

# Describing Robotic Bat Flight with Stable Periodic Orbits

Alireza Ramezani<sup>1</sup>, Syed Usman Ahmed<sup>1,2</sup>, Jonathan Hoff<sup>1,2</sup>,  
Soon-Jo Chung<sup>3</sup>, and Seth Hutchinson<sup>1,2</sup>

<sup>1</sup> Coordinated Science Laboratory

<sup>2</sup> Department of Electrical and Computer Engineering  
University of Illinois at Urbana-Champaign (UIUC)  
Urbana, IL 61801, USA

<sup>3</sup> Graduate Aerospace Laboratories (GAL)  
California Institute of Technology  
Pasadena, CA 91125, USA

**Abstract.** From a dynamic system point of view, bat locomotion stands out among other forms of flight. During a large part of bat wingbeat cycle the moving body is not in a static equilibrium. This is in sharp contrast to what we observe in other simpler forms of flight such as insects, which stay at their static equilibrium. Encouraged by biological examinations that have revealed bats exhibit periodic and stable limit cycles, this work demonstrates that one effective approach to stabilize articulated flying robots with bat morphology is locating feasible limit cycles for these robots; then, designing controllers that retain the closed-loop system trajectories within a bounded neighborhood of the designed periodic orbits. This control design paradigm has been evaluated in practice on a recently developed bio-inspired robot called Bat Bot (B2).

**Keywords:** Bio-inspired robot, bat, Poincare, periodic orbit, control

## 1 Introduction

Bats possess a very sophisticated powered flight mechanism among animals. This is evident by the dynamic conformation properties of their wings. Their flight mechanism has several types of joints, which connect the bones and muscles to one another and synthesize a metamorphic musculoskeletal system that possesses more than 40 degrees of freedom (DoFs), both passive and active [22]. A close look at bat flight kinematics reveals that bat sensory-motor control mobilizes several joints during a single flap cycle, which spans slightly less than 100 ms in some species. One flapping cycle involves two movements of i) a downstroke phase, which is initiated by both left and right forelimbs expanding backwards and sideways while sweeping downward and forward relative to the body, and ii) an upstroke phase, which brings the forelimbs upward and backward and is followed by flexion of the elbows and wrists to fold the wings.

In addition to quick multi-DoF actuation, bat jointed mechanism has adaptive morphing characteristics [1, 2] that behaves differently in various flight maneuvers. For instance, consider a roll maneuver performed by insectivorous bats [18]. Collapsing (bending) a wing and consequently reducing the wing area would increase wing loading on the collapsed wing, as a result, reducing the lift force. Furthermore, pronation of one wing and supination of the other wing yields negative and positive angles of attack, respectively, thereby generating negative and positive lift forces on the wing surface, causing the bat to roll sharply. Bats employ this move to hunt insects as in belly-up configuration they can use the natural camber on their wings to maximize descending acceleration. Agility of flight is vital for insectivore bats because their echolocating system cannot detect insects far away from them; with such restrictions in locating prey, sharp changes in flight direction are important.

From a dynamic system point of view, bat locomotion stands out among other forms of flight (e.g., insect flight) mainly due to the fact that during a large part of bat wingbeat cycle the moving body is not in a static equilibrium. In contrast, insects stay at their static equilibrium. This is because insect flapping frequency is faster than the body dynamic response. Based on this observation, conventional approximation methods such as the celebrated method of averaging is commonly applied to design flight controllers for insect-scale flapping robots [8, 7].

Encouraged by the biological examinations that have revealed bats exhibit periodic and stable limit cycles, this work demonstrates that one effective approach to stabilize articulated flying robots with bat morphology involves locating feasible limit cycles for these robots; then, designing controllers that retain the closed-loop system trajectories within bounded neighborhood of the designed periodic orbits. This work is a sequel to our work [19]. This work further investigates the effectiveness of the control design paradigm by conducting more experiments and simulations.

Control is the integral part of bat robots. Unfortunately often, due to mechanical design complications, these robots are not functional and cannot help explore various control design ideas. Therefore, there are not many control design schemes available. For example, Ro-bat [3] is just a wing design based on *Cynopterus brachyotis*' wing structure. The morphology of this robotic wing closely matches that of the real bat, with jointed legs, humerus, radius and digits. In practice it is impossible to accommodate several actuated coordinates in the wings as this can yield bulky wings and moment of inertia blows up. As a result, in Ro-bat, reducing the complexity from a real bat kinematics is realized by only considering a few joints. Three RC grade servo motors are used to articulate several joint in Ro-bat via a set of cables connected to key joints on the wing. These cables are embedded inside the armwing structure. The robotic wing can produce  $160^\circ$  wingbeat amplitude and 10 Hz wingbeats in flapping frequency. Additionally, the mechanism can realize two movements: i) retraction and protraction at the shoulder joints; ii) extension and flexion at the elbow. Many strings are routed through the armwing and with the help of this trans-

mission mechanism the digits can articulate with respect to the carpus. The robotic wing is not designed for self-sustained free flights as its several actuators make it very heavy. Therefore, it has no controller.

Robotbat[10] is developed with the same objective of studying physiological specializations in bats flight. This robot has an articulated flight mechanism that embodies three degrees of actuations (DoAs) per each wing and two DoAs for flapping amplitude variations. Overall, the robot has eight DoAs and the articulations are in the form of flapping, lead-lag, and feathering motions. The platform, which is mainly designed to run on CPG-based control schemes[6, 10], helped synthesize a bio-inspired flight controller employing a network of symmetric Hopf oscillators for a biologically inspired bat robot.

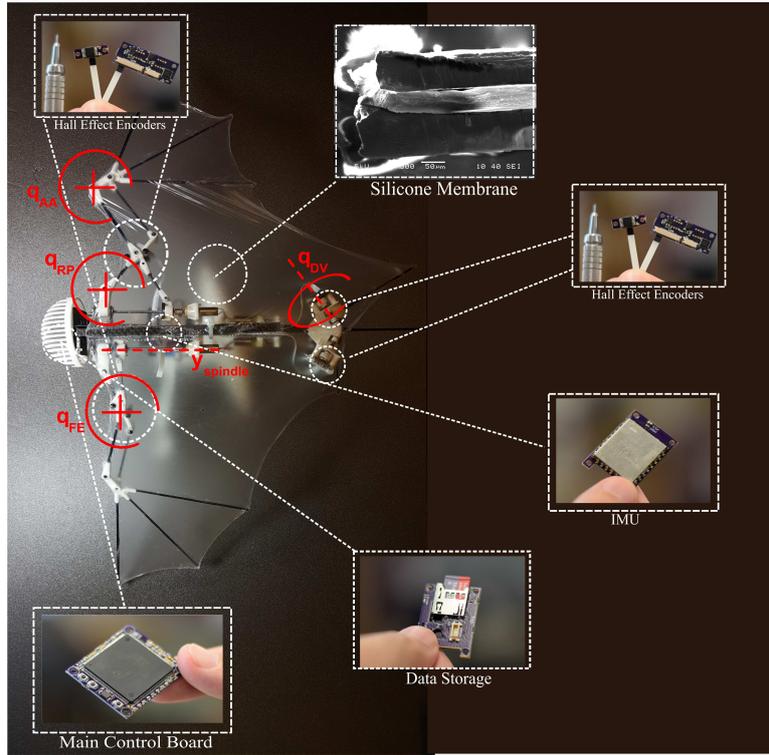


Fig. 1: Bat Bot (B2) [19].

In contrast to the previous examples, bat bot (B2), shown in Fig. 1, can perform free flight. It is a self-contained, autonomous flying robot that weighs 93 grams. This robot has been developed recently to mimic morphological properties of bat wings. Instead of using a large number of distributed control actuators, a highly stretchable membrane wings that are controlled at a reduced number

of dominant wing joints to best match the morphological characteristics of bat flight is implemented. B2 possesses some biologically meaningful DoFs. These biologically meaningful DoFs include asynchronous and mediolateral movements of the armwings and dorsoventral movements of the legs. The continuous surface and elastic properties of bat skin under wing morphing are realized by an ultra-thin (56 micrometers) membranous skin that covers the skeleton of the morphing wings.

This work employs B2 to explore the idea of describing articulated flapping flight with stable and attractive periodic orbits. B2 and conventional flying robots such as fixed-wing and rotary-wing robots are similar in that all of them rely on the air through modulation of the magnitude and direction of aerodynamic forces. However, the control surfaces in this robot are extremely different. Conventional fixed-wing robots are often controlled by thrust and conventional control surfaces such as elevators, ailerons, and rudders. In contrast, B2 possesses 9 active oscillatory joints (5 of which are independent) in comparison to 6 DoFs (attitude and position) that are actively controlled.

The approach that we explore in this paper relies on asymptotically imposing *virtual constraints* (holonomic constraints) on B2's dynamic system through closed-loop feedback. This concept has a long history, but its application in nonlinear control theory is primarily due to [5, 14]. Enforcing these constraints will yield a closed-loop system with periodic trajectories. Then, using the notion of Poincare return map the stability of these periodic trajectories is turned into the stability of the associated fixed-point on the Poincare section. Afterward, a discrete controller is utilized to retain the closed-loop trajectories at a bounded neighborhood of the fixed-point. The advantage of imposing these constraints through closed-loop feedback (software) rather than physically (hardware) is that B2's wing configurations can be adjusted and modified during the flight. We have tested this concept on B2 to generate cruise flights and we anticipate this potentially can help reconstruct the adaptive properties of bat flight for other maneuvers.

This work is organized as follows. First, the concept of virtual constraint, which drives the actuated portion of the system, is described in brevity. In Section 3, the concept of discrete control law is explained after introducing some basic mathematical concepts such as periodic orbit, stability of periodic orbit, Poincare return map, etc. Section 4 will present the performance of the controller in simulation and experiment and relevant discussion will be presented. Last but not least, the conclusion section makes the final remarks and discussions.

## 2 Virtual Constraints and Wing Conformations

For wing articulations, we employ a framework based on defining a set of parametrized and time-varying holonomic constraints [5, 14]. This method allows us to shape the overall system dynamics through such constraints. These holonomic constraints control the posture of the jointed structure of B2 by mobilizing the actuated

portion of the system and take place through the action of the servo actuators that are embedded in the robot.

First, we parametrize the jointed structure of B2 by several configuration variables. The configuration variable vector  $\mathbf{q}_{\text{morph}}$  defines the morphology of the forelimb and hindlimb as they evolve through the action of actuated coordinates and embodies nine biologically meaningful DoFs,

$$\mathbf{q}_{\text{morph}} = (q_{RP}^R; q_{FE}^R; q_{AA}^R; q_{DV}^R; q_{RP}^L; q_{FE}^L; q_{AA}^L; q_{DV}^L; q_{FL}) \quad (1)$$

where  $q_{RP}^i$  describes the retraction-protraction angle;  $q_{FE}^i$  is the radial flexion-extension angle;  $q_{AA}^i$  is the abduction-adduction angle of the carpus;  $q_{FL}$  is the flapping angle;  $q_{DV}^i$  is the dorsoventral movement of the hindlimb, see Fig. 1. Here, the superscript  $i$  denotes the right (R) or left (L) joint angles. Mathematically speaking, the mechanical constraints shown in Fig. 1 yield a nonlinear map from actuated joint angles

$$\mathbf{q}_{\text{act}} = (y_{\text{spindle}}^R; q_{DV}^R; y_{\text{spindle}}^L; q_{DV}^L; q_{FL}) \quad (2)$$

to the morphology configuration variable vector  $\mathbf{q}_{\text{morph}}$ . The spindle action shown in Fig. 1 is denoted by  $y_{\text{spindle}}^i$ . The nonlinear map is explained mathematically in [19, 20, 13]. Now, we proceed by incorporating the dynamics of the robot in the virtual constraint design procedure.

During free-fall ballistic motions, B2 with its links and joints represents an open kinematic chain that evolves under the influence of gravitational and external aerodynamic forces. Obtaining the equations of motion is not trivial. We employ the method of Lagrange to mathematically define this dynamics. This open kinematic chain is uniquely determined with the fuselage Euler angles roll, pitch, and yaw ( $q_x; q_y; q_z$ ); fuselage Center of Mass (CoM) positions ( $p_x; p_y; p_z$ ); and morphing joint angles  $\mathbf{q}_{\text{morph}}$ . Therefore, the robot's configuration variable vector is

$$\mathbf{q} = (q_x; q_y; q_z; p_x; p_y; p_z; \mathbf{q}_{\text{morph}}) \in \mathcal{Q} \quad (3)$$

where  $\mathcal{Q}$  is the robot's configuration variable space. We derived Lagrange equations after computing the total energy of the free open kinematic chain as the difference between the total kinetic energy and the total potential energy. Following Hamilton's principle of least action, the equations of motion for the open kinematic chain with ballistic motions are given by:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{Q}_{\text{gen}} \quad (4)$$

where  $\mathbf{M}$ ,  $\mathbf{C}$ , and  $\mathbf{G}$  are the inertial matrix, the Coriolis matrix, and the gravity vector, respectively. The generalized forces  $\mathbf{Q}_{\text{gen}}$ , which reflect the role of aerodynamic forces as well the action of several morphing motors in B2, are described in [19, 21]. Now, the virtual constraints are given by

$$\mathbf{N}(t, \boldsymbol{\beta}, \mathbf{q}_{\text{act}}) = \mathbf{q}_{\text{act}} - \mathbf{r}_{\text{des}}(t, \boldsymbol{\beta}) \quad (5)$$

where  $\mathbf{r}_{\text{des}}$  is the time-varying desired trajectory associated to the actuated coordinates,  $t$  is time, and  $\boldsymbol{\beta}$  is the vector of the wing kinematic parameters.

These parameters are explained later and we will show how the wing kinematic parameters can be assigned as control commands in the closed-loop system. Once Eq. 5 is enforced (with a feedback law) the posture of B2 varies as the actuated portion of the system now implicitly follows the time-varying trajectory  $\mathbf{r}_{\text{des}}$ . This is simply a classical tracking problem. In order to design  $\mathbf{r}_{\text{des}}$ , we pre-compute the evolution of B2's joint trajectories for  $\mathbf{N}=0$ .

We partition the configuration variable vector  $\mathbf{q}$  into the actuated coordinates  $\mathbf{q}_{\text{act}}$  and the remaining coordinates  $\mathcal{X}$ , which includes Euler angles and body CoM positions. The dynamics Eq. 4 are re-written as following

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathcal{X}} \\ \ddot{\mathbf{q}}_{\text{act}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathcal{X}} \\ \dot{\mathbf{q}}_{\text{act}} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix} = \mathbf{Q}_{\text{gen}}. \quad (6)$$

In the above equation,  $\mathbf{M}_{11}$ ,  $\mathbf{M}_{12}$ ,  $\mathbf{M}_{21}$ ,  $\mathbf{M}_{22}$ ,  $\mathbf{C}_{11}$ ,  $\mathbf{C}_{12}$ ,  $\mathbf{C}_{21}$ , and  $\mathbf{C}_{22}$  are block matrices. The nonlinear system in Eq. 6 shows that the actuated and unactuated dynamics are coupled by the inertial, Coriolis, gravity, and aerodynamic terms.

The actuated dynamics (see Eq. 6) can be reconstructed from the virtual constraints by simply taking time derivative of Eq. 5 and computing the acceleration terms. By integrating Eq. 6 subject to Eq. 5 constraints, it is possible to pre-compute the evolution of  $\mathcal{X}$  coordinates. In addition, it is possible to optimize wing kinematic parameters  $\boldsymbol{\beta}$  to design specific flight kinematic properties [19, 21]. In other words, the algebraic-differential equation (ADE) given by Eq. 6 and Eq. 5 can be solved iteratively to located predefined feasible trajectories for the actuated coordinates in the robot.

Turning to the constraint equation,  $\mathbf{r}_{\text{des}}$  is given by

$$\mathbf{r}_{\text{des}}^i(t, \boldsymbol{\beta}) = a_i \cos(\omega t + \phi_i) + b_i, \quad i \in \{y_{\text{spindle}}^R, q_{DV}^R, y_{\text{spindle}}^L, q_{DV}^L, q_{FL}\} \quad (7)$$

where  $\boldsymbol{\beta} = \{\omega, \phi_i, a_i, b_i\}$  parametrizes the periodic actuator trajectories that define the wing motion. These parameters are the control input to the system. These parameters are changed by a discrete feedback law, which will be explained later, to keep the closed-loop system within a bounded neighborhood of the predefined periodic orbits.

### 3 Closed-Loop Control

#### 3.1 Continuous Feedback Law

The responsibility of the continuous feedback law is to enforce the previously explained virtual constraints Eq. 5 on the dynamic model given by Eq. 4 asymptotically and in finite time. There are several tools from nonlinear control theory (e.g., feedback linearization, control Lyapunov function (CLF), etc.) that can be employed to enforce these constraints in finite time. In this work we focus on the discrete feedback policy and for a comprehensive continuous control design instruction refer to [15].

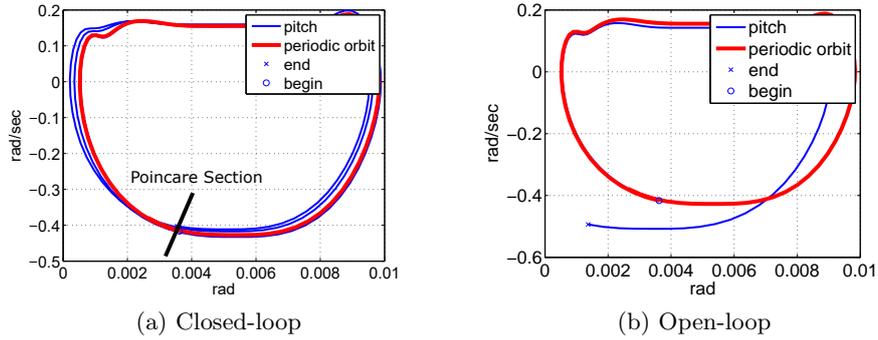


Fig. 2: Comparison between the closed-loop and open-loop systems in keeping the pitch angle trajectory (blue) within a bounded neighborhood of the designed periodic orbit (red).

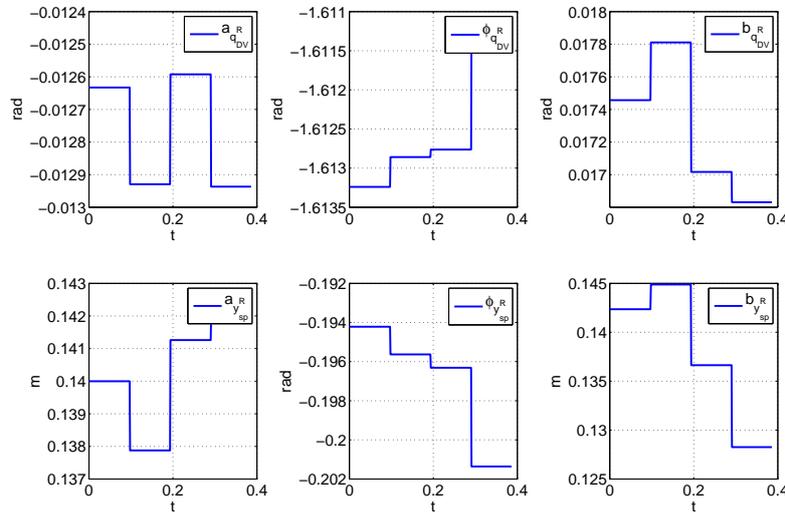


Fig. 3: Flight kinematic parameter  $\beta$  is updated by the discrete control law.

### 3.2 Discrete Feedback Law

Enforcing the constraints Eq. 5 to the model by a continuous feedback law will not guarantee that the system trajectories stay within a bounded neighborhood of the designed predefine periodic orbits. This is shown in Fig. 2, where the simulated open-loop trajectories diverge from the actual periodic orbit. Therefore, another control loop is required to keep the trajectories within a bounded neighborhood of the designed orbits. We achieve this by designing a discrete

controller that steers the system within a bounded neighborhood of the optimal trajectories by updating the wing kinematic parameters  $\beta$  during each wingbeat.

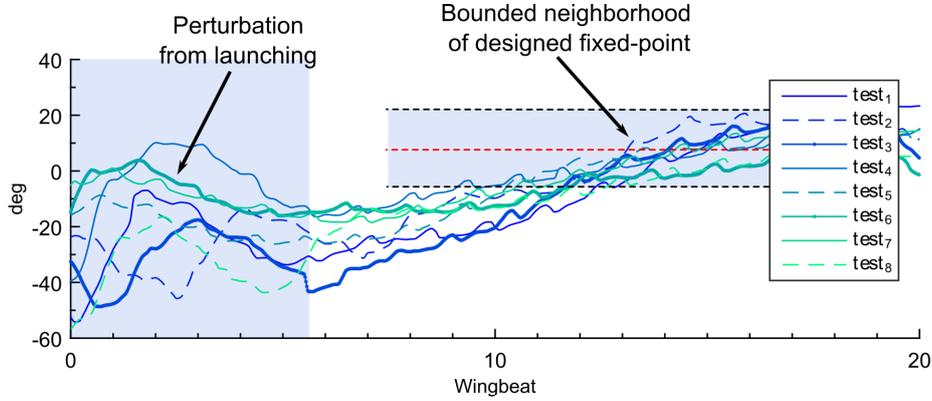


Fig. 4: Pitch angle evolutions for eight untethered closed-loop flights. The first shaded region depicts the perturbations that are introduced to the system after the launch. The second shaded region shows the bounded neighborhood of the fixed-point. The closed-loop trajectories are steered to this region by the controller.

Our discrete closed-loop feedback synthesis for B2 is based on the notion of stability for periodic orbits. We start with some basic definitions for autonomous systems and obviously with slight modifications can be applied to nonautonomous systems. Some of the the material that is presented here is a reiteration of the material in [17, 15, 16, 4, 9, 11, 12, 23].

A solution  $\varphi : [t_0, \infty) \rightarrow \mathcal{U}^4$  is a periodic solution of the autonomous system

$$\dot{x} = f(x) \quad (8)$$

if for all  $t \in [t_0, \infty)$

$$\varphi(t + T) = \varphi(t) \quad (9)$$

for some minimum period  $T > 0$ . A set  $\mathcal{O}$  is a periodic orbit of Eq. 8 if

$$\mathcal{O} = \{\varphi(t) | t \geq t_0\} \quad (10)$$

for some periodic solution  $\varphi(t)$ . An orbit is nontrivial if it contains more than one point (59).

Turning to stability, Lyapunov stability (49) of the periodic orbit  $\mathcal{O}$  is formally stated as if there exists a neighborhood  $\mathcal{V}$  of the periodic orbit  $\mathcal{O}$  such that

<sup>4</sup>  $\mathcal{U}$  is a smooth embedded submanifold of  $\mathbb{R}^n$ .

for every point  $p$  in the neighborhood  $\mathcal{V}$ , there exists a solution  $\varphi : [t_0, \infty) \rightarrow \mathcal{U}$  of the autonomous system given by Eq. 8 satisfying  $\varphi(0) = p$  and

$$\text{dist}(\varphi(t), \mathcal{O}) < \epsilon \quad (11)$$

for all  $t \geq 0$ , where  $\epsilon > 0$ , and  $\text{dist}(p_1, p_2)$  is the Euclidean distance. The orbit  $\mathcal{O}$  is attractive if there exists an open neighborhood  $\mathcal{V}$  of  $\mathcal{O}$  such that for every  $p \in \mathcal{V}$ , there exists a solution  $\varphi : [t_0, \infty) \rightarrow \mathcal{U}$  of Eq. 8 satisfying  $\varphi(0) = p$  and

$$\lim_{t \rightarrow \infty} \text{dist}(\varphi(t), \mathcal{O}) = 0. \quad (12)$$

The periodic orbit  $\mathcal{O}$  is asymptotically stable in the sense of Lyapunov if it is both stable and attractive. Exponential stability of the orbit  $\mathcal{O}$  is achieved if there exists a neighborhood  $\mathcal{V}$  of  $\mathcal{O}$  such that for every  $p \in \mathcal{V}$ , there exists a solution  $\varphi : [0, \infty) \rightarrow \mathcal{U}$  of the autonomous system given by Eq. 8 satisfying  $\varphi(0) = p$  and

$$\text{dist}(\varphi(t), \mathcal{O}) \leq N \exp(-\gamma t) \text{dist}(p, \mathcal{O}) \quad (13)$$

where  $N$  and  $\gamma$  are positive constants.

Two mathematical tools: i) *Poincare section* and ii) *Poincare return map* can be used to evaluate the stability of periodic orbits. A Poincare section is a smooth hypersurface  $\mathcal{S}$  in  $\mathcal{U}$  and it satisfies the following conditions:

- $\mathcal{S}$  is nonempty and there exists a differentiable function  $H : \mathcal{U} \rightarrow \mathbb{R}$  such that

$$\mathcal{S} := \{x \in \mathcal{U} | H(x) = 0\}; \quad (14)$$

- for every  $s \in \mathcal{S}$

$$\frac{\partial H}{\partial x}(s) \neq 0. \quad (15)$$

This condition implies that the periodic orbit  $\mathcal{O}$  is transversal to  $\mathcal{S}$ ;  $\mathcal{S}$  has a lower dimension than  $\mathcal{U}$ .

Any Poincare return map  $\mathbf{P} : \mathcal{S} \rightarrow \mathcal{S}$  defines a discrete dynamic system by

$$x_{k+1} = \mathbf{P}(x_k) \quad (16)$$

where  $x_k \in \mathcal{S}$  is the state and  $\mathbf{P}$  maps the state to the next state  $x_{k+1}$ . For this discrete system

$$\mathbf{P}(x^*) = x^* \quad (17)$$

$x^*$  is the fixed-point. When starting from an initial state  $x_0$ , successive application of the map  $\mathbf{P}$  generates a sequence of states that can reveal useful information about the stability of the discrete time system [17]. In other words, in the method of Poincare, there is a relationship between the periodic orbits of the system (Eq. 8) and the equilibrium points of the sampled system above. This method is interesting because it establishes an equivalence between the stability properties of the periodic orbits of Eq. 8 and the equilibrium points of

the discrete system Eq. 16. This equivalence can be formally expressed in form of a theorem [23].

To stabilize the designed periodic solution, we augmented the desired trajectory  $\mathbf{r}_{\text{des}}$  with a correction term [23]

$$\mathbf{r}_{\text{corr}} = \frac{\partial \mathbf{r}_{\text{des}}}{\partial \boldsymbol{\beta}} \delta \boldsymbol{\beta}. \quad (18)$$

The Poincare return map takes the robot states  $\mathbf{q}_k$  and  $\dot{\mathbf{q}}_k$  (the Euler angles roll, pitch, yaw and their rates) at the beginning of  $k$ -th flapping cycle and leads to the states at the beginning of the next flapping cycle,

$$\begin{bmatrix} \mathbf{q}_{k+1} \\ \dot{\mathbf{q}}_{k+1} \end{bmatrix} = \mathbf{P}(\mathbf{q}_k, \dot{\mathbf{q}}_k, \mathbf{r}_{\text{corr}}). \quad (19)$$

We linearized the map  $\mathbf{P}$  at  $\mathcal{S}$ , resulting in a dynamic system that describes the periodic behavior of the system at the beginning of each flapping cycle

$$\begin{bmatrix} \delta \mathbf{q}_{k+1} \\ \delta \dot{\mathbf{q}}_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \mathbf{q}}(\mathbf{q}^*) & \frac{\partial \mathbf{P}}{\partial \dot{\mathbf{q}}}(\dot{\mathbf{q}}^*) \end{bmatrix} \begin{bmatrix} \delta \mathbf{q}_k \\ \delta \dot{\mathbf{q}}_k \end{bmatrix} + \frac{\partial \mathbf{P}}{\partial \boldsymbol{\beta}}(\boldsymbol{\beta}^*) \delta \boldsymbol{\beta}_k \quad (20)$$

where  $(^*)$  denotes the equilibrium points and  $(\delta \mathbf{q}; \delta \dot{\mathbf{q}})$  denotes deviations from the equilibrium points. The changes in the kinematic parameters are denoted by  $\delta \boldsymbol{\beta}$ . Here, the stability analysis of the periodic trajectories of the bat robot are relaxed to the stability analysis of the equilibrium of the linearized Poincare return map on  $\mathcal{S}$ . As a result, classical feedback design tools can be applied to stabilize the system. We computed a constant state-feedback gain matrix  $\mathbf{K}_{\mathcal{S}}$  such that the closed-loop linearized map is exponentially stable:

$$\text{eig} \left( \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \mathbf{q}}(\mathbf{q}^*) & \frac{\partial \mathbf{P}}{\partial \dot{\mathbf{q}}}(\dot{\mathbf{q}}^*) \end{bmatrix} + \frac{\partial \mathbf{P}}{\partial \boldsymbol{\beta}}(\boldsymbol{\beta}^*) \mathbf{K}_{\mathcal{S}} \right) < 1 \quad (21)$$

We used this state feedback policy at the beginning of each flapping cycle in order to update the kinematic parameters as follows

$$\delta \boldsymbol{\beta}_k = \mathbf{K}_{\mathcal{S}} \begin{bmatrix} \delta \mathbf{q}_k \\ \delta \dot{\mathbf{q}}_k \end{bmatrix}. \quad (22)$$

## 4 Results

Figure 2 shows the simulated results and the performance of the closed-loop system that keep the pitch angle trajectories in a bounded neighborhood of the design orbit. Figure 3 shows the wing kinematic parameters. It is worth noting that pitch dynamics instabilities are unavoidable in open-loop tailless robots as the tail that passively damps out external perturbations is missing. As is shown in Fig. 2, the robot diverges from the designed orbit immediately after the discrete controller is turned off.

We performed extensive untethered flight experiments in a large indoor space (Stock Pavilion at the University of Illinois in Champaign-Urbana) where we could use a net (30m by 30m) to protect the sensitive electronics of B2 at the moment of landing. The flight arena was not equipped with any motion capture system. Although the vehicle landing position was adjusted by an operator to secure landings within the area, which is covered by the net, we landed outside the net many times. The launching task was performed by a human operator, thereby adding to the degree of inconsistency of the launches. In all of these experiments, at the launch moment, the system reached its maximum flapping speed (10 Hz). The hand launch introduced initial perturbations, which considerably affected the first 10 wingbeats. Despite the external perturbations of the launch moment, the vehicle stabilized the pitch angle within 20 wingbeats. In Fig. 4, the pitch angles from experiments are shown. Figure 4 shows how the controller steers the pitch angle trajectories towards the designed trajectories despite external perturbations.

## 5 Concluding Remarks

From a dynamic system point of view, bat locomotion stands out among other forms of flight. During a large part of bat wingbeat cycle the moving body is not in a static equilibrium. Encouraged by this observation, we explored the idea of introducing virtual constraints on B2, which is bio-inspired robot with bat morphology, through closed-loop feedback. Enforcing these constraints will yield a closed-loop system with periodic trajectories. By employing the notion of Poincaré return map the stability of these periodic trajectories was turned into the stability of the associated fixed-point on the Poincaré section where we used a discrete controller to retain the closed-loop trajectories at a bounded neighborhood of the fixed-point. Our work suggests that one effective way to design controller for articulated flapping systems involves locating feasible periodic orbits and designing controllers that retain the closed-loop system in a bounded neighborhood of these orbits.

## References

1. H. Aldridge. Kinematics and aerodynamics of the greater horseshoe bat, *rhinolophus ferrumequinum*, in horizontal flight at various flight speeds. *Journal of Experimental Biology*, 126(1):479–497, 1986.
2. H. Aldridge. Body accelerations during the wingbeat in six bat species: the function of the upstroke in thrust generation. *Journal of Experimental Biology*, 130(1):275–293, 1987.
3. J. W. Bahlman, S. M. Swartz, and K. S. Breuer. Design and characterization of a multi-articulated robotic bat wing. *Bioinspiration & Biomimetics*, 8(1):016009, 2013.
4. R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.

5. C. I. Byrnes and A. Isidori. A frequency domain philosophy for nonlinear systems, with applications to stabilization and to adaptive control. In *IEEE Conference on Decision and Control*, pages 1569–1573. IEEE, 1984.
6. S.-J. Chung and M. Dorothy. Neurobiologically inspired control of engineered flapping flight. *Journal of Guidance, Control, and Dynamics*, 33(2):440–453, 2010.
7. X. Deng, L. Schenato, and S. S. Sastry. Flapping flight for biomimetic robotic insects: Part ii-flight control design. *IEEE Transactions on Robotics*, 22(4):789–803, 2006.
8. X. Deng, L. Schenato, W. C. Wu, and S. S. Sastry. Flapping flight for biomimetic robotic insects: Part i-system modeling. *IEEE Transactions on Robotics*, 22(4):776–788, 2006.
9. J. B. Dingwell and J. P. Cusumano. Nonlinear time series analysis of normal and pathological human walking. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 10(4):848–863, 2000.
10. M. Dorothy and S.-J. Chung. Methodological remarks on CPG-based control of flapping flight. In *AIAA Atmospheric Flight Mechanics Conference*, 2010.
11. M. S. Garcia. *Stability, scaling, and chaos in passive-dynamic gait models*. PhD thesis, Cornell University, 1999.
12. J. Guckenheimer and S. Johnson. Planar hybrid systems. In *International Hybrid Systems Workshop*, pages 202–225. Springer, 1994.
13. J. Hoff, A. Ramezani, S.-J. Chung, and S. Hutchinson. Synergistic design of a bio-inspired micro aerial vehicle with articulated wings. In *The Robotics: Science and Systems (RSS)*, 2016.
14. A. Isidori and C. Moog. On the nonlinear equivalent of the notion of transmission zeros. In *Modelling and Adaptive Control*, pages 146–158. Springer, 1988.
15. H. K. Khalil and J. Grizzle. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.
16. G. Meurant. *An introduction to differentiable manifolds and Riemannian geometry*, volume 120. Academic press, 1986.
17. A. H. Nayfeh. Perturbation methods in nonlinear dynamics. In *Lecture Notes in Physics*, pages 238–314. Springer, 1986.
18. U. M. Norberg. Some advanced flight manoeuvres of bats. *Journal of Experimental Biology*, 64(2):489–495, 1976.
19. A. Ramezani, S.-J. Chung, and S. Hutchinson. A biomimetic robotic platform to study flight specializations of bats. *Science Robotics*, 2(3):eaal2505, 2017.
20. A. Ramezani, X. Shi, S.-J. Chung, and S. Hutchinson. Bat bot (b2), a biologically inspired flying machine. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
21. A. Ramezani, X. Shi, S.-J. Chung, and S. Hutchinson. Lagrangian modeling and flight control of articulated-winged bat robot. *International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, September 28–October 02, 2015.
22. D. K. Riskin, D. J. Willis, J. Iriarte-Díaz, T. L. Hedrick, M. Kostandov, J. Chen, D. H. Laidlaw, K. S. Breuer, and S. M. Swartz. Quantifying the complexity of bat wing kinematics. *Journal of Theoretical Biology*, 254(3):604–615, 2008.
23. E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback control of dynamic bipedal robot locomotion*, volume 28. CRC press, 2007.