UNIVERSITY OF CALIFORNIA,
IRVINE

Experimental Demonstration of the Vibrational Stability Phenomenon in Bio-inspired Flight

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Mechanical and Aerospace Engineering

by

Mohammadali Kiani

Thesis Committee:
Professor Haithem E. Taha, Chair
Professor Tryphon Georgiou
Professor Kenneth Mease

2018

DEDICATION

To:

My mother, who put incredible commitment and energy to support and care for me over the years. I would not have gotten where I am today without her unconditional love and incredible sacrifices.

My father, who guided me throughout my whole life, always believed in me and trusted my capabilities. I would not be who I am today if it were not for him teaching me manners and lighting my path to be a man.

My uncle, who has supported me more than I could have ever imagined. I am incredibly fortunate to have someone like him in my life.

TABLE OF CONTENTS

Page

LIST OF FIGURES **v**

ACKNOWLEDGMENTS **vi**

CURRICULUM VITAE **vii**

ABSTRACT OF THE THESIS **viii**

1 Introduction **1**

2 Modeling **3**
2.1 Longitudinal Flight Dynamics of Bio-inspired Flight Near Hover 3

3 Averaging **7**
3.1 An Introduction to Averaging 7
3.2 Stability analysis of an FWMAV using direct averaging 8
3.3 Stability analysis of an FWMAV using Floquet theorem approach 11
3.4 Stability analysis of an FWMAV using higher order averaging 12

4 Vibrational stabilization **14**
4.1 Vibrational stabilization in inverted pendulum 14
4.2 Proposed experimental setup to verify vibrational stability 15
4.3 Measuring data of degrees of freedom 17
4.4 Experimental Demonstration 21
4.5 Break down of the setup's component 29

5 Physics of Vibrational Stabilization in Insect Flight Dynamics **38**

6 Conclusion and Future Work **42**
6.1 Conclusion 42
6.2 Future work 43

Bibliography **45**

A Visual studio and Arduino Codes **48**

A.0.1 Color detection code (Source.cpp) in C++ 48

A.0.2 Class Header File (SerialPort.h) communication between C++ and Arduino 55

A.0.3 Class Source File (SerialPort.cpp) communication between C++ and Arduino 57

A.0.4 Arduino code to communicate between Arduino Uno and Visual studio 63

LIST OF FIGURES

Page

2.1 Schematic diagram of a hovering flapping bird in the longitudinal plane $x - z$. 4

4.1 flapping setup attached to the pendulum 18
4.2 flapping setup 19
4.3 Gravity 360 Degree Hall Angle Sensor 19
4.4 Flowchart of the sensor system 20
4.5 Robotic flapper motion from (a) zero voltage (no flapping) to (d) high voltage (fast flapping) hovering equilibrium. At zero applied voltage (no flapping), the body is standing vertically ($\theta \simeq 90°$) at the bottom position of the pendulum. As the voltage and consequently the flapping frequency increases, the body moves upward along the circular path of the pendulum (i.e., γ increases) and tilts forward towards the horizontal attitude (i.e., θ decreases). 22
4.6 flapping setup's unstable response at relatively low flapping frequency (12Hz). 23
4.7 flapping setup's response as the flapping frequency is being manually increased. 24
4.8 flapping setup's stable response at relatively high flapping frequency (18Hz). 25
4.9 Propeller pendulum setup 27
4.10 Unstable Response of the Two-DOF Pendulum setup. 28
4.11 Assembly of the experimental setup 29
4.12 (a) 3 view drawing of the test stand. (b) Isometric drawing of the test stand. 30
4.13 Hall sensor mount 31
4.14 Isometric view of the L-shaped arm used to connect flapping setup to the pendulum 32
4.15 3 view drawing of the L-shaped arm used to connect flapping setup to pendulum 33
4.16 Isometric view of the fabricated body of the flapping setup 33
4.17 3 view drawing of the fabricated body of the flapping setup 34
4.18 Simplified schematic of the setup from the hall sensor mount to the body of flapping setup 35
4.19 Propeller setup attached to the arm in a new orientation 36
4.20 3 view drawing of the arm used in the propeller setup 37

5.1 Graphical illustration of the physics behind the vibrationally-induced pitch stiffness. 40

ACKNOWLEDGMENTS

I would like to extend my appreciation to the Department of Mechanical and Aerospace Engineering for giving me the chance to enroll in this program. Also, I am very thankful of all of my teachers every one of whom opened my eye to a vast area of knowledge. I highly doubt I could have such a delightful experience elsewhere.

Professor Taha, I simply cannot express how grateful I am for the lessons that I have learned from you. Mechanical engineering was not my background, yet you spent countless hours mentoring me patiently without showing any sign of exhaustion or discomfort even when I had difficulty understanding primary subjects. Not only are you an academic role model, but also an inspiring source to learn ethics and values from. I have learned a lot in UCI, but among them, morality and manners that I have grasped from you is the most valuable trophy that will direct me throughout my life. You have helped me a lot more than you know.

Few people have the privilege of having such supportive and giving friends. I just wanted to write to let you know how much I appreciate the positive influence you have had in my life. Amir Saman, Ahmed, Amirhossein, Miquel, Joel, Ali, Milad, Zahra, and Joseph, Thank you for your concern, useful advice and feedback! I will be forever grateful.

Some the text of this thesis/dissertation is a reprint of the material as it appears in *Experimental Demonstration of the Vibrational Stabilization Phenomenon in Bio-Inspired Flying Robots,* in *IEEE Robotics and Automation Letters*, 3(2), 643-647. The first author listed in this publication directed and supervised research which forms the basis for this thesis.

Permission to incorporate parts of previously published work into this dissertation has been granted by Prof. Haithem E. Taha and Joel Navarro.

CURRICULUM VITAE

Mohammadali Kiani

EDUCATION

Master of Science in Mechanical and Aerospace Engineering **2018**
University of California, Irvine *Irvine, California*

Bachelor of Science in Electrical Engineering **2016**
Iran University of Science and Technology *Tehran, Iran*

RESEARCH EXPERIENCE

Graduate Researcher **2017–2018**
University of California, Irvine *Irvine, California*

Undergraduate Researcher **2014–2016**
Iran University of Science and Technology *Tehran, Iran*

WORK EXPERIENCE

Senior Design Project Adviser **2017–2018**
University of California, Irvine *Irvine, California*

ABSTRACT OF THESIS

Experimental Demonstration of the Vibrational Stability
Phenomenon in Bio-inspired Flight

By

Mohammadali Kiani

Master of Science in Mechanical and Aerospace Engineering

University of California, Irvine, 2018

Professor Haithem E. Taha, Chair

Flapping wing micro-air-vehicles (FWMAV) are micro-air-vehicles that use biomimetic actuation (oscillatory flapping wing) for lift, propulsion, and control. The dynamic behavior of these bio-inspired systems is expressed by a multi-body, multi-time-scale, nonlinear, time-varying dynamical system model. This rich dynamic leads to unconventional stabilization mechanisms whose study essentially necessitates a mathematically rigorous analysis. Higher-order averaging, which is based on chronological calculus, can be utilized to show that insects and their man-made mimics (FWMAVs) exploit *vibrational control* to stabilize their body pitching angle. Such an unconventional stabilization cannot be proved by direct averaging. To experimentally demonstrate this phenomenon, an experimental setup was constructed. This setup allowed for two degrees of freedom for the body; vertical motion and pitching motion. It is found that there is a flapping frequency threshold beyond which the body pitching response is *naturally* (without feedback) stabilized, which conforms with the vibrational control concept. Moreover, a replica of the setup is fabricated with the FWMAV being replaced by a propeller revolving at a constant speed, which results in a constant aerodynamic force, leaving no room for vibrational control. The response of the propeller-setup is unstable at all frequencies, which also verifies the fact that the observed stabilization of the FWMAV-setup at high frequencies is due to vibrational stabilization.

Chapter 1

Introduction

In general, oscillatory control inputs can create stabilizing actions meaning that unstable equilibrium of system can gain stability due to a high-frequency high amplitude oscillatory control input. One famous case study of this phenomenon is the inverted pendulum also known as the Kapitza pendulum. It is known that the inverted pendulum has an unstable equilibrium. However, the unstable equilibrium of the inverted pendulum gains asymptotic stability when the pivot oscillates vertically at a sufficiently high frequency. The whole phenomenon happens entirely without implementing any closed-loop control. Crude averaging of the simple equations governing the dynamics of the Kapitza pendulum [16, 17] (inverted pendulum whose pivot is subjected to a vertical oscillation) showed no stabilization due to the pivot vibration. However, appropriate averaging techniques; whether it is higher-order averaging [23, 38] based on chronological calculus [4]; higher-order averaging [40, 14] based on Lie transform [19] or direct averaging exploiting the nonlinear variation of constants formula [22, 8]; can capture the vibrationally-induced stabilizing stiffness. Similarly, in nature, the oscillatory motion of insect wings makes the generated flight forces of time-varying nature. As a result, the body of an insect hovering over a flower oscillates almost in all directions; on the average, it is hovering over the flower. Therefore, averaging seems quite intuitive to analyze insect flight dynamics. Following the averaging approach, there has been almost a consensus among flapping flight experts within both biology and engineering communities that insects are unstable at hover. Classical control theorems yield eigenvalues of the stability analysis of the flapping wings flight with positive real parts [35, 36, 27, 26, 39, 13, 12, 9]. mainly due to lack of pitch stiffness. However, using higher order averaging and differential geometry, it can be shown that flapping wing flight can be stable while hovering. The recent efforts by Taha et al. [33, 32, 34, 15] showed an induced vibrational stabilization mechanism in the form of pitch stiffness on the flight dynamics of these bio-inspired robots. In the content of this thesis, we discuss previous efforts using higher-order averaging to show the vibrationally-induced pitch stiffness. Moreover, we experimentally demonstrate such a phenomenon on a flapping apparatus that allows two degrees-of-freedom (DOF) for the body of an FWMAV. In the content of this thesis FWMAV setup and flapping setup are used interchangeably.

Chapter 2

Modeling

2.1 Longitudinal Flight Dynamics of Bio-inspired Flight Near Hover

Figure (2.1) shows a flapping bird at hovering position in the longitudinal plane $x - z$. There exist three DOFs; translations along the x-axis and z-axes in addition to velocity components u and w. Also a rotation about the y axis shown by an angle $theta$ and an angular velocity q. The generalized forces X, Z, and M are the aerodynamic forces in the x and z directions. And M is the aerodynamic pitching moment about the y-axis. Longitudinal equations of motion will be achieved similar to that of a conventional aircraft if the wing structural and inertial effects are neglected [21].

$$\begin{pmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -qw - g\sin\theta \\ qu + g\cos\theta \\ 0 \\ q \end{pmatrix} + \begin{pmatrix} X/m \\ Z/m \\ M/I_y \\ 0 \end{pmatrix}, \tag{2.1}$$

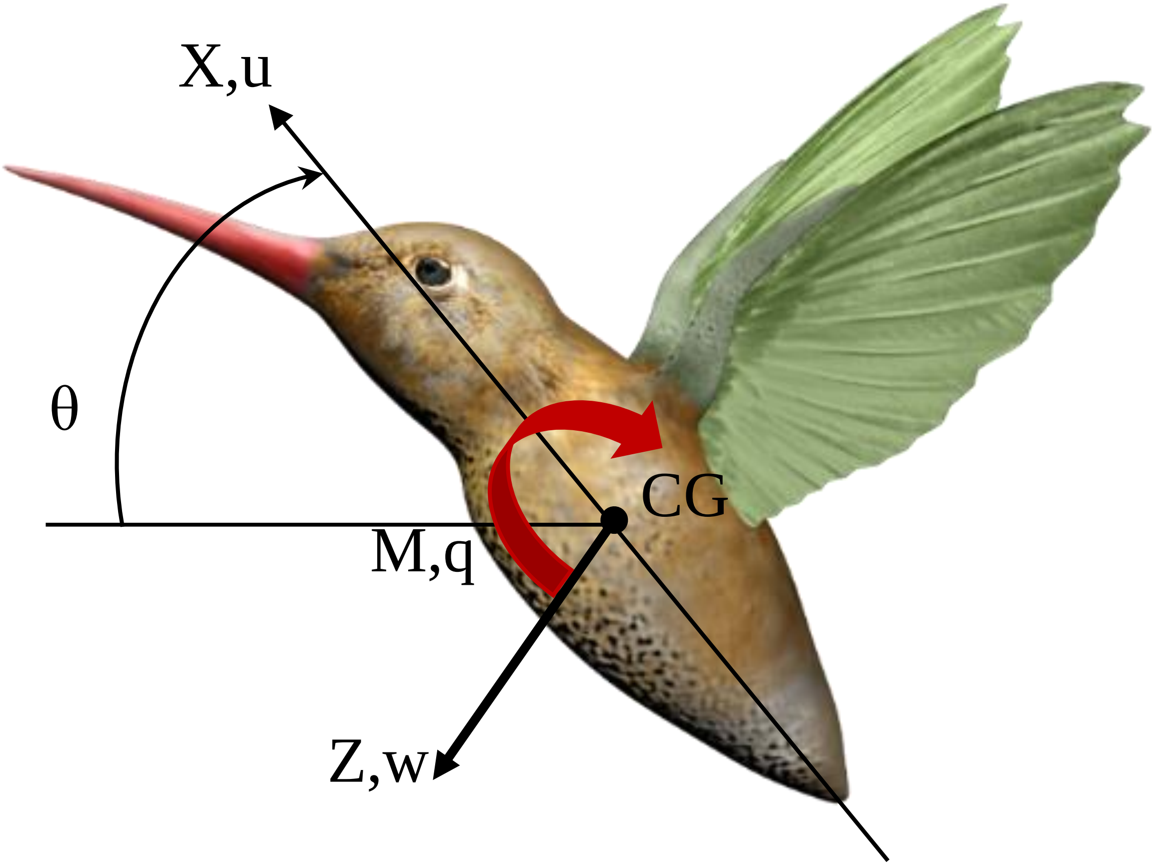

Figure 2.1: Schematic diagram of a hovering flapping bird in the longitudinal plane $x - z$.

Where M is the body mass and I_y is the body pitching inertia and g is the gravitational acceleration. Contradictory to conventional aircraft the aerodynamic loads X,Z, and M are essentially time-varying thus the system (2.1) can be expressed in the following form. In [30] Taha et al. approximated the above relation as

$$\begin{pmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -qw - g\sin\theta \\ qu + g\cos\theta \\ 0 \\ q \end{pmatrix} + \begin{pmatrix} X_0(t)/m \\ Z_0(t)/m \\ M_0(t)/I_y \\ 0 \end{pmatrix} + \begin{bmatrix} X_u(t) & X_w(t) & X_q(t) & 0 \\ Z_u(t) & Z_w(t) & Z_q(t) & 0 \\ M_u(t) & M_w(t) & M_q(t) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} u \\ w \\ q \\ \theta \end{pmatrix} \tag{2.2}$$

presuming a piecewise constant variation in the wing pitch angle η and horizontal stroke plane, parameterized by the "back-and-forth" flapping angle φ, the following relation can be achieved. [30]

$$X_0(t) = -2K_{21}\dot{\varphi}(t)|\dot{\varphi}(t)| \cos\varphi(t) \sin^2\eta \ \ , \ \ Z_0(t) = -K_{21}\dot{\varphi}(t)|\dot{\varphi}(t)| \sin 2\eta$$

$$M_0(t) \ = \ 2\dot{\varphi}(t)|\dot{\varphi}(t)| \sin\eta[K_{22}\Delta\hat{x} \cos\varphi(t) + K_{21}x_h \cos\eta(t) + K_{31} \sin\varphi(t) \cos\eta]$$

where $\Delta\hat{x}$ is the chordwise distance from the center of pressure to this same hinge location, normalized by the chord length and x_h is the distance from the vehicle center of mass to the root of the wing hinge line (i.e., where the hinge line intersects the x-axis). On the other hand, C_{L_α} is the three-dimensional lift curve slope of the wing, ρ is the air density, $c(r)$ is the spanwise chord distribution, $I_{mn} = 2\int_0^R r^m c^n(r)\, dr$, R is the wing radius, and $K_{mn} = \frac{1}{4}\rho C_{L_\alpha} I_{mn}$. In [30] The time-varying stability derivatives are obtained directly in terms of the system parameters as

$$X_u = -4\tfrac{K_{11}}{m}|\dot{\varphi}| \cos^2\varphi \sin^2\eta, \quad X_w = -\tfrac{K_{11}}{m}|\dot{\varphi}| \cos\varphi \sin 2\eta, \quad X_q = \tfrac{K_{21}}{m}|\dot{\varphi}| \sin\varphi \cos\varphi \sin 2\eta - x_h X_w$$

$$Z_u = 2X_w, \quad Z_w = -2\tfrac{K_{11}}{m}|\dot{\varphi}| \cos^2\eta, \quad Z_q = 2\tfrac{K_{21}}{m}|\dot{\varphi}| \sin\varphi \cos^2\eta - \tfrac{K_{rot_{12}}}{m}\dot{\varphi} \cos\varphi - x_h Z_w$$

$$M_u = 4\frac{K_{12}\Delta\hat{x}}{I_y}|\dot{\varphi}| \cos^2\varphi \sin\eta + \frac{m}{I_y}\left(2X_q - x_h Z_u\right)$$

$$M_w = 2\frac{K_{12}\Delta\hat{x}}{I_y}|\dot{\varphi}| \cos\varphi \cos\eta + 2\frac{K_{21}}{I_y}|\dot{\varphi}| \sin\varphi \cos^2\eta - \frac{m x_h}{I_y} Zw$$

$$\begin{aligned} M_q \ \ &= \ \ -\tfrac{2\Delta\hat{x}}{I_y}|\dot{\varphi}| \cos\varphi \cos\eta \left(K_{12}x_h + K_{22} \sin\varphi\right) + \tfrac{1}{I_y}\dot{\varphi} \cos\varphi \left(K_{rot_{13}}\Delta\hat{x} \cos\varphi \cos\eta + K_{rot_{22}} \sin\varphi\right) + \\ &- \ \ \tfrac{2}{I_y}|\dot{\varphi}| \cos^2\eta \sin\varphi \left(K_{21}x_h + K_{31} \sin\varphi\right) - \tfrac{K_v \mu_1 f}{I_y} \cos^2\varphi - \tfrac{m x_h}{I_y} Z_q \end{aligned}$$

where $K_v = \frac{\pi}{16}\rho I_{04}$ and $K_{rot_{mn}} = \pi\rho(\frac{1}{2} - \Delta\hat{x})I_{mn}$. The hinge line is positioned at $30\%c$ $(\Delta\hat{x} = 0.05)$, x_h is supposed to be zero for simplicity, and the value of C_{L_α} is obtained based on the wing aspect ratio using the extended lifting theory according to Taha et al. [29].

The above flight dynamic model has been matured in [30]. The eigenvalues of the obtained averaged, linearized dynamics have been validated against the experimental data of Cheng and Deng [9] and numerical simulations of Navier-Stokes equations by Sun et al. [26].

Chapter 3

Averaging

3.1 An Introduction to Averaging

There are two different timescales in dynamics of hovering birds. One is a fast time scale for variation of the aerodynamic loads like the flapping of the wings and the other is a slow scale for the overall movement of the body. The ratio between these two times scales is usually significant which justifies averaging. It is also worth mentioning that in the averaging approach there is no need to find a solution for the flight condition while hovering. Instead, one can obtain fixed points of the time-averaged system and use these fixed points in correspondence to create or in the initial model. There are mainly two methods to investigate stability analysis of flapping flight. In [33] Taha et al. have discussed these two approaches. In the first approach averaging over the period (T) is being used to on a nonlinear, time-periodic (NLTP) model to obtain the nonlinear time-invariant (NLTI), model. The averaging theorem guarantees the exponential stability of the corresponding periodic orbit of the NLTP model if a fixed point for the NLTI model is exponentially stable. To investigate the exponential stability of the fixed point NLTI model can be linearized to yield a linear time-invariant (LTI), model. To determine the stability of this system eigenvalues of the state matrix can be investigated [35, 36, 37, 27, 26, 39]. In the context of this thesis, this method is being referred to as Direct averaging.

The second approach used by Dietl and Garcia [10] and later by Weihua and Cesnik [25] finds the periodic solution of the hovering motion in the original NLTP model numerically. Next step is to linearize about the periodic orbit to find the linear time-periodic system. Floquet theorem can investigate the stability of such model. Using this method monodromy matrix (state transition matrix assessed at the fundamental period) can be achieved. Monodromy matrix is calculated by solving for the fundamental response of the LTP model over a single period. The content of this thesis is mostly concerned with hovering FWMAV which usually involves respectively high flapping frequency rather than forward flight.

3.2 Stability analysis of an FWMAV using direct averaging

The abstract form of system (2.2) can be written as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t), \tau) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{g}_a(\boldsymbol{x}(t), \tau), \tag{3.1}$$

Where, aerodynamic vector field $\boldsymbol{g}_a$ has explicit time dependence. As mentioned in the introduction chapter there exist two distinct timescales as slow timescale representing the body motion and the fast time scale representing with the accompanying flapping motion and aerodynamic loads. That is why two different symbols t and τ are used in Eq. (3.1). The ratio between these timescales varies among insects and birds but it is generally considered to be large which invokes averaging. For instance, this ratio for the hawkmoth which is known to be one of the slowest flapping insects is $30Hz$ [26, 30],. In this case, the body only counters with the mean value of the time-periodic aerodynamic load. Utilizing averaging, the average dynamics of the system can be written as

$$\dot{\bar{\boldsymbol{x}}} = \bar{\boldsymbol{F}}(\boldsymbol{x}(t)) = \boldsymbol{f}(\bar{\boldsymbol{x}}) + \bar{\boldsymbol{g}}_a(\bar{\boldsymbol{x}}), \tag{3.2}$$

in which over bar represent averaged quantity e.g., $\bar{\boldsymbol{g}}_a(\boldsymbol{x}) = \frac{1}{T}\int_0^T \boldsymbol{g}_a(\boldsymbol{x}, \tau)d\tau$, Where T is the flapping period. Based on [31, 30], an aerodynamic model was derived in [28, 29, 7],. Utilizing that model the aerodynamic loads can be expressed linearly in the state valuable x, close to the hovering position as

$$\boldsymbol{g}_a(\boldsymbol{x}(t), \tau) = \boldsymbol{g}_0(\tau) + [\boldsymbol{G}(\tau)]\, \boldsymbol{x}(t), \tag{3.3}$$

neglecting aerodynamic interactions of the body $\boldsymbol{g}_0$ represents the aerodynamic loads of the flapping motion of the wing and matrix $\boldsymbol{G}$ represents the aerodynamic derivatives (stability derivatives) with respect to the state valuables. In [31, 30, 33], Taha et al. investigated expressions of $\boldsymbol{g}_0$ and $\boldsymbol{G}$ terms in terms of the flapping kinematics.

Direct averaging can be used to clarify a time-periodic system (3.1) to a time-invariant system (3.2). As a result, a periodic orbit representing an equilibrium of (3.1) results to a fixed point of the averaged system (3.2). Due to this procedure stability analysis of a fixed point of a time-invariant system is significantly simpler than that of a periodic solution for a time-periodic model. One conclusion from [18, 22] is that using averaging theorem periodic solution of (3.1) is exponentially stable if the corresponding fixed point of (3.2) is exponentially stable. The subject of this thesis is investigating open loop stability of an FWMAV while it is hovering. Thus we will exclude the non-interval balance problem. Suggesting that we consider that the FWMAV is balanced in hovering position thus the averaged dynamics has a fixed point at the origin.

$$\boldsymbol{f}(\mathbf{0}) + \bar{\boldsymbol{g}}_0 = \mathbf{0} \Longleftrightarrow \bar{Z}_0 = -mg \tag{3.4}$$

Next step is to linearize the averaged dynamics (3.2) about this equilibrium

$$\dot{\bar{\boldsymbol{x}}}(t) = \left[D\boldsymbol{f}(\mathbf{0}) + \bar{\boldsymbol{G}}\right] \bar{\boldsymbol{x}}(t), \tag{3.5}$$

Where $D\boldsymbol{f}(\mathbf{0})$ represents the Jacobian of the vector field $\boldsymbol{f}$ at the origin and $\bar{\boldsymbol{G}}$ is the cycle-averaged stability derivatives. A simplified model of the averaged dynamics that is linearized about the hovering equilibrium can be written as

$$\begin{pmatrix} \dot{\overline{u}} \\ \dot{\overline{w}} \\ \dot{\overline{q}} \\ \dot{\overline{\theta}} \end{pmatrix} = \begin{bmatrix} \overline{X}_u & 0 & 0 & -g \\ 0 & \overline{Z}_w & 0 & 0 \\ \overline{M}_u & 0 & \overline{M}_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \overline{u} \\ \overline{w} \\ \overline{q} \\ \overline{\theta} \end{pmatrix}, \tag{3.6}$$

The linearized model was calculated for the hawkmoth insect, whose morphological parameters were obtained from [11, 26, 33] as

$$\left[D\boldsymbol{f}(\mathbf{0}) + \bar{\boldsymbol{G}}\right] = \begin{bmatrix} -3.59 & 0 & 0 & -9.81 \\ 0 & -3.30 & 0 & 0 \\ 39.95 & 0 & -7.92 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.7}$$

Eigenvalues of the above matrix are

$$0.19 \pm 5.74i, \ -11.89, \ -3.30.$$

These results suggest that the system is unstable as there exist eigenvalues whose real part is positive. These eigenvalues are mainly associated with pitching motion as can be proven by checking the corresponding eigenvectors.

3.3 Stability analysis of an FWMAV using Floquet theorem approach

An advantage of the Floquet theorem approach is that there is no need for a significant separation of time intervals for the flapping and the overall motion. On the other hand, it is necessary to obtain the periodic orbit. Application of this theorem necessitates finding fundamental matrix solution for a linear time-periodic system. In some cases first order (direct) averaging is not sufficiently accurate due to high amplitude periodic forcing. It also might be because timescales are not widely separated. In these cases using second order or even third order averaging is advised. To further investigate stability analysis of flapping bird, Taha et al. [33] applied Floquet theorem numerically and concluded that all Floquet multipliers were inside the unit circle which implies stability of the system. It is also worthed to mention that careless choices for the integration time steps may result in a wrong conclusion.

An alternative approach to study stability of flapping flight is to use higher-order averaging as this approach gives a better approximation for stability analysis of the model. This way higher-order interactions between the system's two time scales would be unveiled. An advantage of this approach is that unlike the Floquet theorem stability analysis can be implemented analytically and Thus numerical errors can be avoided.

3.4 Stability analysis of an FWMAV using higher order averaging

As discussed in the previous chapter direct averaging analysis cannot reveal vibrational stabilization. In this chapter, an alternative approach will be discussed.

Agrachev and Gamkrelidze drafted a paper [4] to lay down the foundation of a new calculus for time-varying vector fields; the *chronological calculus*. Later Sarychev [23] and Vela [38] based on provided tools in the *chronological calculus* developed what they called *complete averaging* to model a time-periodic system as an infinite series.

$$\dot{\overline{\boldsymbol{x}}} = \epsilon \boldsymbol{\Lambda}_1(\overline{\boldsymbol{x}}) + \epsilon^2 \boldsymbol{\Lambda}_2(\overline{\boldsymbol{x}}) + ..., \tag{3.8}$$

where ϵ is a small parameter, typically scaled with the reciprocal of the forcing frequency and

$$\begin{aligned} \boldsymbol{\Lambda}_1(\overline{\boldsymbol{x}}) &= \tfrac{1}{T}\int_0^T \boldsymbol{F}(\boldsymbol{x},\tau)\,d\tau, \\ \boldsymbol{\Lambda}_2(\overline{\boldsymbol{x}}) &= \tfrac{1}{2T}\int_0^T \left[\int_0^\tau \boldsymbol{F}(\boldsymbol{x},s)\,ds, \boldsymbol{F}(\boldsymbol{x},\tau)\right]\,d\tau. \end{aligned}$$

If the flapping frequency is high enough meaning that the ϵ is small enough the series (3.8) can be shortened (just keeping the first term $\boldsymbol{\Lambda}_1$). This is equivalent to direct averaging. For more details regarding this approach studying Refs. [23, 38, 33] is advised. In this section calculation of only two terms in the series (3.8) is investigated which is enough to give a better idea of the stability of the flapping flight. After linearization about the origin hawkmoth linearized hovering dynamics can be obtained as

$$D(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2)(\boldsymbol{0}) = \begin{bmatrix} -3.58 & 0 & 0 & -9.81 \\ 0 & -3.09 & 0 & 0 \\ 29.98 & 0 & -8.13 & -28.45 \\ -2.90 & 0 & 0.96 & 0 \end{bmatrix} \tag{3.9}$$

whose eigenvalues are

$$-0.66 \pm 3.72i, \ -10.40, \ -3.09$$

which represents a stable system. One interesting point to mention is that original hovering flight dynamics of insects lack any *direct* pitch-stiffness [30]. However, a significant value of the element (3,4) in the matrix (3.9) has been revealed which represents pitch stiffness (the essential feature of static stability of conventional aircraft [21]). In other words, this higher-order averaging analysis captured an indirect and unconventional spring action (pitch-stiffness) mechanism $\dot{\bar{q}} = \ddot{\bar{\theta}} = -k_\theta \bar{\theta}$. In the dynamics and control community, this phenomenon is being referred to as *Vibrational Stabilization/Control* (e.g., Baillieul and Lehman [5], Sarychev [23], and Bullo [8]) or stabilization via parametric excitation (e.g., Nayfeh and Mook[20]). This subject will be discussed more in the next chapter.

Chapter 4

Vibrational stabilization

4.1 Vibrational stabilization in inverted pendulum

In [8] Bullo discusses vibrational stabilization of mechanical systems. Here we review a section of his paper to introduce mathematics of vibrational stability. Consider the control system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g_a(x)u^a(t)}, \tag{4.1}$$

A critical point x_1 of f is said to be vibrationally stabilizable if for any $\delta > 0$ there exist almost-periodic zero-average inputs $u^a(t)$ such that the system in equation (4.1) has an asymptotically stable almost periodic solution $\boldsymbol{x^*(t)}$ characterized by

$$\|\overline{x}^* - x_1\| \leqslant \delta,\ \overline{x}^* = \lim_{T\to\infty} \frac{1}{T} \int_0^T x^*(s)\, ds \tag{4.2}$$

In 1908 Stephenson [24] investigated the dynamics of an inverted pendulum whose pivot has a vertical oscillation. The well known unstable equilibrium of the inverted pendulum gains quite a robust stability when the pivot is subjected to a sufficiently high frequency. Kapitza for the first time performed a full nonlinear analysis of this problem in [16, 17] and hence this problem is known as Kapitza pendulum. The linearized governing equation of the Kapitza pendulum can be written as:

$$\ddot{\theta} = \frac{g}{\ell}\theta + \frac{A\omega^2}{\ell}\theta \cos\omega t, \tag{4.3}$$

where θ is the pendulum angle with respect to the vertical position, ℓ is the length of the pendulum rod, A is the amplitude of the vertical oscillation of the pivot, and ω is the frequency of oscillation. Direct averaging of (4.3) yields in no effect of the vertical oscillation (the oscillatory term has zero mean) and shows that the inverted pendulum has an unstable dynamics; i.e., it cannot obtain any stabilization effects due to vibration or periodic forcing. However, If we employ higher-order averaging [23, 38] to the mathematical expression of the Kapitza Pendulum Eq. (4.3), we obtain (for large ω)

$$\ddot{\overline{\theta}} = \left(\frac{g}{\ell} - \frac{A\omega^2}{2\ell^2}\right)\overline{\theta}, \tag{4.4}$$

Above relation suggests that for high frequencies $\omega > \frac{\sqrt{2g\ell}}{A}$ a non-intuitive stabilization happens. The term $\frac{-A\omega^2}{2\ell^2}$ provides a stabilizing spring action ($\ddot{\bar{\theta}} = -k\bar{\theta}$) in which the spring constant $k = \frac{A\omega^2}{2\ell^2}$ increases with the oscillation frequency. This term is captured by higher-order averaging and represents the *net* effect of the vibration on the pendulum dynamics.

4.2 Proposed experimental setup to verify vibrational stability

Since the adopted tool in this study was analytical in nature (chronological calculus), it was possible to derive an analytical expression for the vibrationally-induced spring constant (stiffness) k_θ (matrix 3.9) for insect fight dynamics as

$$k_\theta = \frac{g}{2T}\int_0^T \left[M_u(t)t - \int_0^t M_u(\tau)d\tau\right] dt, \tag{4.5}$$

relation (5.1) suggests that k_θ is in correspondence to moment due to u. Thus, it was decided to propose an experimental setup that would allow study interactions between θ and u. Another purpose was to fabricate a setup that would allow for multiple equilibria. It was concluded that the best to fulfill these requirements is that a flapping setup (FWMAV) is attached to a pendulum. This way the movement of the pendulum at 90° can mimic the longitudinal motion of an insect near hover. Also, if the flapping setup is allowed to pitch this degree of freedom can play the role of θ

To verify the existence of vibrational stability in flapping setup we designed and fabricated an experimental setup. To fabricate a mimic of a flapping setup, we purchased its wing and a gear system to transform rotational motion of a DC motor to flapping motion [1]. We designed and fabricated a body and some accessories to be able to assemble a flapping setup. Same as the case with the inverted pendulum, in this experiment, we did not use any closed-loop control. As mentioned previously the objective of this experimental setup was to investigate vibrational stability in flapping setup while they are hovering. The proposed approach to tackle this issue was to compare the behavior of the flapping setup to the same set up while it is driven by a propeller. From now on we refer to these setups as the flapping setup and the propeller setup respectively. In particular, our study was focused on the stability of these two setups about their body pitching angle. The reason behind the existence of vibrational stability in flapping setup and not in the propeller-driven setup is the time-periodic aerodynamic force made by flapping birds as opposed to the constant aerodynamic force made by a propeller.

4.3 Measuring data of degrees of freedom

Both of these setups are mounted to a pendulum from a hinge point this way a free motion of a bird can be mimicked to some extend. An advantage of this approached is that there is no excessive or unnecessary translational or rotational degrees of freedom figure (4.1) shows the flapping setup which is mounted to a pendulum. We refer to this degree of freedom as Pendulum angle (γ). The other degree of freedom in this setup is the body pitching angle (θ_e) whose stability is the main focus of this study. By collecting data from these two degrees of freedom and plotting it with respect to time it is possible to investigate the performance of each setup and conclude whether there exists a vibrational stability mechanism or not. Note that the collection of data for both degrees of freedom should be in a real-time synchronized fashion. To capture pendulum angle we used a ”Gravity 360 Degree Hall Angle Sensor” (figure 4.2 [2]) which is a rotation sensor that works based on Hall effect. This sensor benefits from a 12 bit analog to digital converter which allows a resolution of 0.088° and accuracy of 0.3 percent full scale. Output of this sensor which is pendulum angle will be sent to an Arduino Uno controller and from there it will be transmitted to Arduino software. Right after data is received in Arduino software, it will be transferred to Visual studio software, where the other degree of freedom is being measured. This procedure happens very fast which allows visual studio environment to access and show pendulum angle in real-time.
To allow for body pitching angle θ, as shown in Fig. 4.1, a pin (hinge) connection is inserted between the body of the flapping setup and the pendulum's rod and for capturing body pitching angle, image processing has been utilized. A camera is responsible for taking a real-time video of the experiment. On the other hand, the body is marked by two different colored marks (at the nose and tail of the flapping setup) and the camera is responsible for detecting these two marks using color detection commands on the source code in the visual studio environment exploiting the image processing library OpenCV. These marks are put on the body in a way that the line connecting their centers would yield body orientation and

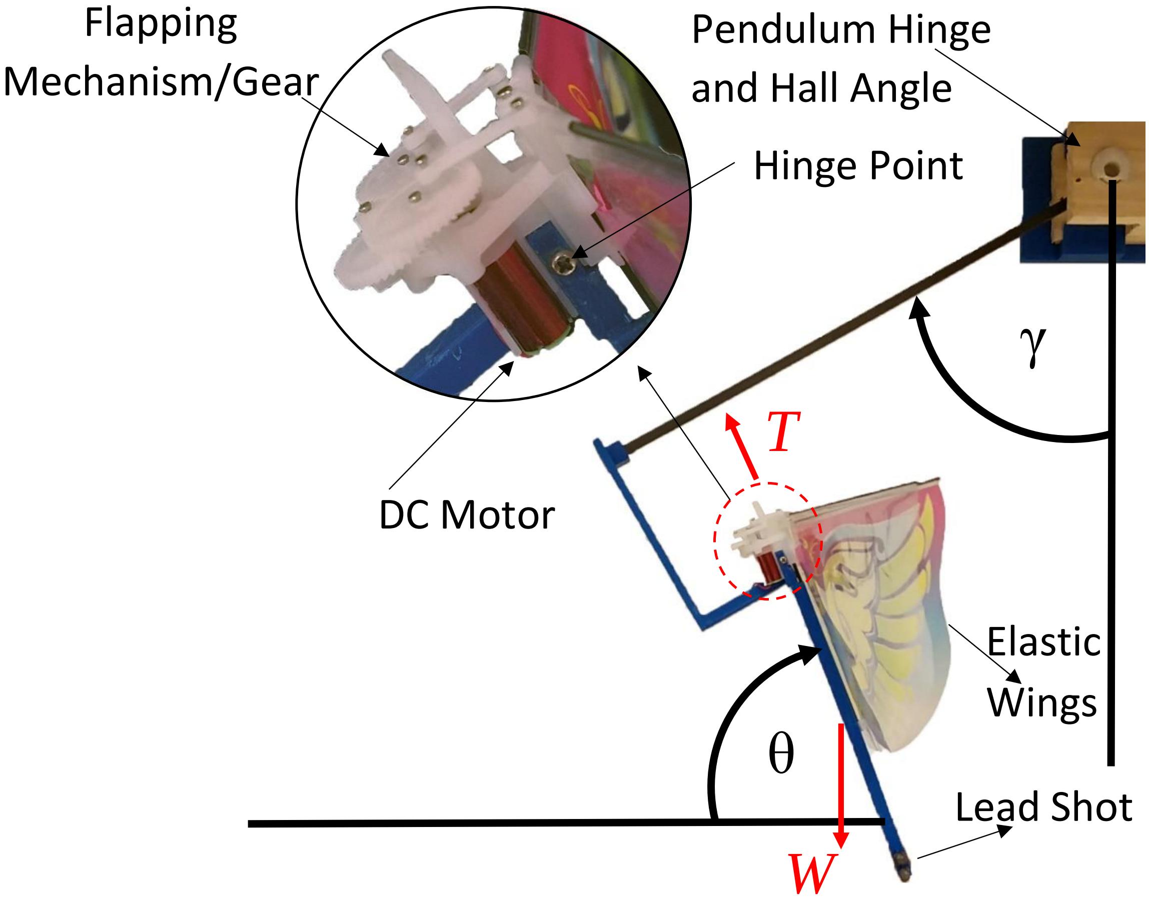

Figure 4.1: flapping setup attached to the pendulum

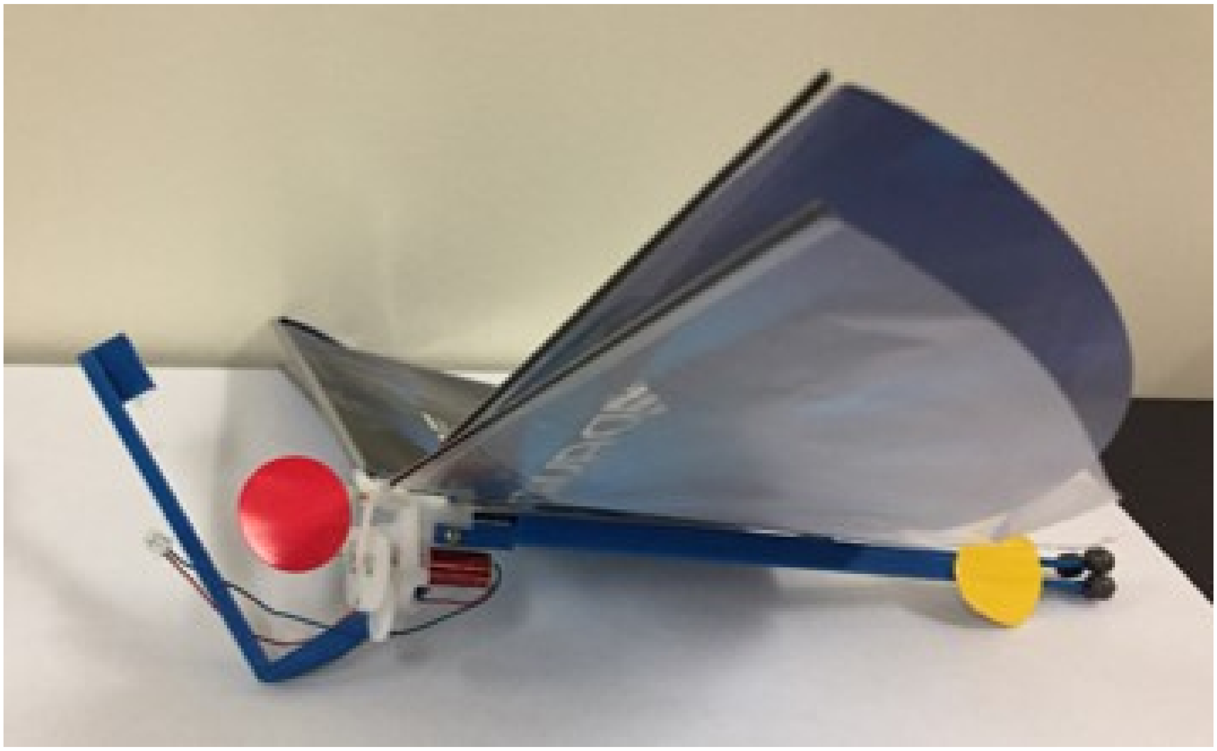

Figure 4.2: flapping setup

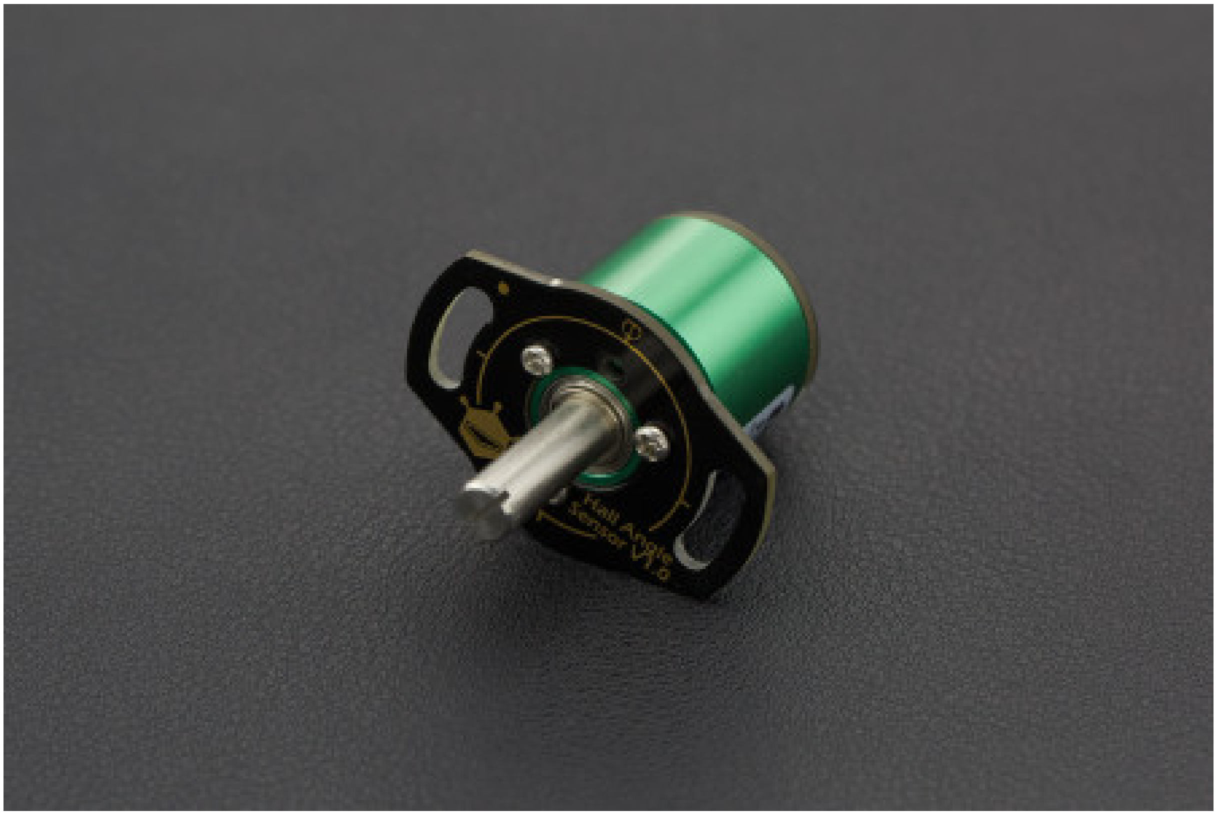

Figure 4.3: Gravity 360 Degree Hall Angle Sensor

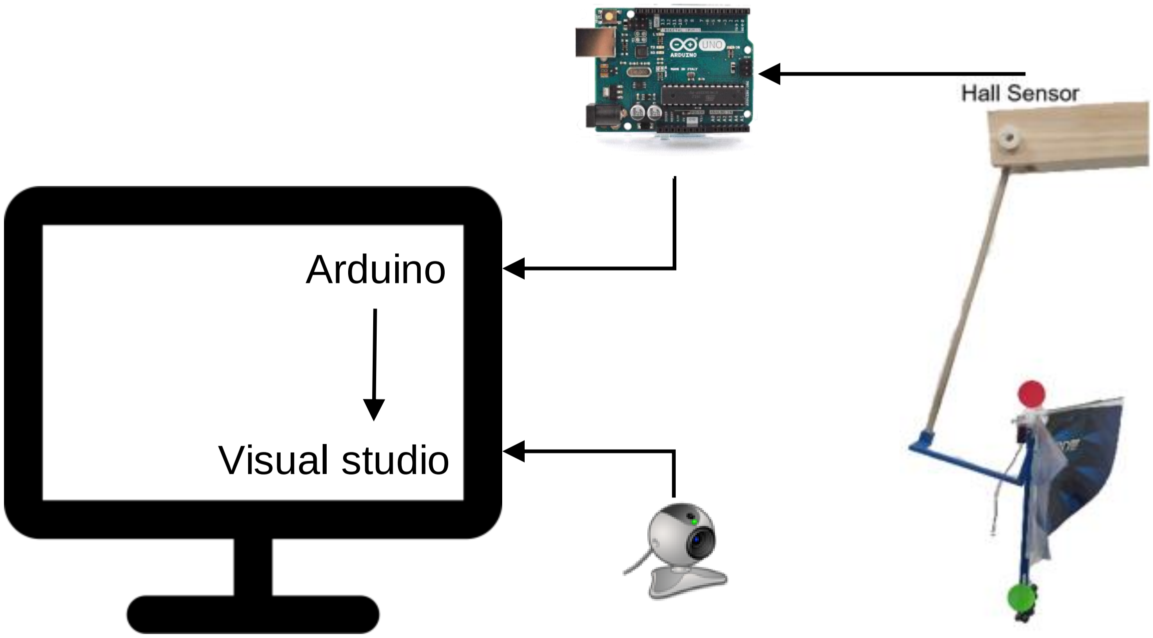

Figure 4.4: Flowchart of the sensor system

as a result body pitching angle can be captured. The written code is responsible to detect both marks and calculate their centers. The final step is to use tangent formulation to find the angle between the line connecting centers (which are aligned with the body) and the horizon. Sampling frequency here is dependent on the processing capabilities of the CPU. In the charts provided in next sections, the sampling frequency is less than 50 ms. Using this procedure both degrees of freedom will be accessible in the visual studio environments. In a loop on the source code, a clock is utilized to print the time that the experiment is running constantly. This way in every cycle the values of pendulum angle and body pitching angle will be tagged in time. This procedure will allow us to later plot the charts of these two degrees of freedom with respect to time. A flowchart of this procedure can be seen in figure (4.4)

4.4 Experimental Demonstration

Fig. 4.1 shows a simple pendulum with its mass replaced by a flapping setup, as The hovering equilibrium position, in this case, is when the pendulum's rod becomes horizontal (γ=90).

One important advantage of this setup is that it will allow for multiple equilibria at different frequencies; convenient to investigate vibrational control, which is evident only at high enough frequencies. This way operating at a slower flapping frequency is possible, which results in a different equilibrium *position* γ_e. Measurement of the pendulum angle provides a way for the generated thrust from the flapping setup as the flapping frequency changes, according to the balance equation

$$F_T = \left(m_{\text{FWMAV}} + \frac{1}{2} m_{\text{rod}} \right) g \sin \gamma_e,$$

where F_T is the cycle-averaged generated thrust force, m_{FWMAV} is the mass of the flapping setup (13 gm), m_{rod} is the mass of the pendulum's rod (1.8 gm), and g is the gravitational acceleration. The flapping mechanism of the bird is actuated by a DC motor and as the applied voltage is increased, the flapping frequency increases and the flapping setup rises up (i.e., γ_e increases). To obtain the flapping frequency of the bird, a time stamp of a recorded video (from a separate camera) at a rate of 240 frames per second is analyzed.

The line of action of the thrust force is above the body longitudinal axis and consequently hinge point. Thus there is an unbalanced pitching moment which will preclude equilibria. Therefore, we added four split shot size lead of 3g total weight near the tail of the flapping setup, as shown in Fig. 4.1 (the black dots near the tail) to shift the center of gravity of the flapping setup backward along the longitudinal axis. This way, the pitching moment at the hinge point due to the weight will balance that of the thrust force according to the balance

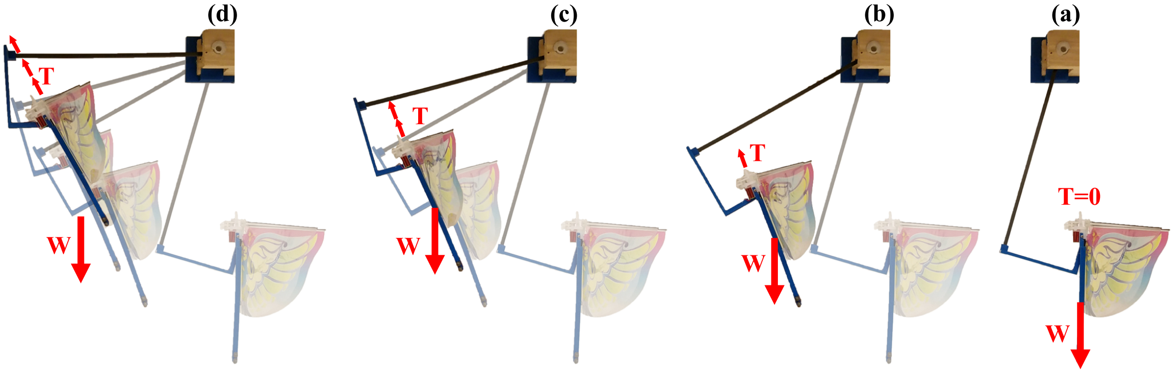

Figure 4.5: Robotic flapper motion from (a) zero voltage (no flapping) to (d) high voltage (fast flapping) hovering equilibrium. At zero applied voltage (no flapping), the body is standing vertically ($\theta \simeq 90°$) at the bottom position of the pendulum. As the voltage and consequently the flapping frequency increases, the body moves upward along the circular path of the pendulum (i.e., γ increases) and tilts forward towards the horizontal attitude (i.e., θ decreases).

equation

$$F_T e_T = mge_g \cos\theta_e,$$

where e_T and e_g are the offsets of the thrust and gravity forces, respectively, from the hinge point, and θ_e is the equilibrium value of the pitching angle. Obviously, with no applied voltage (zero thrust force), the flapping setup is at the bottom position of the pendulum ($\gamma_e = 0°$) standing vertically($\theta_e = 90°$). The body moves upward along the circular path of the pendulum (i.e., γ increases) and tilts forward towards the horizontal attitude (i.e., θ decreases), as shown in Fig. 4.1 as the voltage and consequently the flapping frequency increases. Figure (4.5) shows multiple equilibria states for the robotic flapper corresponding to different flapping frequencies (voltages). An interesting point worthed to be mentioned here is that most insects have their center of gravity behind the hinge location along their longitudinal axis. As a result of this phenomenon they achieve hovering equilibria at body inclination with respect to the horizontal (i.e., θ_e) around 50° [11, 26]; i.e., similar to the current setup. Having equilibrium established, studying vibrational stability comes promptly. Direct averaging and higher-order averaging were discussed theoretically in previous chapters. To verify and demonstrate the vibrational control/stabilization phenomenon in flapping setup experimentally, we apply different voltages to the DC motor driving the flapping mechanism to achieve different equilibrium positions (γ_e and θ_e) at different flapping frequencies, thanks to the pendulum configuration. The response of the pendulum angle γ and the body pitching angle θ, is then attainable as explained above, at each input voltage/operating frequency.

Figure (4.6) shows the response of the flapping setup at a flapping frequency of 12Hz. At this low flapping frequency, the flapping setup barely goes up ($\gamma_e \sim 24°$), and the equilibrium pitching angle is quite large ($\theta_e \sim 76°$). At this low flapping frequency, the response is inherently unstable, even without any external disturbance; the oscillatory wing motion naturally provides a sufficient disturbance. Figure (4.7) shows the response of the flapping setup as the applied voltage (flapping frequency) is manually increased from 1V to 3V. The bird rises towards the hovering position (γ goes from 20° to 60° and θ changes from 77° to 62°). In the beginning, the flapping setup's response experiences instability during the transition period and then gains stability beyond a certain pendulum angle (i.e., flapping

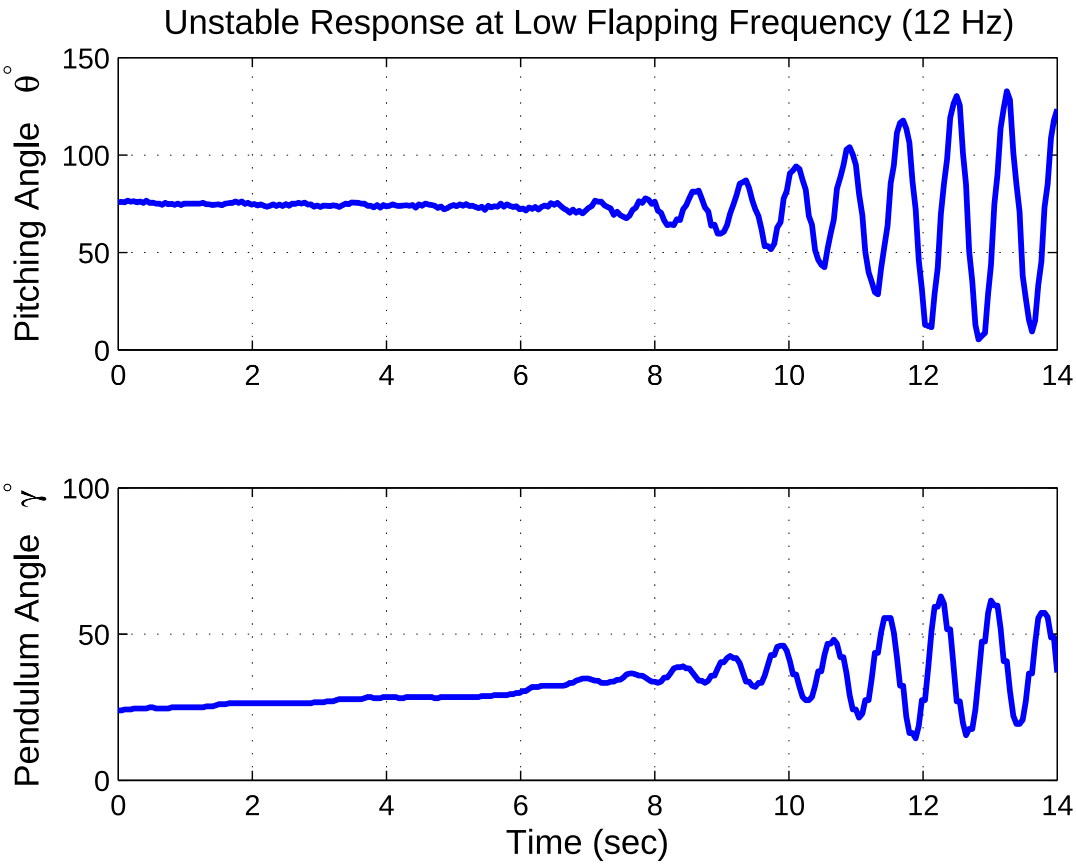

Figure 4.6: flapping setup's unstable response at relatively low flapping frequency (12Hz).

frequency). We apply fixed different voltages (corresponding to different flapping frequencies) and observe the system response in each case. The threshold flapping frequency below which the flapping setup's response is unstable and beyond which it is naturally stabilized is found to be 15Hz.

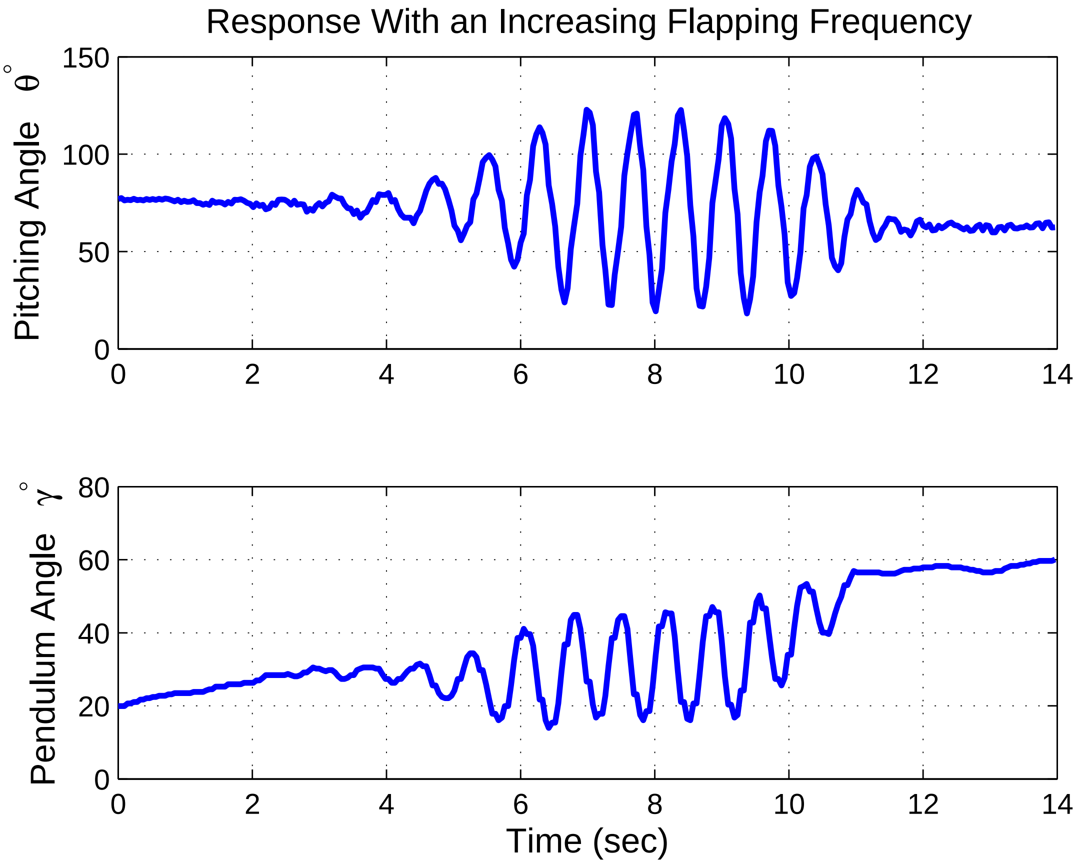

Figure 4.7: flapping setup's response as the flapping frequency is being manually increased.

The response of the flapping setup at a relatively high flapping frequency of 18Hz is plotted in figure (4.8). The flapping setup is almost at the hovering position ($\gamma_e \sim 85°$) at this relatively high flapping frequency, and the equilibrium pitching angle $\theta_e \sim 50°$ is close to the natural values observed in nature for hovering insects [11, 26], the response has gained stability at this frequency. Even when a relatively large disturbance ($\Delta\theta \sim 50°$) is applied at $t = 8.6$ sec, the system regain its equilibrium periodic orbit (i.e., the hovering periodic orbit).

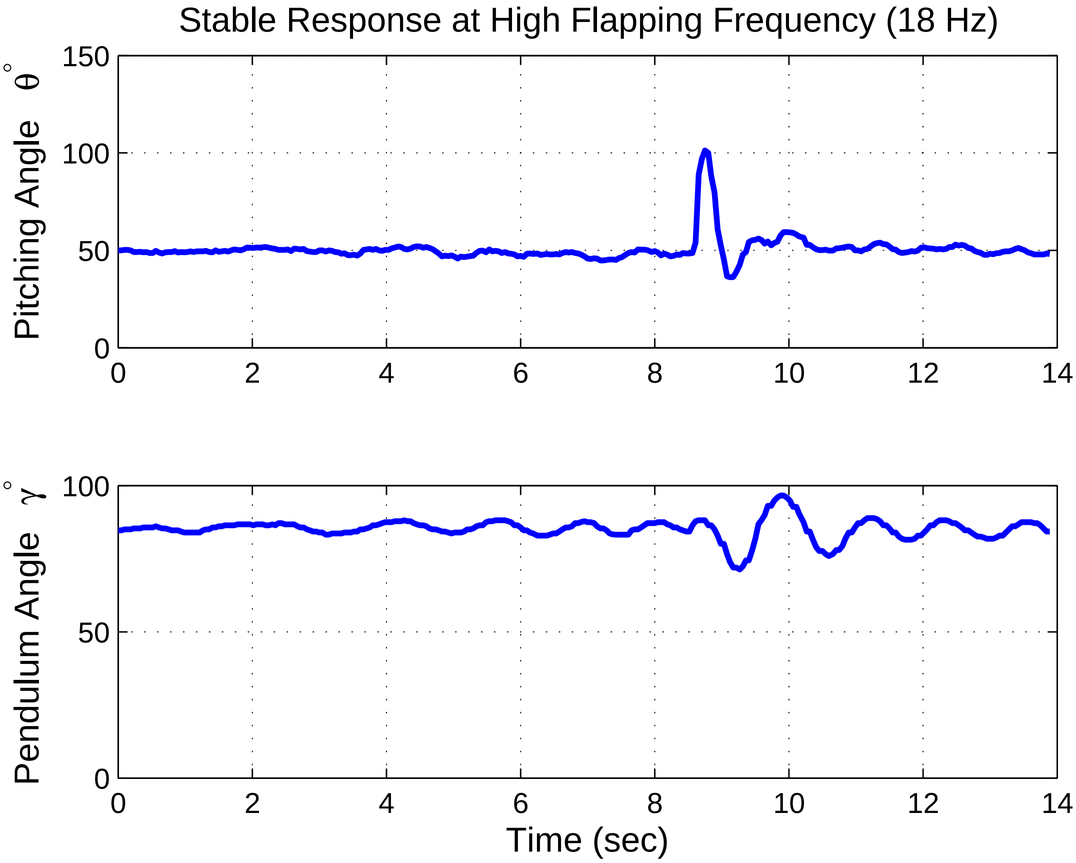

Figure 4.8: flapping setup's stable response at relatively high flapping frequency (18Hz).

So far, it can be seen that the response of flapping setup (particularly the body pitch response) is naturally (without feedback) stabilized beyond a certain threshold of flapping frequency. This observation conforms the vibrational control concept [16, 17, 24, 6] and suggests that the observed natural stabilization at high forcing frequencies is a vibrational stabilization phenomenon. However, one might argue that because the intricate dynamics of the system, the frequency not only affects stability but also balance/equilibrium; obviously increasing the frequency leads to a different equilibrium, which may or may not have similar stability characteristics to equilibria corresponding to low frequencies. Addressing this concern is the motivation behind constructing a replica of the experimental setup with the flapping mechanism being replaced by a small propeller revolving with a constant speed, as shown in Fig. 4.9 (we refer to this setup as the propeller-pendulum system). This way it is possible to prove that the induced stabilizing mechanism is indeed due to vibrational control that mainly stems from the time-periodic nature of the driving aerodynamic thrust force and not because of operating at a different equilibrium. The main difference is that the flapping setup produces a periodic thrust force, and consequently, a time-periodic dynamics allowing for vibrational control, while the propeller setup produces a constant thrust force, and consequently a time-invariant dynamics does not conclude to vibrational control.

Using split shot size lead, we managed to match the weight and inertia of the propeller system with the flapping setup. Figure (4.10) shows the response of the two-DOF propeller-pendulum setup (pendulum setup) at a relatively small propeller speed (i.e., at a small pendulum equilibrium angle $\gamma_e \sim 9°$). The response is exponentially unstable. Increasing the applied voltage to attain higher pendulum equilibrium angles (closer to the hovering position) will only worsen the stability characteristics so much that the system structure becomes prone to breaking.

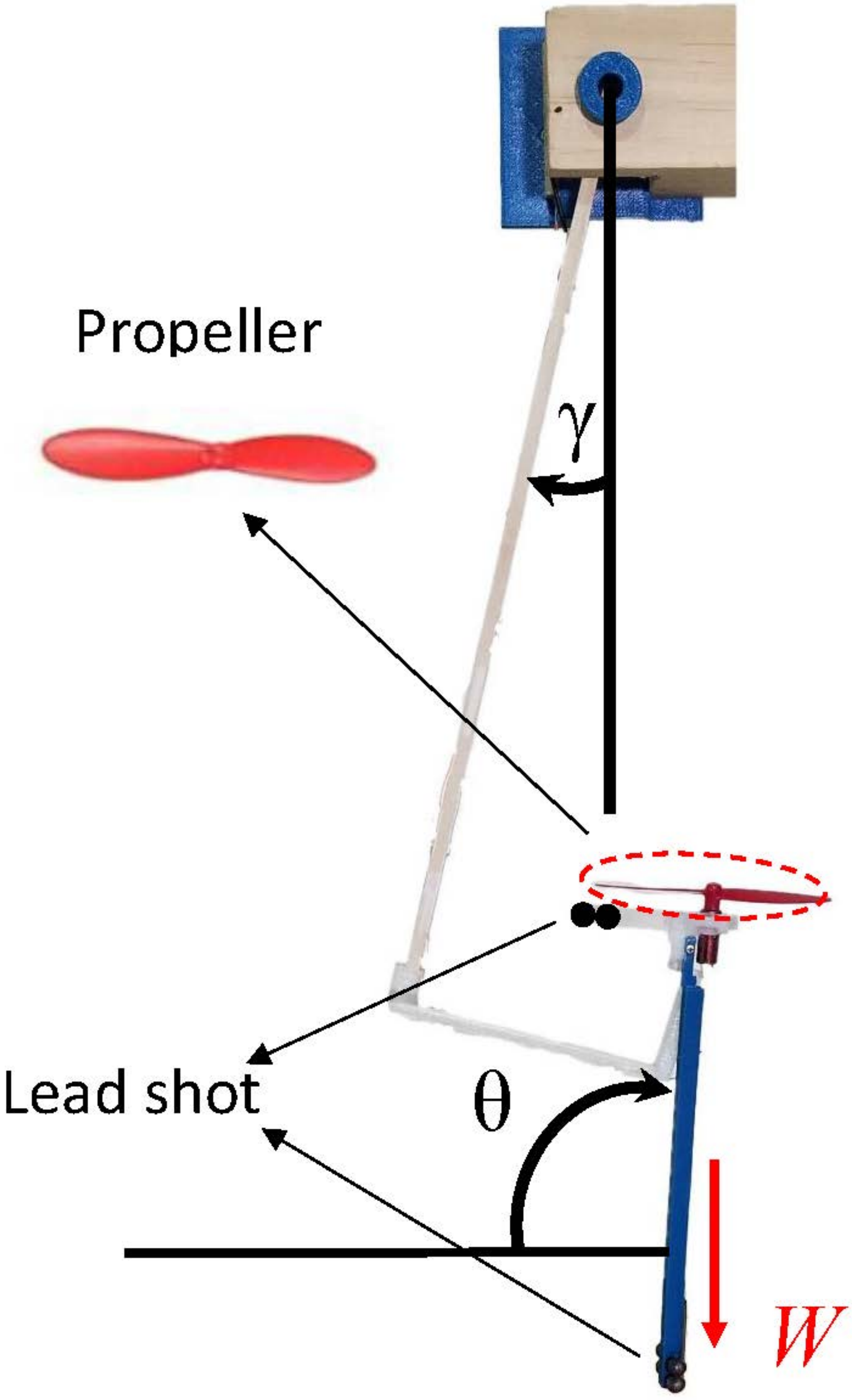

Figure 4.9: Propeller pendulum setup

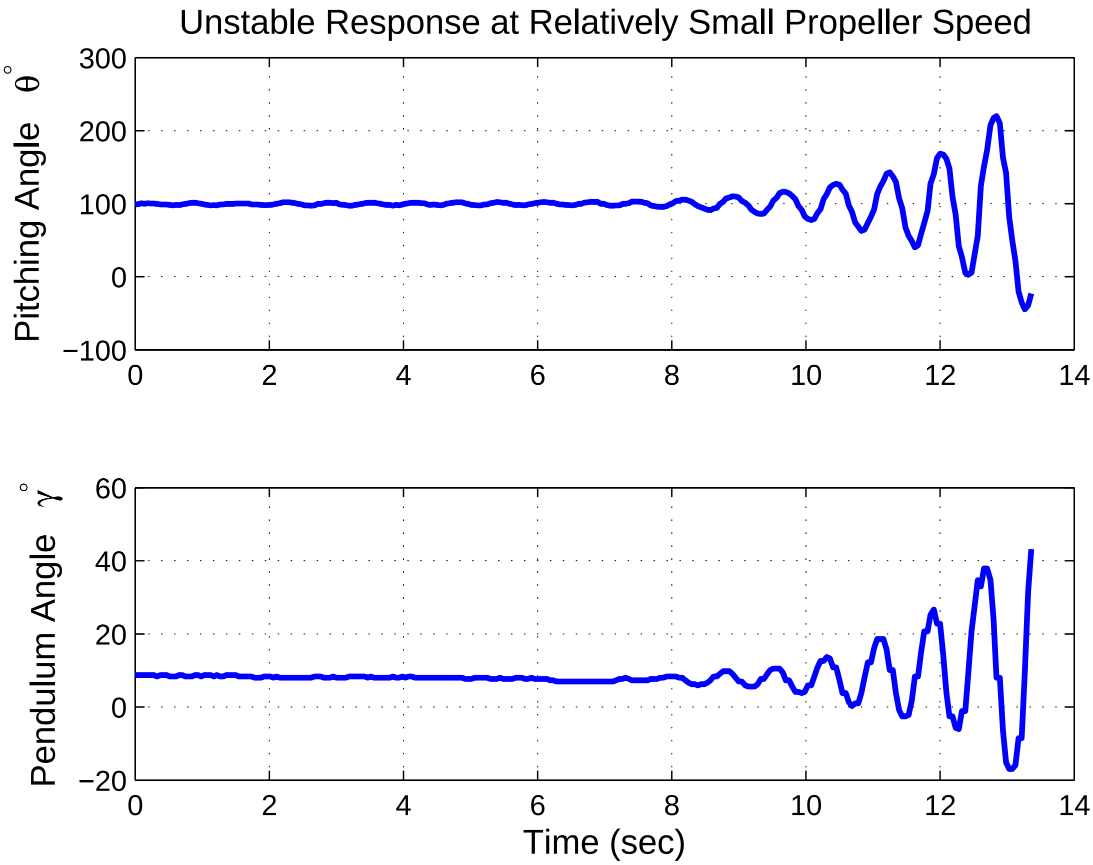

Figure 4.10: Unstable Response of the Two-DOF Pendulum setup.

4.5 Break down of the setup's component

Figure (4.13) shows an assembly experimental setup made to investigate vibrational stability. Main components of this setup are

1−A Test stand; shown as a brown wooden stand in figure (4.12). 2−A Pendulum; shown as a black stick in figure (4.14). 3−An arm responsible for connecting pendulum to the flapping setup; shown as a black stick in figure (4.14). 4−A Flapping wing micro air vehicle; shown in figure (4.11).

A test stand is needed to provide a basis for the motion of both degrees of freedom. The pendulum can be mounted to this stand allowing for a circular trajectory for the flapping setup as shown in figure (4.5)

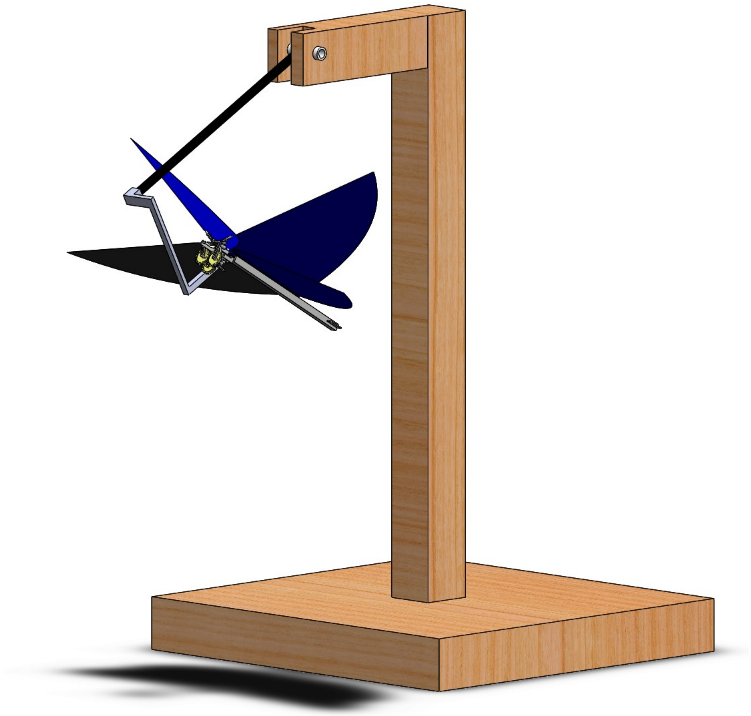

Figure 4.11: Assembly of the experimental setup

3 view and isometric drawing of this test stand can be seen in figure (4.12).

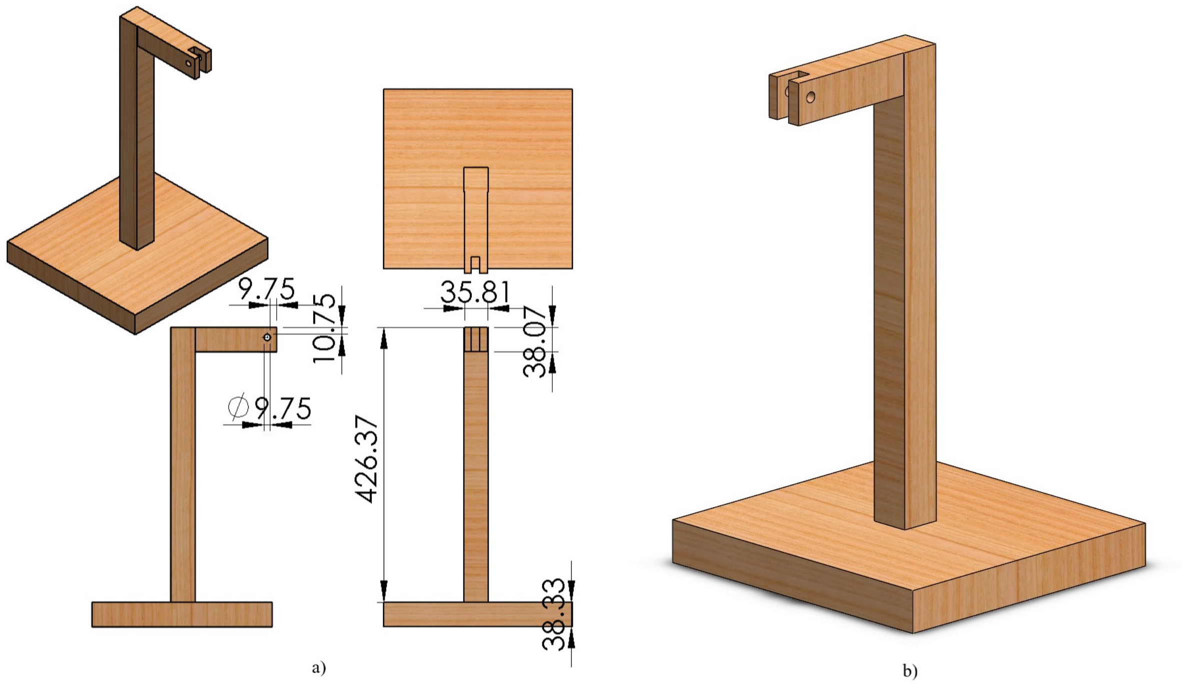

Figure 4.12: (a) 3 view drawing of the test stand. (b) Isometric drawing of the test stand.

To mount the pendulum to test stand and the hall sensor altogether, a mount figure (4.13) was made. This mount is designed and 3d printed to transfer the movement of the pendulum to that of the hall sensor shaft (4.3). This way, pendulum angle can be collected and sent to Arduino. All 3d printed parts are made of ABS-plastic.

If the flapping setup is attached directly to the pendulum, motion of degrees of freedom will interfere each other. That is because the orientation of the setup cannot allow the pendulum to be connected to the flapping setup from its backside. That is why an L-shaped arm is designed and 3d printed (figure 4.15). This piece will be referred to as arm from now on. This has a square hole to fit in the pendulum. Also, on the other end of it there a circular hole is implemented. Another circular hole is embedded in the structure of the flapping setup. These two holes are supposed to be aligned and connected using a pin or a screw. The screw that was selected for this matter has a diameter of 1.3mm. It is noteworthy to

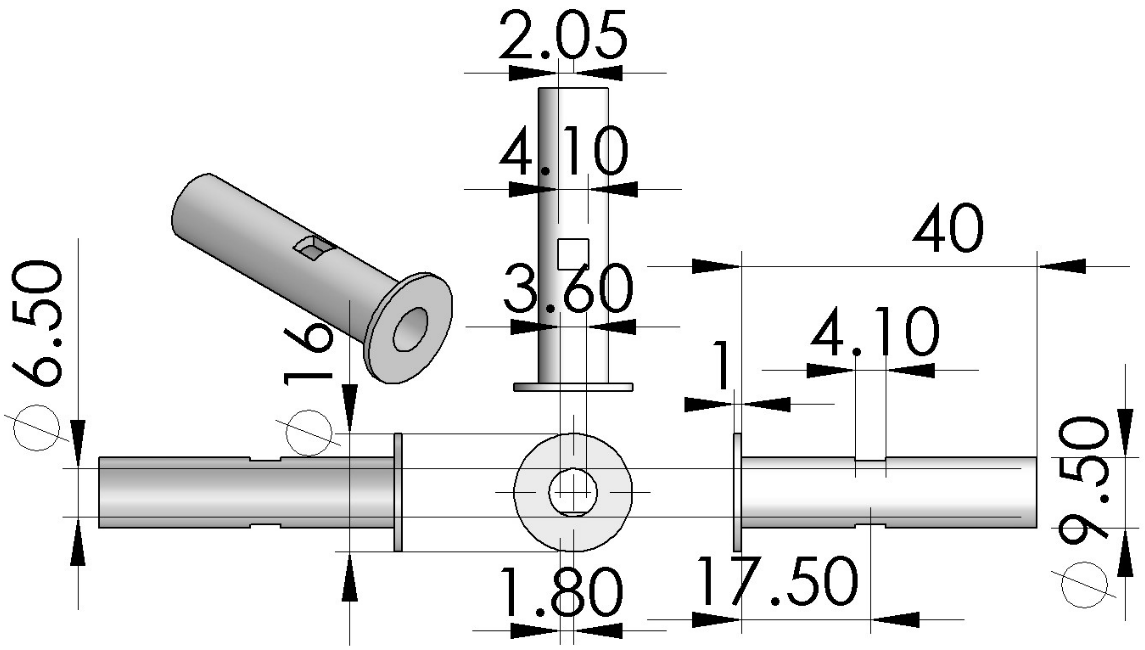

Figure 4.13: Hall sensor mount

mention here that since there is a tolerance in the procedure of 3d printing, dimensions of parts may need to be altered depending on the 3d printer accuracy and type of filament used for 3d printing.

A body was designed and 3d printed to withhold the wings and gear system figure (4.16). It is necessary to design this part in a way that the body is allowed to pitch freely with minimum friction. The performance of this setup near hover was one of the main concerns and played a major role in designing this part. The DC motor used to drive the flapping mechanism can be found in [3]. Figure (4.18) shows attachment of the main parts in the flapping setup. The wings and gear system is omitted in this figure as the purpose is to show the orientation and position of each part.

As mentioned previously, a replica of the flapping setup is constructed with the flapping mechanism being replaced by a small propeller revolving with a constant speed. Alignment of the body and arm in the propeller setup is shown in figure (4.19). Note that the orientation

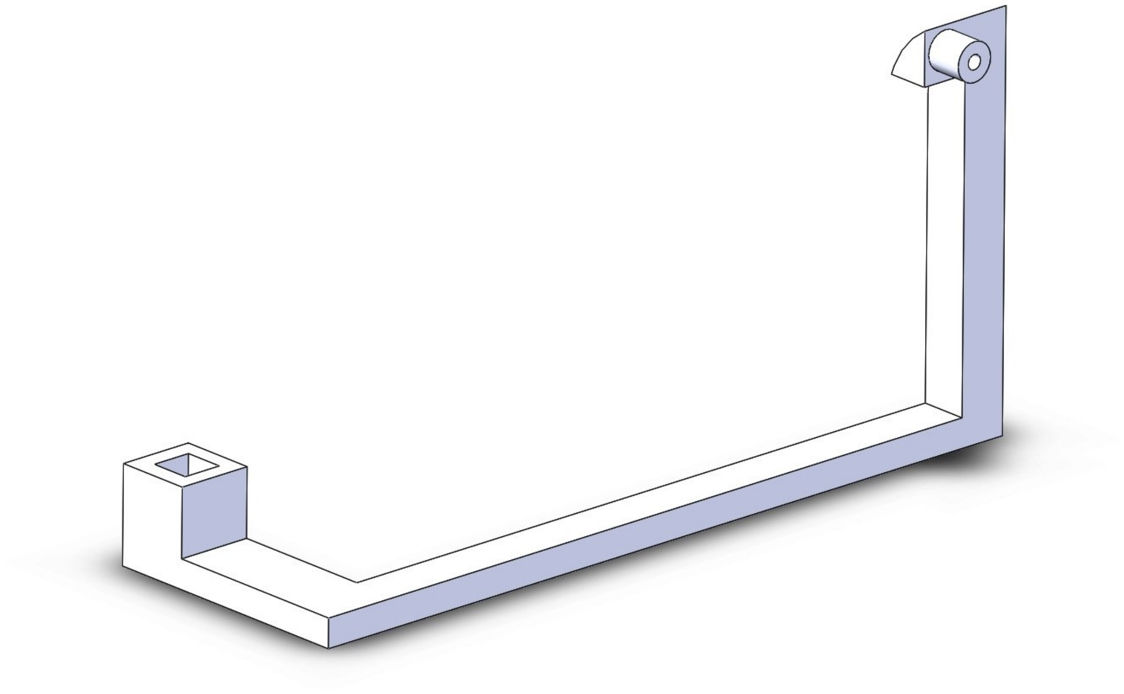

Figure 4.14: Isometric view of the L-shaped arm used to connect flapping setup to the pendulum

between body and arm is different in this setup than that of the flapping setup. Also, the arm is modified to not limit motion of the setup in this new orientation (see figure 4.20). Reason for this orientation is that the sequence of the action line of trust, hinge point and center of the gravity should be the same for both setups. Inertia and the position of CG are matched in both setups by adding lead shots in different locations on the body.

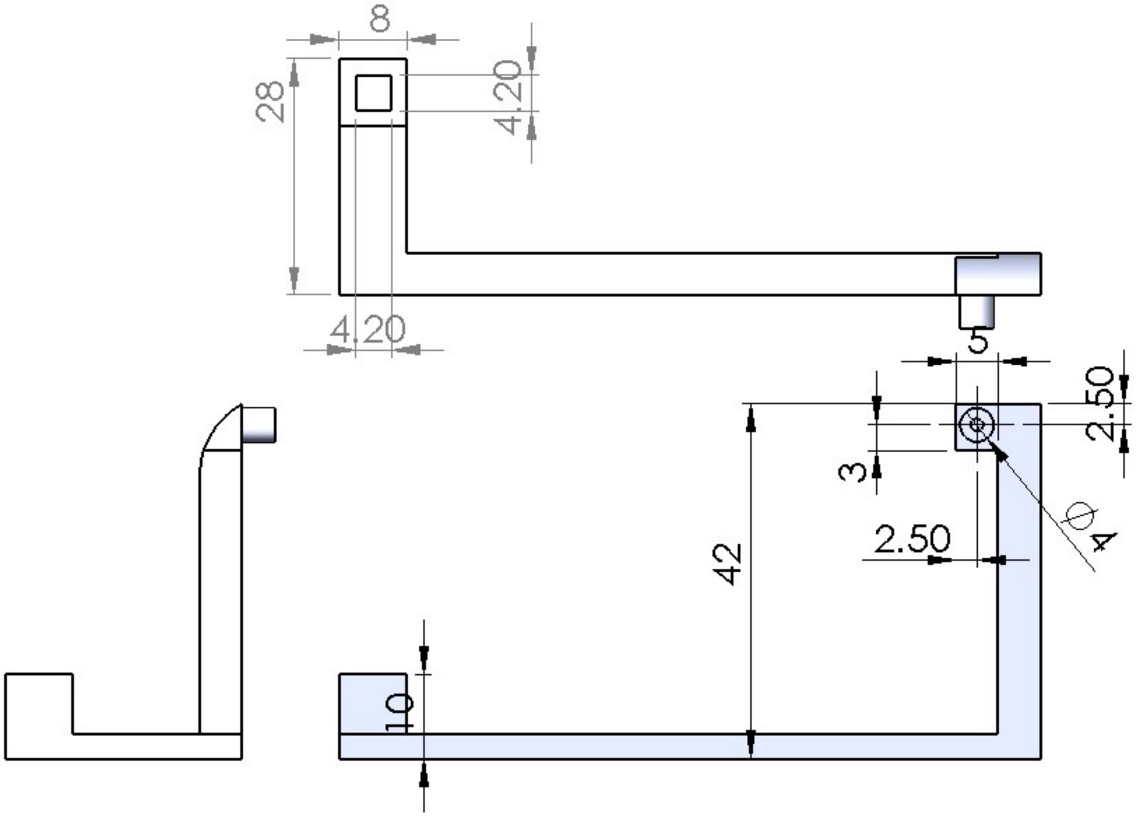

Figure 4.15: 3 view drawing of the L-shaped arm used to connect flapping setup to pendulum

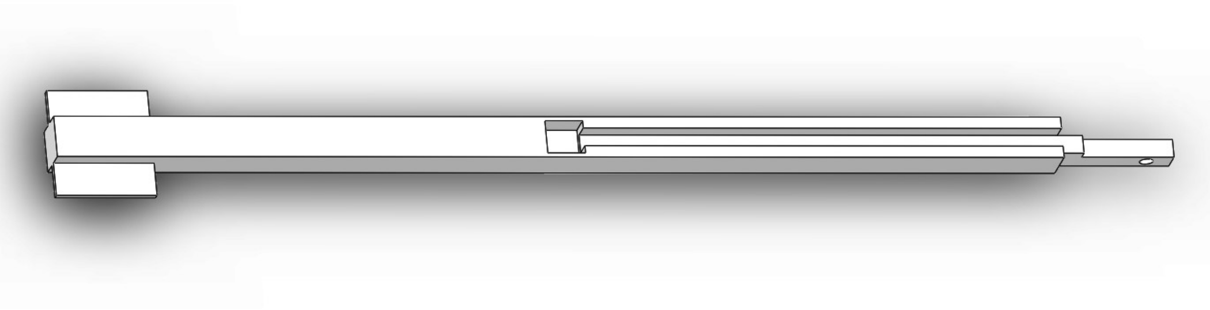

Figure 4.16: Isometric view of the fabricated body of the flapping setup

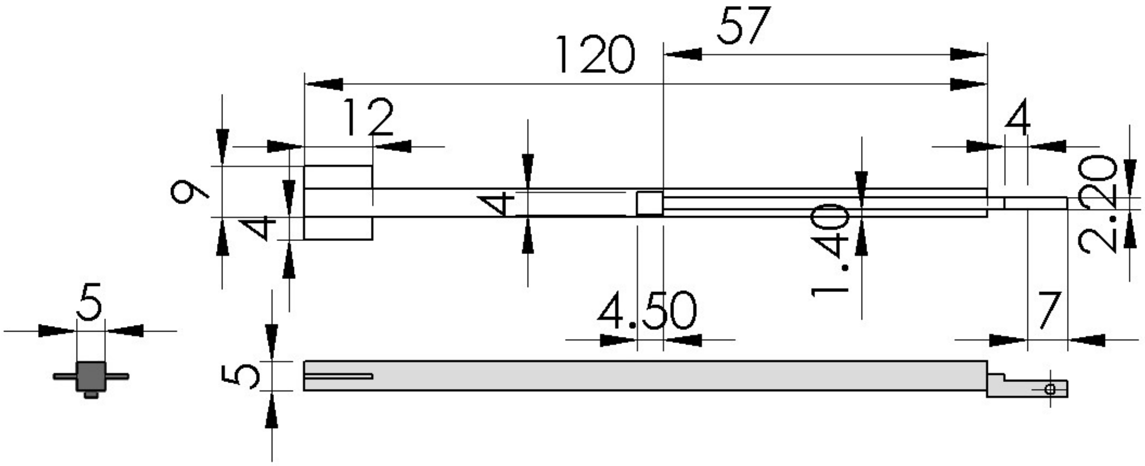

Figure 4.17: 3 view drawing of the fabricated body of the flapping setup

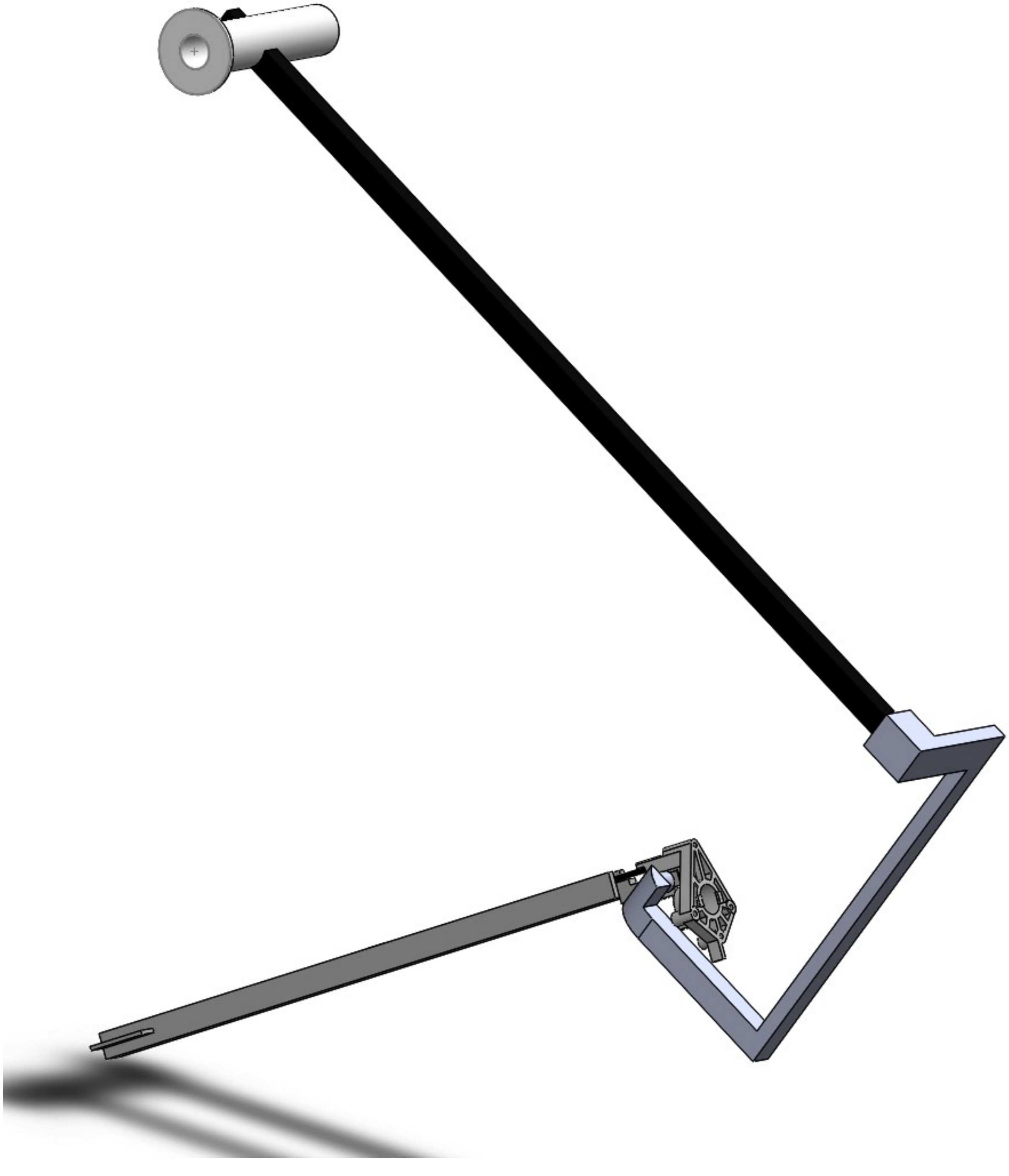

Figure 4.18: Simplified schematic of the setup from the hall sensor mount to the body of flapping setup

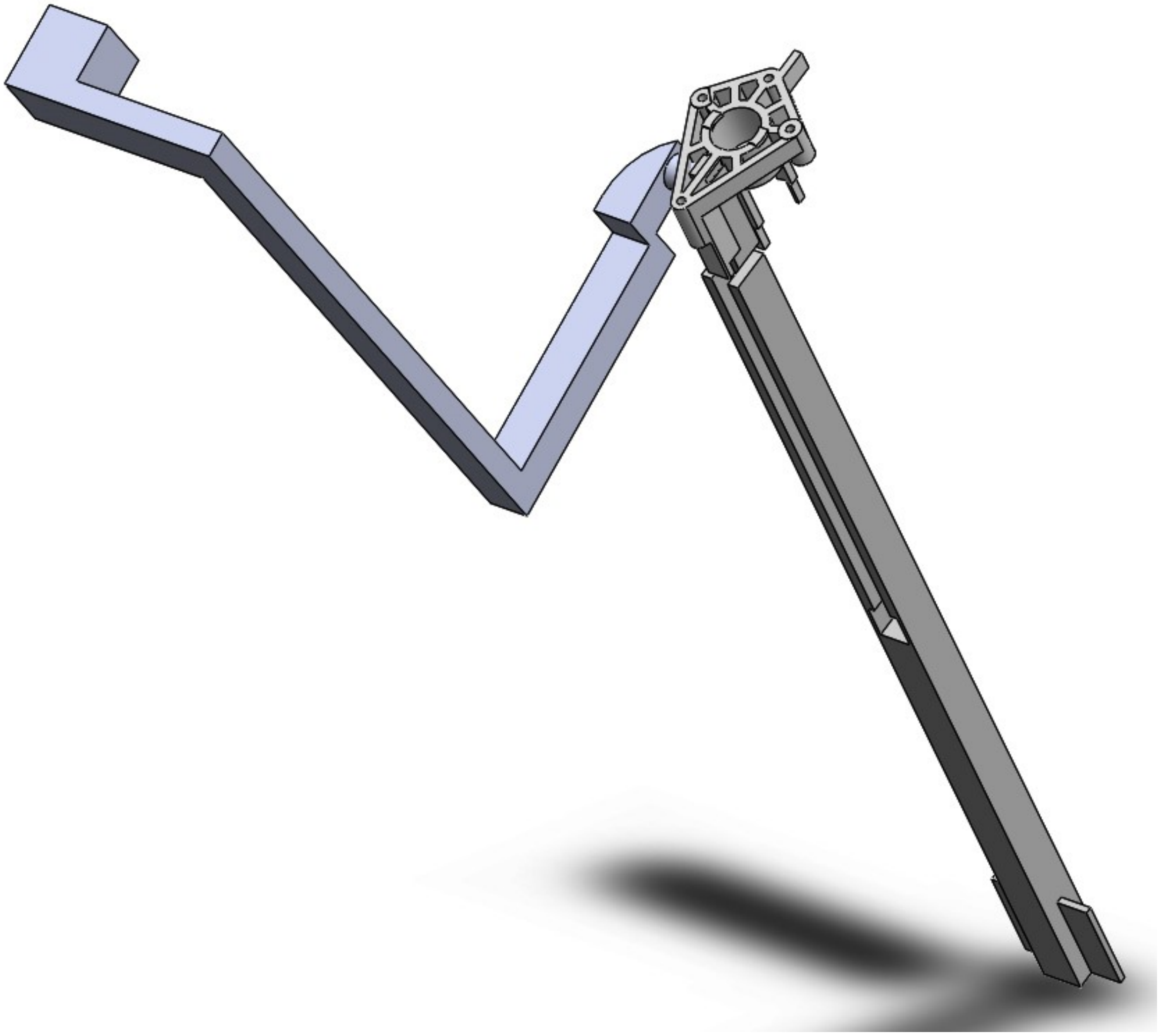

Figure 4.19: Propeller setup attached to the arm in a new orientation

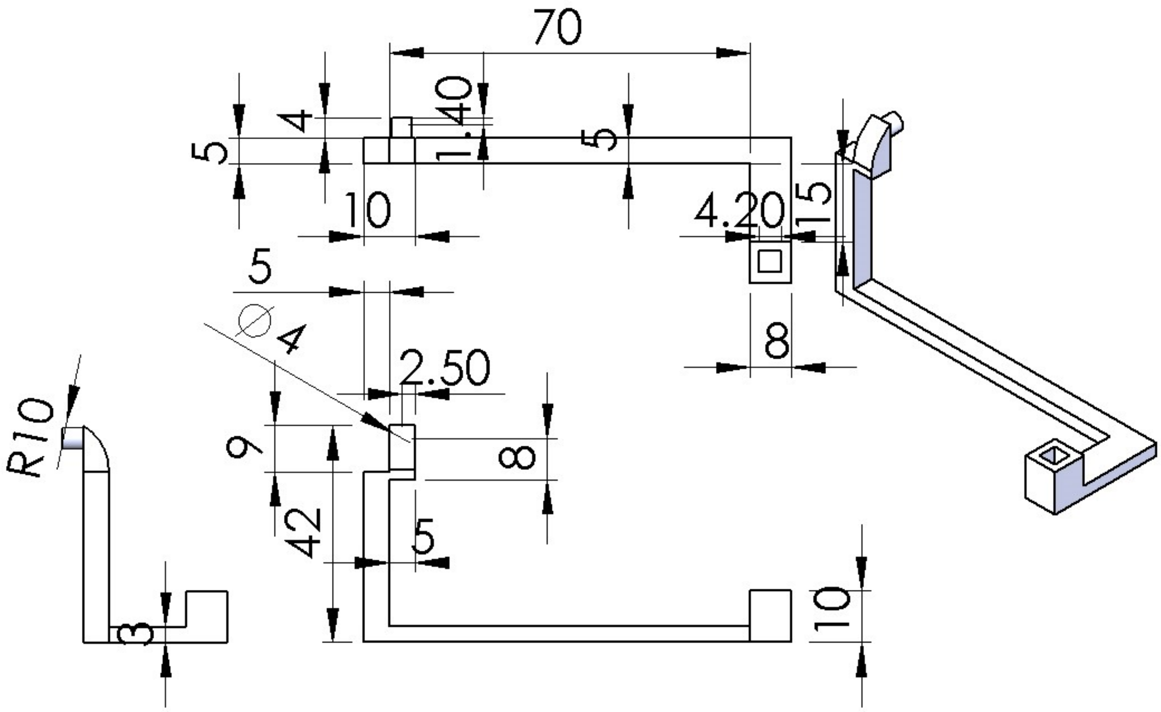

Figure 4.20: 3 view drawing of the arm used in the propeller setup

Chapter 5

Physics of Vibrational Stabilization in Insect Flight Dynamics

Here we refer to the previously discussed analytical expression for the vibrationally-induced spring constant (stiffness) k_θ (matrix 3.9)

$$k_\theta = \frac{g}{2T} \int_0^T \left[M_u(t)t - \int_0^t M_u(\tau) d\tau \right] dt, \tag{5.1}$$

where T is the flapping period. Equation (5.1) can be used to describe the nature of the vibrationally-induced stiffness; k_θ merely is related to gravity and the variation of M_u over the flapping cycle, implying that the interaction between the slow body dynamics and the fast wing flapping dynamics is what makes the vibrationally-induced stiffness. When the body experiences a pitch disturbance $\Delta\theta$ (e.g. pitch up), gravity comes in play and pulls it back along the x-axis as the first equation in the slow body dynamics suggests (2.1 or 3.6). The effect of this disturbance on the forward speed u diminishes on the slow time-scale corresponding to the body dynamics due to the damping action X_u. This damping contribution by X_u is, however, time-varying on the fast time-scale. As any typical aerodynamic load, this aerodynamic damping depends quadratically on the wing speed. The damping coefficient X_u has a similar sinusoidal pattern with a non-zero mean value [9, 30]. Because the flapping speed varies along the flapping cycle. The response of the translational speed of the body is affected by this oscillation of the damping action. Therefore, the body (on the average) is recovering from the initial u-disturbance while having an oscillatory u component. In other words, the u-response can be decomposed as $u = \overline{u} + \Delta u$, where $\overline{u}$ is the cycle-averaged speed determined by the averaged dynamics (e.g., 3.6) and Δu is a zero-mean oscillatory component that is often neglected by direct averaging.

On the other hand, having in mind that M_u is the aerodynamic pitching moment due to a disturbance in the body forward speed u, it has a positive mean value $\overline{M}_u$ [9, 30] that facilitates a restoring (negative) pitching down moment due to the body backward (negative) disturbance $\overline{u}$. That means the restoring pitching moment $\overline{M}_u\overline{u}$ is the contribution obtained by direct averaging. However, an additional zero-mean, oscillatory pitching moment contribution ΔM_u is often similarly ignored. With the flapping of the wing, the center of pressure moves back and forth accordingly, changing the moment arm of the aerodynamic loads concerning the body center of gravity (CG) as a consequence, as shown in figure (5.1). The interesting point is that the oscillatory body speed Δu is synchronized with the oscillatory moment arm due to a pitch disturbance, as shown in the figure (5.1). This harmony makes the resulting pitching moment$\Delta M = \Delta M_u \Delta u$ an always negative value (i.e. pitching down), as shown in figure (5.1). As such, its net effect is a restoring pitching moment proportional to the pitch disturbance; i.e., a spring action. In other words, it is precisely the fact that M_u has a fast-varying component ΔM_u stemming from the fast wing flapping oscillation which leads to the stiffness k_θ when the body speed u is disturbed upon pitch disturbance. This can be seen from Eq. (5.1) which results in zero k_θ if M_u is constant (i.e., replaced with its averaged value $\overline{M}_u$). The other point that can be extracted from figure (5.1) is that if the moment arm always has the same sign, the resulting pitching moment due to the oscillatory Δu would not always have the same sign (have a zero mean value). Similarly, if the

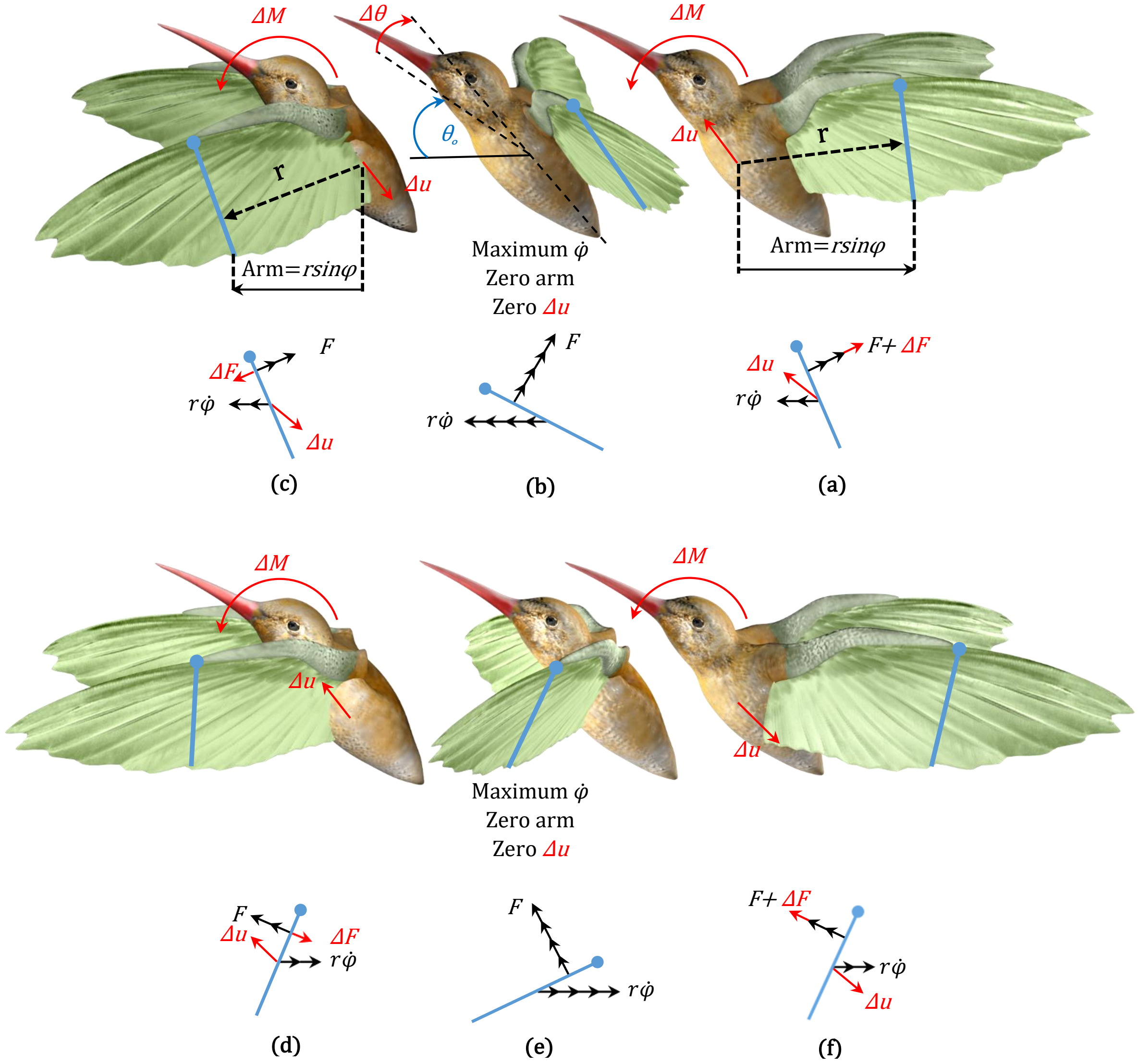

Figure 5.1: Graphical illustration of the physics behind the vibrationally-induced pitch stiffness.

oscillatory Δu is replaced with a constant u-disturbance (i.e., $\overline{u}$), no net restoring pitching moment would be obtained. The discovered pitch stiffness is the result of the interaction (multiplication) between two zero-mean contributions ΔM_u, Δu (both neglected by direct averaging).

Figure (5.1) shows the sequence of events over one flapping cycle after a disturbance $\Delta\theta$ in body pitch is experienced. Below each frame, a wing section is shown with the applied forces. The flapping speed of the wing section, which is a distance r from the root, is $r\dot{\varphi}$. The aerodynamic force due to the flapping speed is F. A pitch up disturbance $\Delta\theta$ (shown in (b)) causes a backward translation of the body, which damps out on the slow time-scale, though with an oscillatory component Δu. ΔF is the change in the aerodynamic force due to this speed component. ΔM represents the associated pitching moment due to the oscillatory moment arm ($r \sin \varphi$) with respect to the body cg. (a)-(c) Downstroke: The wing accelerates forward while pitching down until (b) and then decelerates while pitching up. (c)-(d) Reversal: The wing flips (pitches up) and reverses its direction of motion. (d)-(f) Upstroke: The wing accelerates backward while pitching down until (e) and then decelerates while pitching up. (a) Right after the initial position in downstroke: The body oscillatory speed Δu is forward (with the wing motion) increasing the aerodynamic force. The wing is behind the body CG resulting in a pitching down moment ΔM. (b) Middle position in downstroke: The flapping speed is maximum, but the body oscillatory speed reaches zero along with the moment arm. (c) Right before the farthest position in downstroke: The body oscillatory speed Δu is backward (against the wing motion) decreasing the aerodynamic force, but the wing is ahead of the body CG resulting in a pitching down moment ΔM. (d) Right after the initial position in upstroke: The wing reverses its inclination and direction. The body oscillatory speed Δu changes too; becomes forward (against the wing motion) decreasing the aerodynamic force and resulting in a pitching down moment ΔM as the wing is still ahead of the body cg. (e) Middle position in upstroke: The flapping speed is maximum, but the body oscillatory speed reaches zero along with the moment arm. (f) Right before the farthest position in upstroke: The body oscillatory speed Δu is backward (with the wing motion) increasing the aerodynamic force, but the wing is behind the body CG resulting in a pitching down moment ΔM. As a result, a net pitching down moment ΔM is formed over the flapping cycle that is proportional to the pitch disturbance (because Δu is mainly due and proportional to the $\Delta \theta$); i.e., a restoring spring action.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Flight dynamics of insects and their man-made counterparts, flapping-wing micro-air-vehicles (FWMAVs), was discussed in the longitudinal plane at hover position. This system has a nonlinear, time-periodic model with a large separation between the system's two time scales, which invokes averaging. Utilizing direct averaging, insects/FWMAVs are found to be unstable at hover; mainly due to lack of pitch stiffness. However, by exploiting more rigorous and advanced mathematical tools (higher-order averaging based on chronological calculus), it can be proven that the high-frequency oscillatory aerodynamic forces induce a vibrational control mechanism resulting in a pitch stiffness on the hovering flight dynamics. In other words, vibrational stabilization may express the missing passive mechanism that was not captured by the usual modeling and analysis of insect flight. This unconventional stabilization technique originates from the interaction between the fast wing flapping dynamics and slow body dynamics, which will not be eminent using only direct averaging. An experimental setup in which two degrees of freedom for the body (vertical motion and pitching motion) are implemented, is fabricated to verify/demonstrate such a phenomenon. Since vibrational control is an *open loop* stabilization technique due to the application of a *sufficiently high-frequency periodic forcing*, the stability of the system is investigated at different flapping frequencies. It is shown that the system is naturally (without feedback) stabilized beyond a certain threshold of the flapping frequency (18Hz in the current setup), which conforms with the vibrational control concept. Also, a replica of the system is fabricated in which the flapping bird is replaced with a propeller that revolves at a constant speed to check whether the induced stabilization at high frequencies is mainly due to the periodicity of the driving force (i.e., a vibrational control) or not. The result is that the propeller system replica is unstable at all applied voltages and becomes even more unstable at larger applied voltages (i.e., when it comes closer to the hovering position). At last, it is concluded that insects/FWMAVs, truly, benefit from the wonderful phenomenon of vibrational stabilization, whether or not it suffices to conclude stability for the whole system.

6.2 Future work

To provide the community with a universal scaling, one can study ways to find the threshold frequency in which the vibrational stability happens relating to different aerodynamic characteristics of an insect/FWMAV. This metric can also be nondenominational for instance one can find the ratio between the natural frequency of a segment or the whole bird and the flapping frequency.

One may find it interesting to develop more optimal designs for bio-inspired flying robots by relaxing the stability requirements. Promoting vibrational stabilization in new designs and focusing more on aerodynamic performance might be a suitable approach to design flapping-wing micro-air-vehicles without complicated sensory-control-actuator-processing systems, hence making miniaturization more feasible.

To support the explanation provided regarding physics behind vibrational stability one can use advanced motion capture equipment to study oscillatory yet minimal motion of the body in the flapping setup.

In this study interaction between wing and body in flapping flight was studied. One may find it interesting to find it interesting to investigate interactions between the wing, body and the surrounding air flow. One way of doing it is to equip the surrounding with smoke and study the air flow.

It is interesting to see whether there is a way to implement an external stabilizing mechanism for the propeller setup. Maybe manipulate the input to be oscillatory. This way the propeller makes a time-periodic aerodynamic force as well. If the setup gains stability, then it might provide even more support for the claims this thesis has made.

Bibliography

[1] Diy electric dove super capacitor wing flapping bird toy gift. `https://www.banggood.com/Wing-Flapping-Bird-Toy-Gift-p-1013167.html`.

[2] Gravity 360 degree hall angle sensor. `https://www.dfrobot.com/wiki/index.php/Gravity:_Hall_Angle_Sensor_SKU:_SEN0221`.

[3] Nano qx 6x15mm ballistic 19,600kv. `https://www.amazon.com/Apex-RC-Products-Inductrix-Ballistic/dp/B0711251VV/ref=pd_ybh_a_52?_encoding=UTF8&psc=1&refRID=NASVAW8SBAW1Z82R4CXW`.

[4] A. A. Agrachev and R. V. Gamkrelidze. The exponential representation of flows and the chronological calculus. *Matematicheskii Sbornik*, 149(4):467–532, 1978.

[5] J. Baillieul and B. Lehman. Open-loop control using oscillatory inputs. *CRC Control Handbook*, pages 967–980, 1996.

[6] J. M. Berg and I. M. Wickramasinghe. Vibrational control without averaging. *Automatica*, 58:72–81, 2015.

[7] G. J. Berman and Z. J. Wang. Energy-minimizing kinematics in hovering insect flight. *Journal of Fluid Mechanics*, 582(1):153,168, 2007.

[8] F. Bullo. Averaging and vibrational control of mechanical systems. *SIAM Journal on Control and Optimization*, 41(2):542–562, 2002.

[9] B. Cheng and X. Deng. Translational and rotational damping of flapping flight and its dynamics and stability at hovering. *IEEE Transactions On Robotics*, 27(5):849–864, 2011.

[10] J. M. Dietl and E. Garcia. Stability in ornithopter longitudinal flight dynamics. *Journal of Guidance, Control and Dynamics*, 31(4):1157–1162, 2008.

[11] C. P. Ellington. The aerodynamics of hovering insect flight: Ii. morphological parameters. *Philosophical Transactions Royal Society London Series B*, 305:17–40, 1984.

[12] I. Faruque and J. S. Humbert. Dipteran insect flight dynamics. part 1 longitudinal motion about hover. *Journal of theoretical biology*, 264(2):538–552, 2010.

[13] N. Gao, H. Aono, and H. Liu. A numerical analysis of dynamic flight stability of hawkmoth hovering. *Journal of Biomechanical Science and Engineering*, 4(1):105–116, 2009.

[14] Hartono and A. H. P. Van Der Burgh. Higher-order averaging: periodic solutions, linear systems and an application. *Nonlinear Analysis: Theory, Methods & Applications*, 52(7):1727–1744, 2003.

[15] A. Hassan and H. Taha. Differential-geometric-control formulation for flapping flight multi-body dynamics. *Submitted to Journal of Nonlinear Sciences.*

[16] P. L. Kapitza. Pendulum with a vibrating suspension. *Uspekhi Fiz. Nauk*, 44(1):7–20, 1951.

[17] P. L. Kapitza. Dynamical stability of a pendulum when its point of suspension vibrates. *Collected Papers by PL Kapitza*, 2:714–725, 1965.

[18] H. K. Khalil. *Noninear Systems.* Prentice-Hall, New Jersey, 2002.

[19] A. H. Nayfeh. *Perturbation Methods.* John Wiley and Sons, Inc., 1973.

[20] A. H. Nayfeh and D. T. Mook. *Nonlinear Oscillations.* John Wiley and Sons, Inc., 1979.

[21] R. C. Nelson. *Flight stability and automatic control*, volume 2. WCB/McGraw Hill New York, 1998.

[22] J. A. Sanders, F. Verhulst, and J. A. Murdock. *Averaging methods in nonlinear dynamical systems*, volume 2. Springer, 2007.

[23] A. Sarychev. Stability criteria for time-periodic systems via high-order averaging techniques. In *Nonlinear Control in the Year 2000*, volume 2 of *Lecture Notes in Control and Information Sciences*, pages 365–377. Springer-Verlag, 2001.

[24] A. Stephenson. On an induced stability. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 15(86):233–236, 1908.

[25] W. Su and C. E. S. Cesnik. Flight dynamic stability of a flapping wing micro air vehicle in hover. AIAA-Paper 2011-2009.

[26] M. Sun, J. Wang, and Y. Xiong. Dynamic flight stability of hovering insects. *Acta Mechanica Sinica*, 23(3):231–246, 2007.

[27] M. Sun and Y. Xiong. Dynamic flight stability of a hovering bumblebee. *Journal of Experimental Biology*, 208(3):447–459, 2005.

[28] H. E. Taha, M. R. Hajj, and P. S. Beran. Unsteady nonlinear aerodynamics of hovering mavs/insects. AIAA-Paper 2013-0504.

[29] H. E. Taha, M. R. Hajj, and P. S. Beran. State space representation of the unsteady aerodynamics of flapping flight. *Aerospace Science and Technology*, 34:1–11, 2014.

[30] H. E. Taha, M. R. Hajj, and A. H. Nayfeh. On the longitudinal flight dynamics of hovering mavs/insects. *Journal of Guidance Control and Dynamics*, 37(3):970–978, 2014.

[31] H. E. Taha, A. H. Nayfeh, and M. R. Hajj. Aerodynamic-dynamic interaction and longitudinal stability of hovering mavs/insects. AIAA-Paper 2013-1707.

[32] H. E. Taha, A. H. Nayfeh, and M. R. Hajj. Effect of the aerodynamic-induced parametric excitation on the longitudinal stability of hovering mavs/insects. *Nonlinear Dynamics*, 78(4):2399–2408, 2014.

[33] H. E. Taha, S. Tahmasian, C. A. Woolsey, A. H. Nayfeh, and M. R. Hajj. The need for higher-order averaging in the stability analysis of hovering mavs/insects. 10(1):016002. Selected in the Bioinspiration & Biomimetics Highlights of 2015.

[34] H. E. Taha, C. A. Woolsey, and M. R. Hajj. Geometric control approach to longitudinal stability of flapping flight. *Journal of Guidance Control and Dynamics*, 39(2):214–226, 2016.

[35] G. K. Taylor and A. L. R. Thomas. Animal flight dynamics ii. longitudinal stability in flapping flight. *Journal of Theoretical Biology*, 214, 2002.

[36] G. K. Taylor and A. L. R. Thomas. Dynamic flight stability in the desert locust. *Journal of Theoretical Biology*, 206(6):2803–2829, 2003.

[37] G. K. Taylor and R. Zbikowski. Nonlinear time periodic models of the longitudinal flight dynamics of desert locusts. *J. Roy. Soc. Interface*, 1(3):197221, 2005.

[38] P. A. Vela. *Averaging and control of nonlinear systems (with application to biomimetic locomotion)*. PhD thesis, California Institute of Technology, Pasadena, CA, May 2003.

[39] Y. Xiong and M. Sun. Dynamic flight stability of a bumble bee in forward flight. *Acta Mechanica Sinica*, 24(3):25–36, 2008.

[40] K. Yagasaki and T. Ichikawa. Higher-order averaging for periodically forced weakly nonlinear systems. *International Journal of Bifurcation and Chaos*, 9(03):519–531, 1999.

Appendix A

Visual studio and Arduino Codes

A.0.1 Color detection code (Source.cpp) in C++

```
//The author used a source code written by NATHAN GLOVER to creat the ...
    following source code
// https://handmap.github.io/angle-mapping-based-on-hsv-colour-detection/
#include <opencv2\core\core.hpp>
#include <opencv2\highgui\highgui.hpp>
#include <opencv2\imgproc\imgproc.hpp>
#include <iostream>
#include <math.h>
#include <ctime>
#include <cstdio>
#include <iomanip>
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "SerialPort.h"
```

```
#define PI 3.14159265


using std::cout;

using std::endl;
char incomingData[MAX_DATA_LENGTH];
float interval = 0.001; //seconds
unsigned long currentMillis = 0;
unsigned long previousMillis = 0;
char incomingData2;

namespace cv
{
using std::vector;
}


int main(int, char**)
{


char *port_name = ”COM5”;

SerialPort arduino(port_name);

if (arduino.isConnected()) cout << ”Connection Established” << endl;

else cout << ”ERROR, check port name”;


std::cout << std::fixed;
std::cout << std::setprecision(4);
```

```
cv::VideoCapture cap(1); // open the default camera
if (!cap.isOpened()) // check if we succeeded
return -1;

cv::Mat image_HSV;
cv::Mat image_Color1;
cv::Mat image_Color2;
cv::Moments moments_color1;
cv::Moments moments_color2;

cv::vector<cv::vector<cv::Point> > contours_1;
cv::vector<cv::vector<cv::Point> > contours_2;
cv::vector<cv::Vec4i> hierarchy_1;
cv::vector<cv::Vec4i> hierarchy_2;

cv::Scalar color_1 = cv::Scalar(0, 0, 255);
cv::Scalar color_2 = cv::Scalar(255, 255, 0);
cv::Mat image_frame;
cv::namedWindow(”Angles”, 0);

std::string point1;
std::string point2;


cv::Point center_1;
cv::Point center_2;
for (;;)
{
cap >> image_frame; // get a new frame from camera
if (image_frame.empty()) break;

// Generates HSV Matrix
cv::cvtColor(image_frame,// Input image
image_HSV, // output image in HSV
CV_BGR2HSV); // constant refering to color space transformation

//filtering image for colors
cv::inRange(image_HSV, //input image to be filtered
cv::Scalar(160, 100, 100), //min thresshold value
cv::Scalar(190, 255, 255), // max threshold value
image_Color1); //output image

cv::inRange(image_HSV, //input image to be filtered
cv::Scalar(50, 40, 50), //min thresshold value
cv::Scalar(80, 255, 255),

//cv::Scalar(05, 100, 100), //min thresshold value for yellow
//cv::Scalar(65, 255, 255), //max thresshold value for yellow

//cv::Scalar(100, 80, 20), //min thresshold value for dark green
//cv::Scalar(140, 120, 60), //max thresshold value for dark green
image_Color2); //output image

/// Find contours
findContours(image_Color1, // input image
contours_1, // vector to save contours
hierarchy_1,
CV_RETR_TREE,
CV_CHAIN_APPROX_SIMPLE,
cv::Point(0, 0));

findContours(image_Color2, // input image
contours_2, // vector to save contours
hierarchy_2,
CV_RETR_TREE,
CV_CHAIN_APPROX_SIMPLE,
cv::Point(0, 0));


// Get the moments
cv::vector<cv::Moments> mu_1(contours_1.size()); // initialize a vector ...
    of moments called mu, vector size the number of contours
for (int i = 0; i < contours_1.size(); i++)
{
mu_1[i] = moments(contours_1[i], false);
}

cv::vector<cv::Moments> mu_2(contours_2.size()); // initialize a vector ...
    of moments called mu, vector size the number of contours
for (int i = 0; i < contours_2.size(); i++)
{
mu_2[i] = moments(contours_2[i], false);
}

///  Get the mass centers:
cv::vector<cv::Point2f> mc_1(contours_1.size()); //vector to store all ...
    the center points of the contours.
for (int i = 0; i < contours_1.size(); i++)
{
mc_1[i] = cv::Point2f(mu_1[i].m10 / mu_1[i].m00, mu_1[i].m01 / ...
    mu_1[i].m00);
}

cv::vector<cv::Point2f> mc_2(contours_2.size()); //vector to store all ...
    the center points of the contours.
for (int i = 0; i < contours_2.size(); i++)
{
mc_2[i] = cv::Point2f(mu_2[i].m10 / mu_2[i].m00, mu_2[i].m01 / ...
    mu_2[i].m00);
```

```
}


/// Draw contours

for (int i = 0; i< contours_1.size(); i++)
{
if (mu_1[i].m00>75) {
center_1 = mc_1[i];

drawContours(image_frame, contours_1, i, color_1, 2, 8, hierarchy_1, 0, ...
    cv::Point());
circle(image_frame, mc_1[i], 4, color_1, -1, 8, 0);

//std::cout<< "red pen: " <<mc[i] << '\n';
}

}
for (int i = 0; i< contours_2.size(); i++)
{
if (mu_2[i].m00>50) {
center_2 = mc_2[i];
drawContours(image_frame, contours_2, i, color_2, 2, 8, hierarchy_2, 0, ...
    cv::Point());
circle(image_frame, center_2, 4, color_2, -1, 8, 0);

line(image_frame, center_2, center_1, color_2, 4, 8, 0);
line(image_frame, center_2, cv::Point(-image_frame.cols, center_2.y), ...
    color_2, 4, 8, 0);


}

```

```
}

cv::imshow(”Angles”, image_frame);
if (cv::waitKey(10) >= 0) break;
std::clock_t start;
double duration;

start = std::clock();

duration = (std::clock()) / (double)CLOCKS_PER_SEC;



//Check if data has been read or not

long read_result = arduino.readSerialPort(incomingData, MAX_DATA_LENGTH);

//printf(”%s”, incomingData);
//prints out data

//puts(incomingData);

duration = (std::clock()) / (double)CLOCKS_PER_SEC;

//wait a bit

//Sleep(10);

currentMillis = (std::clock());

if ((currentMillis - previousMillis) >= interval) {
previousMillis = currentMillis;
//seconds = currentMillis;
//            incomingData2 = incomingData  ;
//      std::cout << incomingData << '\n';
//std::cout << "P:" << incomingData << '\t';
std::cout << "t:" << duration << '\t';
std::cout << "a:" << atan2((center_2.y - center_1.y)*1.0, (center_2.x - ...
    center_1.x)*1.0)*(180 / PI) << '\t';
std::cout << "P:" << (incomingData);
Sleep(10);
std::cout << '\n';
//  puts(incomingData);
//  Sleep(10);

}

}
// the camera will be deinitialized automatically in VideoCapture ...
    destructoratan ( ...
    (abs(center_1.y-center_2.y)/abs(center_1.x-center_2.x))* PI / 180.0 )
return 0;
}
```

A.0.2 Class Header File (SerialPort.h) communication between C++ and Arduino

```
// Credit for the outline of this code goes to Manash Kumar Mandal
// https://blog.manash.me/
// The author of this thesis modified it to be able to use it in ...
    correspond to context of his experiment
#ifndef SERIALPORT_H

#define SERIALPORT_H
```

```



#define ARDUINO_WAIT_TIME 0

#define MAX_DATA_LENGTH 7



#include <windows.h>

#include <stdio.h>

#include <stdlib.h>



class SerialPort

{

    private:

    HANDLE handler;

    bool connected;

    COMSTAT status;

    DWORD errors;

    public:

```

```
    SerialPort(char *portName);

    ~SerialPort();



    int readSerialPort(char *buffer, unsigned int buf_size);

    bool writeSerialPort(char *buffer, unsigned int buf_size);

    bool isConnected();

};



#endif // SERIALPORT_H
```

A.0.3 Class Source File (SerialPort.cpp) communication between C++ and Arduino

```
// Credit for the outline of this code goes to Manash Kumar Mandal
// https://blog.manash.me/

#include "SerialPort.h"



SerialPort::SerialPort(char *portName)

```

```
{

    this->connected = false;



    this->handler = CreateFileA(static_cast<LPCSTR>(portName),

    GENERIC_READ | GENERIC_WRITE,

    0,

    NULL,

    OPEN_EXISTING,

    FILE_ATTRIBUTE_NORMAL,

    NULL);

    if (this->handler == INVALID_HANDLE_VALUE) {

        if (GetLastError() == ERROR_FILE_NOT_FOUND) {

            printf("ERROR: Handle was not attached. Reason: %s not ...
                available\n", portName);

        }

        else

        {

            printf("ERROR!!!");

        }

    }

    else {

        DCB dcbSerialParameters = { 0 };



        if (!GetCommState(this->handler, &dcbSerialParameters)) {

            printf("failed to get current serial parameters");

        }

        else {

            dcbSerialParameters.BaudRate = CBR_115200;

            dcbSerialParameters.ByteSize = 8;

            dcbSerialParameters.StopBits = ONESTOPBIT;

            dcbSerialParameters.Parity = NOPARITY;

            dcbSerialParameters.fDtrControl = DTR_CONTROL_ENABLE;



            if (!SetCommState(handler, &dcbSerialParameters))

            {

                printf(”ALERT: could not set Serial port parameters\n”);

            }

            else {

                this->connected = true;

                PurgeComm(this->handler, PURGE_RXCLEAR | PURGE_TXCLEAR);

                Sleep(ARDUINO_WAIT_TIME);

            }

        }

    }

}



SerialPort::~SerialPort()

{

    if (this->connected) {

        this->connected = false;

        CloseHandle(this->handler);

    }

}



int SerialPort::readSerialPort(char *buffer, unsigned int buf_size)

{

    DWORD bytesRead;

    unsigned int toRead = 0;



    ClearCommError(this->handler, &this->errors, &this->status);



    if (this->status.cbInQue > 0) {

        if (this->status.cbInQue > buf_size) {

            toRead = buf_size;

        }

        else toRead = this->status.cbInQue;

    }
    if (ReadFile(this->handler, buffer, toRead, &bytesRead, NULL)) ...
        return bytesRead;


    return 0;

}



bool SerialPort::writeSerialPort(char *buffer, unsigned int buf_size)

{

    DWORD bytesSend;



    if (!WriteFile(this->handler, (void*)buffer, buf_size, &bytesSend, ...
        0)) {

        ClearCommError(this->handler, &this->errors, &this->status);

        return false;

    }

    else return true;

```

```
}



bool SerialPort::isConnected()

{

    return this->connected;

}
```

A.0.4 Arduino code to communicate between Arduino Uno and Visual studio

```
#include <stdio.h>
//#include <printf.h>
#include <stdarg.h>

unsigned long previousMillis = 0;
float interval = 60; //Milliseconds
float seconds = 0;
float analogValue = 0;
float sensorValue = 0;
float ThetaOffset = 185;
unsigned long currentMillis = 0;
float x = 75;//
float PMW = 0;
float pendulumangle=0;
long int pendulumangle00=0;
```

```
String pendulumanglestr;

void setup()
{
    pinMode(A0, INPUT);
    pinMode(4,OUTPUT);
    digitalWrite(4,HIGH);
    pinMode(7,OUTPUT);
    pinMode(9, INPUT);
    //pinMode(potPin2, INPUT);
    Serial.begin(115200);
    PMW = 255*x/100;
    analogWrite(9, PMW);

}
float i = 1000.99;
void loop()
{

    analogValue = analogRead(A0);
    sensorValue = analogValue/1024.0*360;
    //ThetaOffset = 273.87;
    currentMillis = millis();

    if ( (currentMillis - previousMillis) >= interval){
        previousMillis = currentMillis;
        seconds = currentMillis;
        pendulumangle=round((sensorValue-ThetaOffset+180)*10000);
        pendulumangle00=round(pendulumangle);
        //Serial.print(”\A:\t”);
        //pendulumanglestr=String(pendulumangle,2);
        // Serial.write(pendulumangle);
        // Serial.write(str)
        Serial.print(pendulumangle00);
        // i = i + 1.0;
        //Serial.print(”\tT:”);
        // Serial.println(seconds/1000);
        //Serial.printf(”p%3d\n”), pendulumangle);



    }
}
```