry. Flapping flight for biomimetic robotic insects: Part II - Flight control design. *IEEE Transactions on Robotics*, 22(4):789–803, 2006.

[9] *A Passive Dynamic Approach for Flapping-Wing Micro-Aerial Vehicle Control*, volume ASME 2010 Dynamic Systems and Control Conference, Volume 1 of *Dynamic Systems and Control Conference*, 09 2010.

[10] H. Mahjoubi and K. Byl. Steering and horizontal motion control in insect-inspired flapping-wing mavs: The tunable impedance approach. In *2012 American Control Conference (ACC)*, pages 901–908, 2012.

[11] Joong-Kwan Kim, Jong-Seob Han, Jun-Seong Lee, and Jae-Hung Han. Hovering and forward flight of the hawkmoth manduca sexta: trim search and 6-DOF dynamic stability characterization. *Bioinspiration & Biomimetics*, 10(5):056012, sep 2015.

[12] Mao Sun. Insect flight dynamics : Stability and control. 86(June):615–646, 2014.

[13] Na Xu and Mao Sun. Lateral dynamic flight stability of a model bumblebee in hovering and forward flight. *Journal of Theoretical Biology*, 319:102–115, 2013.

[14] Na Gao, Hikaru Aono, and Hao Liu. Perturbation analysis of 6dof flight dynamics and passive dynamic stability of hovering fruit fly drosophila melanogaster. *Journal of Theoretical Biology*, 270(1):98–111, 2011.

[15] H. 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, 2019.

[16] Matej Karasek, Florian T. Muijres, Christophe De Wagter, Bart D.W. Remes, and Guido C.H.E. De Croon. A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns. *Science*, 361(6407):1089–1094, 2018.

[17] Matěj Karásek and André Preumont. Flapping flight stability in hover: A comparison of various aerodynamic models. *International Journal of Micro Air Vehicles*, 4(3):203–226, 2012.

[18] H. Glauert. *The Elements of Aerofoil and Airscrew Theory*. Cambridge University Press, 6 1983.

[19] 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.

[20] 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.

[21] Haithem E. Taha, Sevak Tahmasian, Craig A. Woolsey, Ali H. Nayfeh, and Muhammad R. Hajj. The need for higher-order averaging in the stability analysis of hovering, flapping-wing flight. *Bioinspiration and Biomimetics*, 10(1), 2015.

[22] 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.

[23] A. Serrani. Robust nonlinear control design for a minimally-actuated flapping-wing mav in the longitudinal plane. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7464–7469, 2011.

[24] A. Serrani. Robust hovering control of a single-dof flapping wing mav. In *Proceedings of the 2010 American Control Conference*, pages 1302–1307, 2010.

[25] M. Alkitbi and A. Serrani. Robust control of a flapping-wing mav by differentiable wingbeat modulation. *IFAC-PapersOnLine*, 49(18):290–295, 2016. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.

[26] 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.

[27] 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.

[28] Shiba Biswal, Marc Mignolet, and Armando A Rodriguez. Modeling and control of flapping wing micro aerial vehicles. *Bioinspiration & Biomimetics*, 14(2):026004, jan 2019.

[29] Ben Parslew. Predicting power-optimal kinematics of avian wings. *Journal of the Royal Society Interface*, 12(102), 2015.

[30] 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.

[31] J. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, second edition, 2010.

[32] Ali H. Nayfeh and Balakumar Balachandran. Applied Nonlinear Dynamics. *Applied Nonlinear Dynamics*, 1995.