

AN ABSTRACT OF THE THESIS OF

Tassilo Selover-Stephan for the degree of Honors Baccalaureate of Science in Mechanical Engineering presented on May 30, 2016. Title: Simulation of Steady Flight of Butterflies and Improvement of Parameter Optimization .

Abstract approved: _____
Roberto Albertani

Recent interest in nano aerial vehicles (NAVs) has sparked an increase in research on the mechanics of biological flapping flight. The ability of NAVs to fly in confined spaces and perform tight maneuvers make them a valuable technological asset. Researching the mechanisms behind small-scale flapping flight promises progress toward the development of such vehicles. This thesis expands previous work on a numerical model of Tree Nymph (*Idea leuconoe*) hovering flight.

The model of the butterfly and the genetic algorithm used for optimizing flight parameters were first reviewed, then improved to streamline the data collection process. Parallel processing was introduced to reduce computation time, automatic result generation was implemented to remove the need for manual transcription, optimization precision was increased significantly, and the ability to continue optimization of older parameter sets was developed.

The model was then expanded to cover steady forward flight. Forward flight was then studied in relation to the butterfly body's pitch angle. Pitching the body forward was resulted in a forward velocity requiring adjustments of other flight parameters such as wing and abdomen parameters. In addition, the effects of flapping frequency and body mass on the steady forward

flight were observed. A direct relationship between flapping frequency and upward velocity was found, although large deviations produced very unstable results. Conversely, an indirect relationship was found between mass and ascent rate.

Key Words: butterfly model forward flight optimization

Corresponding e-mail address: tassilo@selover.net

© Copyright by Tassilo Selover-Stephan
May 30, 2016
All Rights Reserved

Simulation of Steady Flight of Butterflies and Improvement of Parameter Optimization

by
Tassilo Selover-Stephan

A THESIS

submitted to
Oregon State University
University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Mechanical Engineering
(Honors Associate)

Presented May 30, 2016
Commencement June 2016

Honors Baccalaureate of Science in Mechanical Engineering project of Tassilo Selover-Stephan presented on May 30, 2016.

APPROVED:

Roberto Albertani, Mentor, representing Mechanical Engineering

Douglas R. Warrick, Committee Member, representing Integrative Biology

Yiğit Mengüç, Committee Member, representing Mechanical Engineering

Toni Doolen, Dean, University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, University Honors College. My signature below authorizes release of my project to any reader upon request.

Tassilo Selover-Stephan, Author

ACKNOWLEDGEMENTS

First and foremost, I would like to acknowledge and thank Dr. Roberto Albertani, who guided me through the process of researching and writing my first thesis. This project could not have been completed without his support, and as such I am extremely grateful. Secondly, I would like to thank Tyler Wilson for, despite having already graduated from Oregon State University, taking time to help me learn to use the code he developed three years ago. His assistance was critical to the completion of this work. Thanks go out to Congcong Hu, as well, for providing me with the necessary files to rebuild the numerical model and coming in at odd hours to help me. I would also like to acknowledge Connor Kelsay, who recommended me to Dr. Albertani and helped give me the opportunity to participate in this research. The work of Dr. Ty Hedrick is greatly appreciated for laying the groundwork of the numerical model used in this thesis, as well as his work on hawkmoths for helping me understand the theory behind the numerical model used in this thesis. I am grateful to Mr. Johnny Evers of the Air Force Munitions Directorate at the Air Force Base in Elgin, Fl and Dr. Thomas Emmel, Director of the McGuire Center for Lepidoptera and Biodiversity in the Florida Museum of Natural History for their work helping lay the foundations and experimental basis of this thesis. Thanks go out to Matthew Goettl at the University of West Florida for his outstanding work on the post-processing of experimental butterfly data. Lastly, I would like to thank the members of my committee, Dr. Yiğit Mengüç and Dr. Douglas Warrick, for taking the time out of their hectic schedules to participate in my thesis defense.

TABLE OF CONTENTS

| | |
|---|----|
| Chapter 1: Introduction | 1 |
| Chapter 2: Literature Review & Theoretical Background | 3 |
| 2.1 Study of Insect Flight | 3 |
| 2.2 Modeling of Insect Flight | 4 |
| 2.3 Butterfly in Forward Flight | 5 |
| 2.4 Flight Parameter Estimation Methods | 6 |
| Chapter 3: Methodology | 8 |
| 3.1 Model | 8 |
| 3.1.1 Morphology | 8 |
| 3.1.2 Coordinate Systems and Angles | 9 |
| 3.1.3 Kinetic Parameters | 10 |
| 3.1.4 Forces | 11 |
| 3.1.5 Inverse Parameter Optimization | 12 |
| 3.2 Forward Flight | 13 |
| 3.3 Modifying Flapping Frequency and Mass | 14 |
| 3.3.1 Flapping Frequency | 14 |
| 3.3.2 Mass | 15 |
| Chapter 4: Improvements to Model and Optimization Process | 16 |
| 4.1 Original Parameter Optimization Process | 16 |
| 4.2 Parallel Processing | 19 |
| 4.3 Auto-generation of Results | 20 |
| 4.4 Increases in Precision | 21 |
| 4.5 Starting Optimization with a Founder | 22 |
| Chapter 5: Results & Discussion | 24 |
| 5.1 Forward Flight Optimization with Varying Pitch | 24 |
| 5.2 Modified Flapping Frequency and Mass | 29 |
| 5.2.1 Flapping Frequency | 29 |
| 5.2.2 Mass | 30 |
| Chapter 6: Conclusions | 32 |
| Chapter 7: Future Work | 34 |
| Bibliography | 35 |

List of Figures

| | |
|---|----|
| Figure 2.1: Forces acting on the butterfly during (a) hovering and (b) constant velocity forward flight..... | 5 |
| Figure 3.1: Image of the butterfly model used in simulations | 8 |
| Figure 3.2: Illustrations of the angles and coordinate systems used..... | 10 |
| Figure 4.1: The settings used to manipulate settings of the genetic algorithm before running the optimization. | 17 |
| Figure 4.2: The simulation interface window allowing a parameter set to be inputted into the model and data about its flight to be collected. | 19 |
| Figure 4.3: Positions over time for (a) a set with a parameter precision of 24 digits to the right of the decimal point and (b) the same set with only 6 digits to the right of the decimal. | 22 |
| Figure 5.1: Positions over time for (a) a randomly generated parameter set, (b) a parameter set optimized at a pitch angle of 10° , (c) another set optimized at 30° , and (d) a set optimized at 50° | 25 |
| Figure 5.2: Pitch over time for (a) a randomly generated parameter set, (b) a parameter set optimized at a pitch angle of 10° , (c) another set optimized at 30° , and (d) a set optimized at 50° | 27 |
| Figure 5.3: Position history of a parameter set optimized at a pitch angle of 10° for 50 flaps..... | 28 |
| Figure 5.4: Positions over time of the set optimized at a pitch of 30° with (a) a flapping frequency of 8.8 Hz, (b) a flapping frequency of 9 Hz, and (c) a flapping frequency of 9.2 Hz . | 30 |
| Figure 5.5: Positions over time of the set optimized at a pitch of 30° with (a) a mass scaling of 0.95, (b) a mass scaling of 1.00, and (c) a mass scaling of 1.10..... | 31 |

List of Tables

| | |
|---|----|
| Table 3.1: Parameters of the inverse parameter optimization | 12 |
| Table 3.2: Objective pitch and altitude of the three flights optimized to achieve forward flight | 14 |
| Table 3.3: Frequencies* used to test the effect of flapping frequencies on modeled flight | 14 |
| Table 3.4: Scaling factors used to test the effect of changing mass on flight | 15 |
| Table 5.1: Kinematic parameters of simulations run at various pitch angles..... | 24 |

Chapter 1: Introduction

Due to recent advances in technology, especially the commercial drone market, it has become more accessible than ever to research unmanned vehicles on progressively smaller scales. The field of very small unmanned aerial vehicles, or nano aerial vehicles (NAVs), is of particular interest due to the possibility novel applications of drone technology. NAVs offer many advantages over traditional drones, and have the potential to fill many niches in the civilian and military markets. They have the ability to fly confined, smaller spaces under more turbulent conditions than traditional unmanned aerial vehicles. An additional advantage of NAV size is their ability to perform rapid maneuvers with their low inertial masses and their difficulty to detect, allowing them potentially covertly perform missions in locations that traditional methods cannot access unnoticed. With the objective of enabling the production of ever smaller NAVs, interest in the flight mechanics of small flying systems has increased in recent decades. Small flapping animals and insects such as hummingbirds or dragonflies are a constant source of bio-inspiration for research of NAVs. Study of their kinetics, clever use of unsteady fluid mechanics phenomena at low Reynolds numbers, and potential applications to materials development has produced promising results and hints at yet further advances.

In the interest of furthering the research on small flappers, this thesis seeks to continue and improve upon the work on T. Wilson in their work with the Tree Nymph butterfly *Idea leuconoe* modeling its flight [1]. This research is focused, in particular, on their work developing a model of the butterfly, written in MATLAB, which was able to successfully simulate the kinetics and fluid mechanics involved in the butterfly's hovering flight with an input of 14 independent variables, or parameters. By making use of an optimization method involving a

genetic algorithm, T. Wilson developed a process by which optimized parameters could be generated to produce stable hovering flight, desired result in the model.

The original code needed modifications for more general and user-friendly operation. Once the model had been reviewed and improved, work began on developing a parameter set for the model to simulate the butterfly in stable forward flight. During this process, progress was also made improving the optimization process and data collection methods.

Background information on the study and modeling of insect flight, and the estimation of flight parameters are outlined in Chapter 2. In Chapter 3, methodology is covered, including an explanation of T. Wilson's numerical model and the optimization methods they used, the methodology for achieving forward flight in the numerical model, and the methodology of manipulating flapping frequency and mass during steady flight. Outlined in Chapter 4 is the process of running the parameter optimization algorithm and improvements made to it. Parallel processing was introduced to reduce optimization time, automatic generation of results was implemented to save the user time they would have spent transcribing results, precision of the parameter optimization process was increased significantly, and the ability to continue optimization of older parameter sets using a founder set was developed. Chapter 5 presents the results of optimizing for forward flight in the numerical model at varying body pitch angles and tests of forward flight involving varying flapping frequencies and body masses. Conclusions and recommendations for future work on this subject are discussed in Chapters 6 and 7, respectively.

Chapter 2: Literature Review & Theoretical Background

2.1 Study of Insect Flight

The science of modeling human inventions on naturally occurring mechanics, also known as biomimetics, is an ancient method dating back many centuries. Life on earth has had billions of years to adapt its physiology to its environment, providing a vast ocean of innovations waiting to be integrated into the progress of technology and science. Flight has been a specific point of interest in biomimetics, occupying the time of many successful inventors including Leonardo da Vinci and the Wright Brothers [2, 3]. The process of understanding the principles of flight and building the first manmade heavier-than-air vehicles was a long journey that lasted from the dawn of humanity to the beginning of the 20th century and relied heavily on the observation of nature. Even so, the mechanics of flight were known as far back as the 18th century [2]. The study of insect flight, however, was more difficult due to the complexities of fluid dynamics at smaller scales. Insects appear to disregard widely accepted aerodynamic design principles through their use of flat, non-streamlined wings and the very large angles of attack they employ during flight [4]. To account for this apparent contradiction, mechanisms have been theorized such as leading edge vortices and clap-and-fling, or the Weis-Fogh Mechanism [5, 6]. These proposed mechanisms have been studied more closely and become better understood through the use of computational fluid dynamics [7, 8].

2.2 Modeling of Insect Flight

As the computational power available to researchers has increased, their ability to employ significantly complex models in their research also expanded. While experimental data will almost always be the most accurate way to gain information about a system such as a flying insect, data can also be collected more quickly and cheaply through use of a computed model. Use of finite element analysis or computational fluid dynamics do not require an experimental set up, and may, depending on usage, deliver data accurate enough to derive results from. In addition, a model can be queried for more data than an experimental setup can, such as internal stresses, temperature distribution, or moments.

In this thesis, the model employed is a forward differential equation model, as developed by T. Wilson in his work [1]. Inputted into the model as initial conditions are the position and velocity of the butterfly as well as parameters for the articulation of the wings and abdomen. These initial conditions and computed forces on the body and wings are then used to calculate the position and velocity of the butterfly in the next time step using a set of differential equations derived from Newton's Laws of Motion. Wing forces are computed using a quasi-steady blade-element approach which accounts for unsteady effects by treating them as inertia due to added fluid mass [9]. Inertial and gravitational forces and moments can be calculated easily because the dimensions and masses of each body part are known. Through the use of an ordinary differential equation solver, the model can be computed using a given time span, time step, and initial conditions.

2.3 Butterfly in Forward Flight

To achieve a steady forward flight of the butterfly, the system was considered in terms of force balances. The forces acting on the butterfly during any given flight can be simplified to lift, drag, and weight, which are not dissimilar to those acting on a helicopter. Helicopters fly forward by pitching their rotor plane down and increasing thrust [10]. To maintain altitude while moving forward, the lift needs to overcome both the forces of weight and drag, which is why horizontal component of lift needs to be increased. Figure 2.1 illustrates the forces of both hovering and forward flight. To achieve a similar result in the model butterfly, a series of parameter sets were optimized at a range of pitch angles to maintain both their pitch and their altitude. Ideally, the optimization routine would then adjust the flight of the butterfly to maintain altitude at the new pitch angle and gain increased forward thrust due to the greater horizontal component of the lift. This increased thrust should come in the form of an increased flapping frequency. The inverse relationship between pitch angle and forward velocity has been demonstrated in *Drosophila* [11].

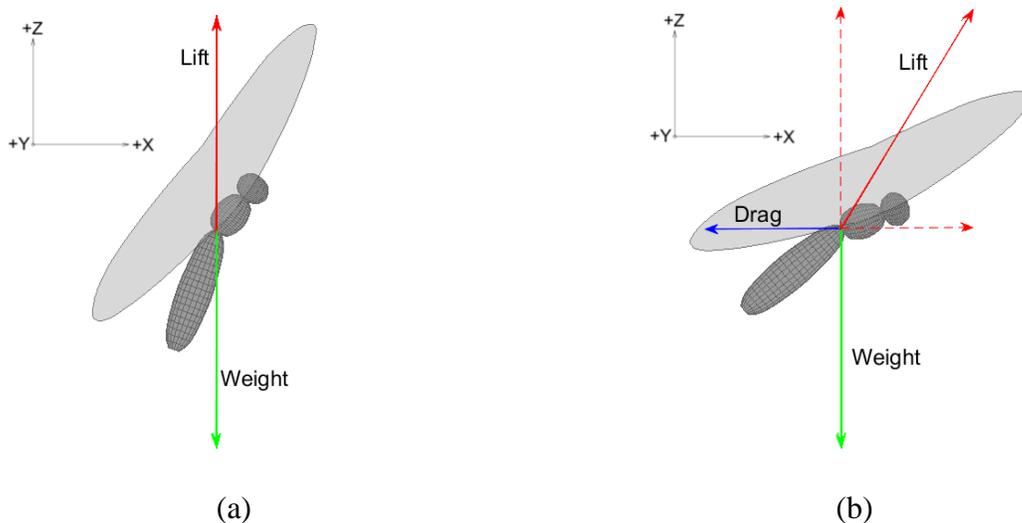


Figure 2.1: Forces acting on the butterfly during (a) hovering and (b) constant velocity forward flight

2.4 Flight Parameter Estimation Methods

Parameter estimation is an inverse problem in which a certain number of parameters are estimated in order to approach a desired outcome within the bounds of the model. Inverse problems, in contrast to forward problems, seek to solve for inputs and have known outputs. An example of a forward problem is the maximum speed of a car could be solved as a function of its shape, power, and tire size. The inverse problem complementing this case would be to solve for a combination of shape, power, and tire size that produce a predetermined outcome speed. In this thesis, the kinematic parameters of the butterfly's wings and body are estimated to produce the expected flight pattern.

The approach used by T. Wilson, and thereby also this thesis, to estimate the kinematic parameters of the butterfly is known as a genetic algorithm [1]. Genetic algorithms were developed by John Holland in collaboration with others in the sixties and seventies, and were intended to mimic the process of natural selection through use of mechanisms such as selection, mutation, crossover, and recombination [12]. Like evolution, the process of running a genetic algorithm takes place in generations. Each generation consists of a population of a certain number of individuals, each of which represents set of parameters for the model. The fitness of each member of a given generation's population is measured relative to the other individuals and used to create a successor generation. The parameters of the individuals of the following generation are determined by selecting the best individuals of the parent generation, mutating individuals randomly, or crossing over parameters between individuals randomly.

To expand on the analogy with the maximum speed of the car, a genetic algorithm could be used to generate a set of parameters that result in the correct top speed. The algorithm would begin by randomly selecting a set of shape, power, and tire size values for each individual and

ranking them by how close their top speed is to the desired top speed. The next generation is then filled with a mix of individuals chosen to move on from the previous generation and individuals referred to as children that are generated using members of the previous generation as parents.

The individuals allowed to pass directly to the next generation, or selected, are chosen randomly, but with weighting applied to those individuals with higher fitness to increase their chances of passing to the next generation. Elitism works similarly to selection, but simply chooses the top nth percentile of the previous generation and automatically selecting them, thereby ensuring the fittest members of a given population won't be lost. To fill the rest of the next generation, the genes, or parameters of the parent generation are crossed to create child individuals. In addition to crossing the genes of parents over into children, the genes may mutate during crossover. This process of producing new generations is then iterated until a satisfactory set of input parameters are found. The best parameter set is that of the fittest individual over all generations.

In this work, the genetic algorithm was used to optimize the flight parameters used by the model to simulate a flapping butterfly. The individuals in each generation were represented by sets of parameters, each parameter being analogous to a gene. Fitness was rated by simulating flight and comparing the results to the expectations set. In the case of T. Wilson's work, the goal was steady hovering [1]. In the case of this work, the goal is steady forward flight. Each successive generation's fittest individual, ideally, flies more closely to expectations than the last.

Chapter 3: Methodology

3.1 Model

The model used to simulate the butterfly flight in this thesis is based on the model developed by T. Wilson in his work with the same subject matter [1]. This section will give an overview of the numerical model so that the results of experiments performed using it and improvements made to it may be better understood.

3.1.1 Morphology

The butterfly is modeled as a body with five parts: the head, the thorax, the abdomen, and the two wings. Although actual *Idea leuconoe* have four wings, two front and two rear, T. Wilson was able to create a composite wing that approximates the aerodynamic properties of its actual counterparts. The body parts are modeled with constant density throughout, and are illustrated in Figure 3.1.

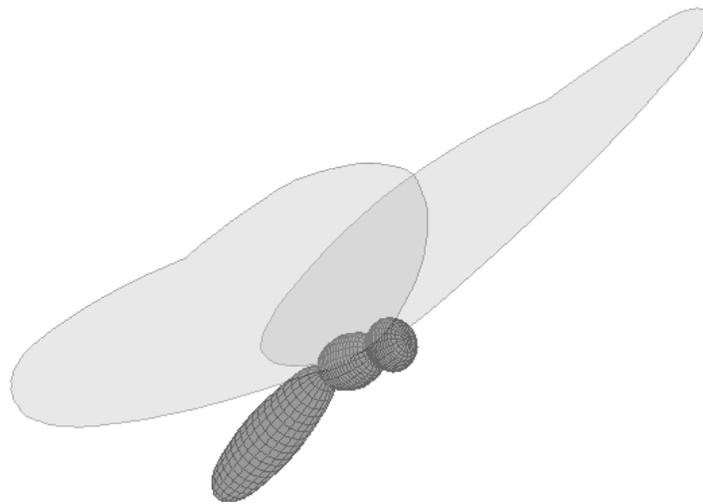


Figure 3.1: Image of the butterfly model used in simulations

3.1.2 Coordinate Systems and Angles

Two coordinate systems are used to define the position and orientation of the butterfly at any given time. A fixed, global system functions as a reference point for the position of the butterfly and a moving body system helps define the orientation and position of each body part. The body system's X and Z axes are depicted in green and blue, respectively, in Figure 3.2 (c). Also in Figure 3.2 (c) can be seen the fixed global coordinate system in dotted lines labeled as X' and Z'. The XZ planes of both systems are mated together, restricting the butterfly's movement to X or Z directions.

To understand the way the model is controlled, particularly the wings, it is important to gain an understanding of the angles used to define the orientation of the body parts. These angles can be categorized as wing angles and body angles. There are three wing angles: the flapping angle ϕ is defined as the wing's angle relative to the Y axis on the YZ plane and can be seen in Figure 3.2 (a). The feathering angle α is the angle the wing is rotated about its own axis, moving the leading edge of the wing up or down. It is also illustrated in Figure 3.2 (a). The last wing angle, the sweeping angle θ , is the wing's angle relative to the Y axis on the XY plane and can be seen in Figure 3.2 (b). There are only two body angles: γ and θ_p , both of which are illustrated in Figure 3.2 (c). The abdomen angle γ is the angle between the longitudinal axes of the abdomen and the thorax on the XZ plane. The pitching angle θ_p is the angle between the X axis of the global coordinate system and the X axis of the body coordinate system.

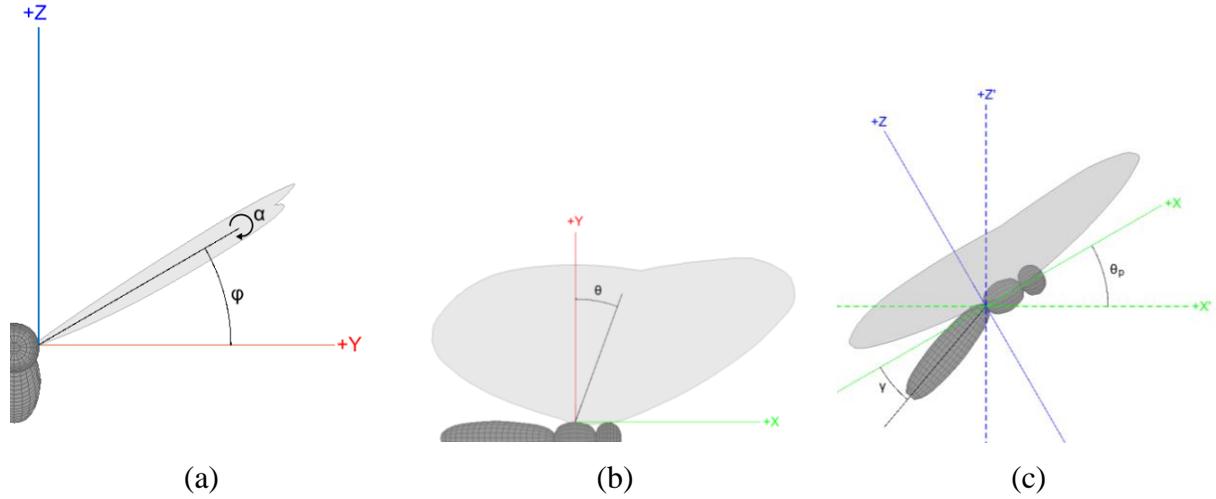


Figure 3.2: Illustrations of the angles and coordinate systems used. (a) Flap angle ϕ and feather angle α . (b) Sweep angle θ . (c) Abdomen angle γ and pitch angle θ_p . XYZ is the body coordinate system, while X'Y'Z' is the global coordinate system.

3.1.3 Kinetic Parameters

The butterfly's body, wing, and abdomen orientations have been defined, which leads to the subject of articulating the butterfly's body parts. During simulated flight, each angle as described in Section 3.1.2 is defined by a time-dependent sinusoidal function except θ_p , which is calculated as a dependent variable. The value of ϕ , α , θ , or γ at any given time, defined here as θ_t can be defined using Equation 3.1 [1]:

$$\theta_t = \theta_A \sin(2\pi ft + \theta_{ph}) + \theta_m \quad (3.1)$$

where θ_A is the angle's amplitude, f is the frequency, θ_{ph} is the angle's phase, and θ_m is the angle's mean. All wing angles share a flapping frequency f while the abdomen angle γ has its own abdomen frequency f_a .

Because the value at any given time, θ_t , of the articulated angles has 3 independent variables, this results in 3 parameters defining each angle, plus 2 more for the frequencies.

Altogether, this adds up to 14 parameters, called a parameter set, defining any particular configuration of the model.

3.1.4 Forces

The instantaneous forces acting on the butterfly are the key to solving for its acceleration, velocity, and position at any given time t . There are two types of forces acting on the body parts of the butterfly: aerodynamic forces and inertial forces.

Aerodynamic forces on the wings are calculated using a quasi-steady blade-element approach [1]. This involves splitting the wings into 20 elements and calculating the aerodynamic forces on each to account for changing conditions along the wing span. The forces on each element were computed as a combination of three aerodynamic forces: a rotational force, a force due to added fluid mass, and a drag force. This approach was developed by Sane and Dickinson [9].

Inertial forces are calculated in both rotational and linear modes, while gravitational forces are solved for with application of Newton's Second Law. To find the total forces and moments acting on the butterfly's body, the center of mass is first calculated between all of the body parts followed by the additions of both types of forces to estimate a total force. Finally, the moment of each force acting on the body center of mass is calculated and they are summed to find a total moment.

From the total force and moment calculated, a set of ordinary differential equations are developed using the laws of motion. And ODE solver is then employed to compute the position and orientation of the butterfly and its body parts over a defined time span.

3.1.5 Inverse Parameter Optimization

Because the model has 14 input parameters defining flight, inversely finding a configuration that results in stable flight becomes a very complicated problem. A genetic algorithm was used to iteratively optimize all 14 inputs, which are listed in Table 3.1. A total of 40 individuals were used per generation. Each optimization routine was run until 20 generations occurred without improvement, which was considered to indicate the current gene pool was at its best state.

Table 3.1: Parameters of the inverse parameter optimization

| Flapping | | | Sweeping | | | Feathering | | | Abdomen | | | Flap | Abd. |
|----------|----------|----------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------|-------|
| Amp. | Mean | Phase | Amp. | Mean | Phase | Amp. | Mean | Phase | Amp. | Mean | Phase | Freq. | Freq. |
| ϕ_A | ϕ_m | ϕ_p | θ_A | θ_m | θ_p | θ_A | θ_m | θ_p | γ_A | γ_m | γ_p | f | f_a |

Mutation probability between generations was 5%, crossover probability 90%, and elitism 10% [1]. Unlike the work of T. Wilson, the objective of the genetic algorithm was steady forward flight. In the scope of this research, it was characterized by a steadily maintained altitude and orientation, and a constantly increasing X position or steady velocity. As such, the genetic algorithm's objective in this thesis was to minimize vertical displacement and pitch deviation while allowing free movement horizontally. An objective function for steady altitude maintenance was defined as:

$$f_{obj} = W\Delta_{pitch} + (1 - W)\Delta_z \quad (3.2)$$

where f_{obj} is the objective fitness value, W is the weighting factor, Δ_{pitch} is the pitch deviation, and Δ_z is the altitude deviation. A weighting factor of 0.8 was used in this thesis.

Although Δ_{pitch} and Δ_z are calculated in the same way as in [1], only deviation in the Z-

direction is considered in this case [1]. This is achieved by integrating the instantaneous deviations over the time span of the flight and dividing them by a normalizing factor.

The individuals in the optimization routines were simulated for a total of 10 flapping cycles. Initial positions were set to a pitch of θ_{obj} , a position of zero in all axes, and with zero velocity. Because θ_{obj} was manipulated as an independent variable, it will be discussed later in Sections 3.2 and 5.1. The bounds of variation for each parameter were based on observed flight data of butterfly flight by R. Albertani, M. Goettl, and T. Wilson [13].

3.2 Forward Flight

A steady forward flight of the butterfly within the bounds of the model requires a change in pitch and an increase in lift as is discussed in Section 2.3. The increase in lift at the lower angle will produce a horizontal component, visible in Figure 2.1, which accelerates the butterfly forward. Because the optimization method used to generate parameter set does not allow the used to explicitly increase lift, a method needed to be found to produce the greater lift required for forward flight. The solution was to decrease the pitch, which can be specified, and optimize the model to maintain altitude. The optimization algorithm will find the best parameter set to meet the pitch angle and altitude requirements, ideally by increasing lift. If lift increases as the butterfly's pitch angle is decreased, the lift's horizontal component should grow and thereby increase the forward velocity of the butterfly. Table 3.2 shows the objective pitches and altitudes of each flight optimization performed.

Table 3.2: Objective pitch and altitude of the three flights optimized to achieve forward flight

| Flight Number | Objective Pitch $\theta_{P,obj}$ | Objective Altitude Z_{obj} |
|---------------|----------------------------------|------------------------------|
| Flight 1 | 10° | 0 cm |
| Flight 2 | 30° | 0 cm |
| Flight 3 | 50° | 0 cm |

3.3 Modifying Flapping Frequency and Mass

In this section, a parameter set was manipulated to modify its flapping frequency and the overall mass of the modeled butterfly. The intent behind this test was to verify that faster flapping does produce more lift, and that a decrease in overall mass will result in update acceleration.

3.3.1 Flapping Frequency

To isolate the effects of higher or lower flapping frequency, a parameter set in stable flight was chosen and its flapping frequency f was manipulated. The set used is the set optimized at 30° pitch and is outlined in more detail in Section 5.1. In order to mitigate instability during flight, the abdomen frequency f_a was set to the same value as the flapping frequency f , which can be justified by results in Section 5.1 showing that abdomen frequency during optimization was very close to flapping frequency. Table 3.3 shows the flapping frequencies used.

Table 3.3: Frequencies* used to test the effect of flapping frequencies on modeled flight

| Flight # | Flight 1 | Flight 2 | Flight 3 |
|--------------------|----------|----------|----------|
| Flapping Frequency | 8.8 Hz | 9 Hz | 9.2 Hz |

* Both wing and abdomen frequencies were set to these values

3.3.2 Mass

Similarly to the experiment performed on flapping frequency, the experiment involving changing the overall mass of the butterfly was performed by choosing a flying parameter set and manipulating the mass of each body part by a predetermined scaling factor. The model chosen was the same set used in Section 5.1, whose parameters can be found in Table 5.1. To demonstrate the effect of mass change, each body part was scaled by a factor as listed in Table 3.4.

Table 3.4: Scaling factors used to test the effect of changing mass on flight

| Flight # | Flight 1 | Flight 2 | Flight 3 |
|---------------------|----------|----------|----------|
| Mass scaling factor | 0.95 | 1.00 | 1.10 |

Chapter 4: Improvements to Model and Optimization Process

In addition to work done on studying the model and its behavior, significant effort was spent during this research to improve the data collection methods. This chapter will provide an overview of each of the improvements to data collection and the optimization process that were integrated into the original MATLAB code [1]. Section 4.1 provides a description of the data collection process, while Sections 4.2 through 4.5 will detail the applied improvements.

4.1 Original Parameter Optimization Process

The parameter optimization process at the beginning of this thesis project consisted of three parts: 1) the definition of the optimization objective and genetic algorithm settings, 2) the initialization of the optimization process, and 3) the collection of optimized parameters.

To define the objective function, it is necessary to find the objective file and edit it manually. The objective file contains the settings for the desired outcomes of the flight such as objective position or pitch. In addition, this file includes all the settings for the initial conditions of the butterfly flight, including initial position, velocity, atmospheric density, gravitational acceleration, and the number of flapping cycles to simulate.

The settings for the genetic algorithm are stored in a separate file that will be referred to as the algorithm settings file. In this file, the settings for the bounds of each parameter to be optimized are stored, as well as the generation method, mutation method, and crossover method.

Once the settings for the objective function and the genetic algorithm have been finalized, the next step is to begin the optimization process. There is a file included, the initialization file, which begins the process of running the genetic algorithm's routine. Once the genetic algorithm

has been started, a window is shown to allow the user to change more settings of the algorithm as illustrated in Figure 4.1. Parameters include the number of individuals in the population per generation, the mutation probability, the crossover probability, the elitism percentile, and conditions to stop the process. Once ‘go’ is pressed, the genetic algorithm will generate a founding generation randomly within the parameter bounds defined in the algorithm settings file. Following this, each individual’s fitness is rated by inputting its parameters into the objective file, where flight will be simulated.

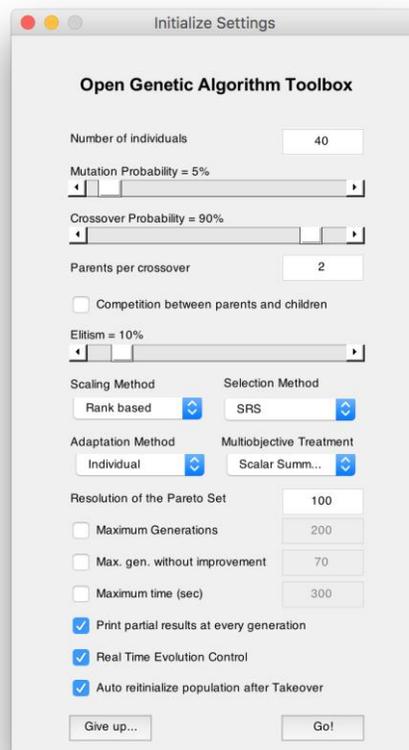


Figure 4.1: The settings used to manipulate settings of the genetic algorithm before running the optimization.

The objective file measures fitness by collecting the kinetic parameters inputted into it and using the predefined initial conditions mentioned above to perform the calculations detailed

in Section 3.1. The result of these calculations is a list of positions and orientations over time that describe the flight path of the butterfly. Once the position and pitch histories have been extracted from the flight simulation, they can be used to score the flight, or fitness. Using the method outlined in Section 3.1.5, a fitness value representing deviation from the objective values is allocated to the individual parameter set.

After the fitness values for the generation's entire population have been computed, the genetic algorithm reports the parameters of the set with the lowest fitness values, thereby deviating least from the objective, and begins populating the next generation as is described in Section 2.4's description of the genetic algorithm process. This process then iterates until the stop conditions are met or the user aborts it.

Once the optimization process has ended, the user can extract its results. This was originally performed by scrolling through the output logs of the genetic algorithm and searching for the fittest individual. This individual's parameter set was then copied and saved for later use or checked immediately using the simulation interface described below. The process of finding the fittest individual and copying its parameters by hand was improved in the current work and is covered in more detail in Section 4.3. The simulation interface, shown in Figure 4.2, is a window allowing the user to specify initial conditions, motion parameters, environmental parameters and scaling factors for size and mass. Using the settings inputted, it is then possible to compute the parameter set using the model for a certain number of flaps. This data can then be plotted using the 'plot results' button, which plots position, velocity, pitch, and pitch velocity over time or the 'plot forces' button, which plots forces and moments of the wing and abdomen over time. The flight can also be animated and saved to a video file using the 'animate results' button. Finally, a parameter set can be saved to or loaded from a file. The saving function was originally the only

way to save newly optimized parameter sets, but was later improved as will be discussed below.

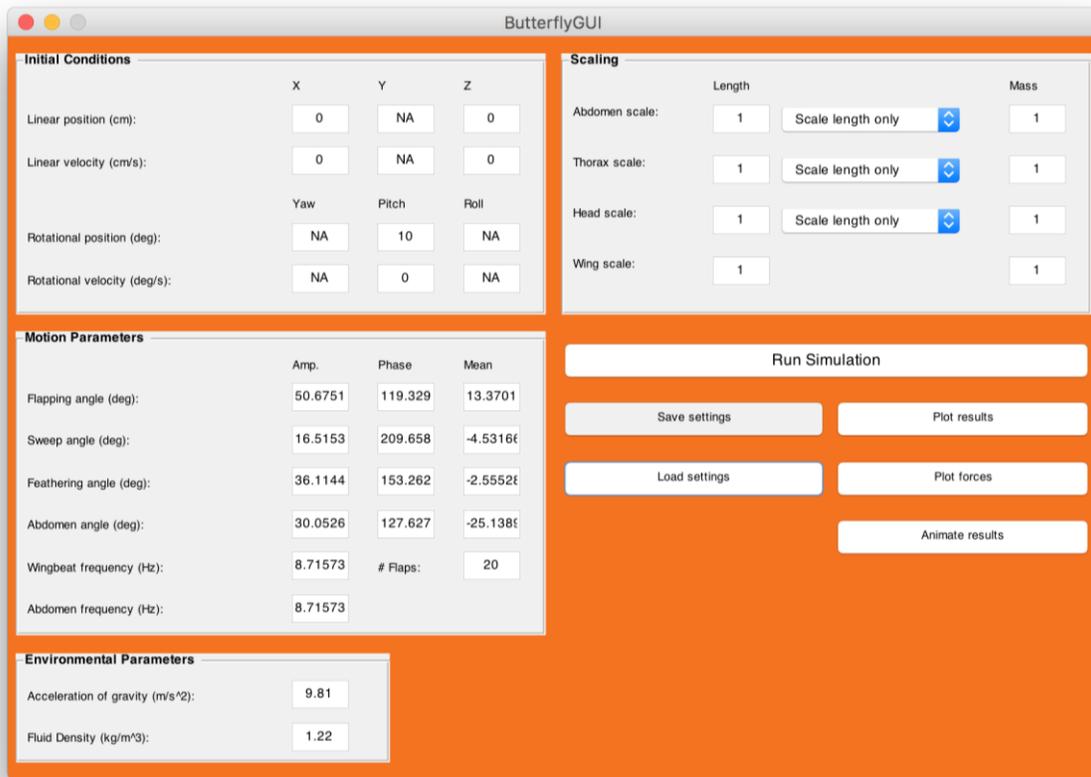


Figure 4.2: The simulation interface window allowing a parameter set to be inputted into the model and data about its flight to be collected.

4.2 Parallel Processing

Because the genetic algorithm needs to evaluate the fitness of each individual in a generation's population, the position and pitch histories of every parameter set needs to be computed. For 10 flaps, this process generally takes 10 to 15 seconds on a consumer computer. At 40 individuals per generation, it can take 6 to 10 minutes to evaluate a population. Most of the optimizations performed within the scope of this thesis took at least 40, sometimes as many as 80 generations to converge on a good solution, resulting in optimizations that needed to be run

overnight or throughout the day to produce any results.

To help alleviate the time burden on the user running optimizations, the evaluation of each individual was broken up into a set of jobs to be run in parallel by all the processor cores available to the computer running the optimization. This was done by employing the use of the MATLAB function `parfor`, which enables the loop's iterations to be run in parallel under the condition that each loop iteration must not depend on any other iteration [14]. Because of the nature of parallel processing, the iterations of a loop may be allocated to each processor core in unexpected ways, resulting in the loop, or even individual lines of code, being evaluated out of order. The task of computing each individual of a population fits these criteria because each set's parameters are predefined and can be modeled completely independently of each other.

Through the implementation of parallel processing, the time to evaluate a generation decreased to 3 to 4 minutes, and the time for a full optimization closer to 3 hours. This allows more optimizations to be run more quickly, and saves the user time.

4.3 Auto-generation of Results

During the process of collecting data detailed in Section 4.1, it was explained that collection of the result of an optimization routine, often the work of hours, relied on searching through logs for the fittest parameter set and transcribing it by hand. The next step was to input these parameters into the simulation interface pictured in Figure 4.2 and save them manually. This process was tedious and also imprecise in ways that will be covered in Section 4.4, so a new solution was implemented to eliminate the need for transcription.

To streamline the task of extracting parameter sets from the optimization process and examining their effects on the flight of the modeled butterfly, a method of automatically

generating records of the fittest parameter sets of each generation was implemented. This method produces a file after each generation is evaluated for its fittest member and transcribes this member's parameter set to the file. This file can then be loaded into the simulation interface and used to run a flight. In addition to eliminating a step in the process of data collection, this feature produces files with initial conditions matching those of the objective file, so the initial position, velocity, orientation, and gravity and air density no longer need to be manually entered. A final advantage of this system is that it produces parameter set files smaller than 2 kilobytes, while the previous method produced files of at least 7 megabytes, or 3,500 times larger.

4.4 Increases in Precision

To accurately simulate and collect data from the parameter sets produced by the genetic algorithm, it becomes necessary to precisely transcribe each parameter to the simulation window. When the ODE solver computes the flight of the butterfly using 100 time steps per flapping cycle, any small change in the input parameters, can have a disproportionately large effect on the final result of the simulation. As the ODE solver extrapolates from the very small differences in values between the optimized parameter and the transcribed parameter, the gap between the two increases with each step.

To give an example, Figure 4.3 shows position over time of two parameter sets that are identical except that Figure 4.3 (a) uses transcribed parameters with 24 digits after the decimal point and Figure 4.3 (b) only uses 6. This means the maximum difference between any two parameters between the two sets is $1 * 10^{-6}$. When looking at the results, however, it becomes apparent that the final X position of the two sets differs by almost 50%.

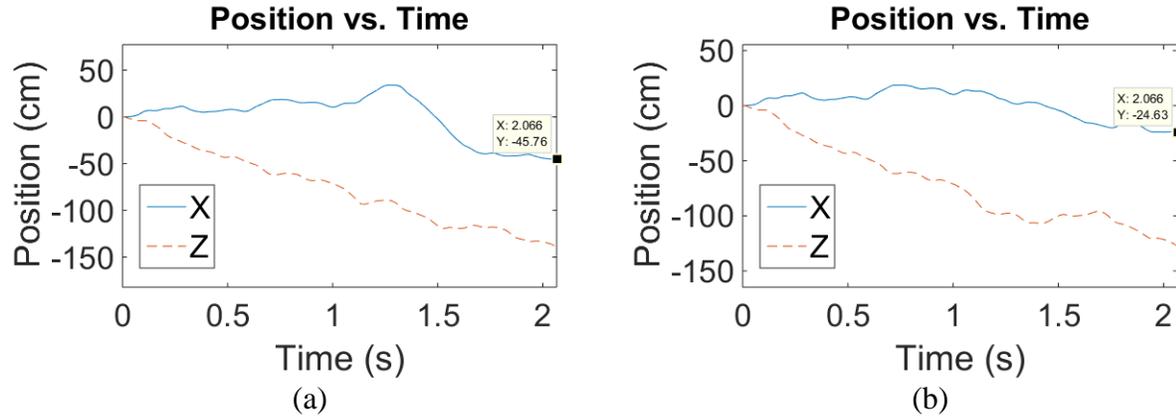


Figure 4.3: Positions over time for (a) a set with a parameter precision of 24 digits to the right of the decimal point and (b) the same set with only 6 digits to the right of the decimal.

The original method of transcribing parameter sets directly from the optimization routine to the simulation interface had a precision of 5 significant digits, which resulted in confusing simulations that did not reflect optimized parameter sets. The solution was the implementation of the automatically generated result files, which feature a total of 24 digits to the right of the decimal point, eliminating the problem of losing precision during transcription and improving the accuracy of data collected.

4.5 Starting Optimization with a Founder

As mentioned in Section 4.1, initiating a new optimization causes the genetic algorithm to randomly assign the first generation with parameters upon which later generations can improve. One problem with this process, referenced in Section 4.2, is the processing time required to converge upon an optimized parameter set. During the collection of data for this thesis, it became necessary to interrupt optimizations before they converged and continue them at a later time.

A feature called a founder was implemented allowing the user to specify a parameter set in the algorithm settings file to be included in the first generation of optimization. Because the

particular parameter set is the result of a partially or fully converged optimization, it will almost always be evaluated as the fittest individual of the first generation and continue the optimization where it was interrupted before. The unlikely case that one of the remaining randomly generated parameter sets outperforms the founder is considered a happy accident and a natural result of the genetic algorithm process.

Chapter 5: Results & Discussion

This section covers results of optimizing the numerical model for a series of cases. The first case explores the results of attempting to optimize the model for forward flight at a variety of pitch angles. The second case presents results of varying the mass and flapping frequency of butterfly in stable forward flight.

5.1 Forward Flight Optimization with Varying Pitch

In this section, the results of attempting to optimize the numerical model for forward flight at steady altitude are presented. Three optimizations were performed at pitch angles of 10° , 30° , and 50° , respectively, setting the initial and objective pitches of each set to their respective angles as is discussed in Section 3.2. These optimizations were performed using the genetic algorithm outlined in Section 2.4 and the numerical model described in Section 3.1. For comparison, a randomly generated, unoptimized parameter set was run. The parameters of each set are listed in Table 5.1.

Table 5.1: Kinematic parameters of simulations run at various pitch angles

| Set | Flapping (deg) | | | Sweeping (deg) | | | Feathering (deg) | | | Abdomen (deg) | | | (Hz) | |
|------------------|----------------|----------|----------|----------------|------------|------------|------------------|------------|------------|---------------|------------|------------|-------|---------|
| | ϕ_A | ϕ_m | ϕ_p | θ_A | θ_m | θ_p | θ_A | θ_m | θ_p | γ_A | γ_m | γ_p | f^1 | f_a^2 |
| Random | 14.6 | 14.8 | 241.5 | 12.6 | 348.8 | 14.1 | 0.1 | -25.0 | 204.4 | 16.9 | 6.4 | 44.8 | 7.91 | 9.34 |
| 10° Pitch | 49.3 | 13.3 | 119.3 | 16.5 | -4.5 | 209.7 | 36.1 | -2.6 | 153.3 | 30.1 | -25.1 | 127.6 | 8.72 | 8.71 |
| 30° Pitch | 48.3 | 16.2 | 120.9 | 10.5 | -2.6 | 201.8 | 28.2 | -3.0 | 160.5 | 31.7 | -23.7 | 138.1 | 8.99 | 9.00 |
| 50° Pitch | 44.5 | 4.2 | 134.2 | 14.0 | -4.4 | 161.3 | 19.0 | -13.8 | 177.5 | 4.4 | 2.6 | 26.5 | 9.35 | 9.32 |

1: flapping frequency

2: abdomen frequency

Figure 5.1 shows the position of the butterfly's estimated center of mass relative to its initial position over 20 flapping cycles for each of the parameter sets generated by the optimization routines. Although the optimizations were only run for a total of 10 flapping cycles each, results do not appreciably change by optimizing for a greater number of cycles, as can be seen in Figure 5.3. The results are shown for 20 flapping cycles, illustrating more clearly the trend of each flight and assuring a steady state is reached.

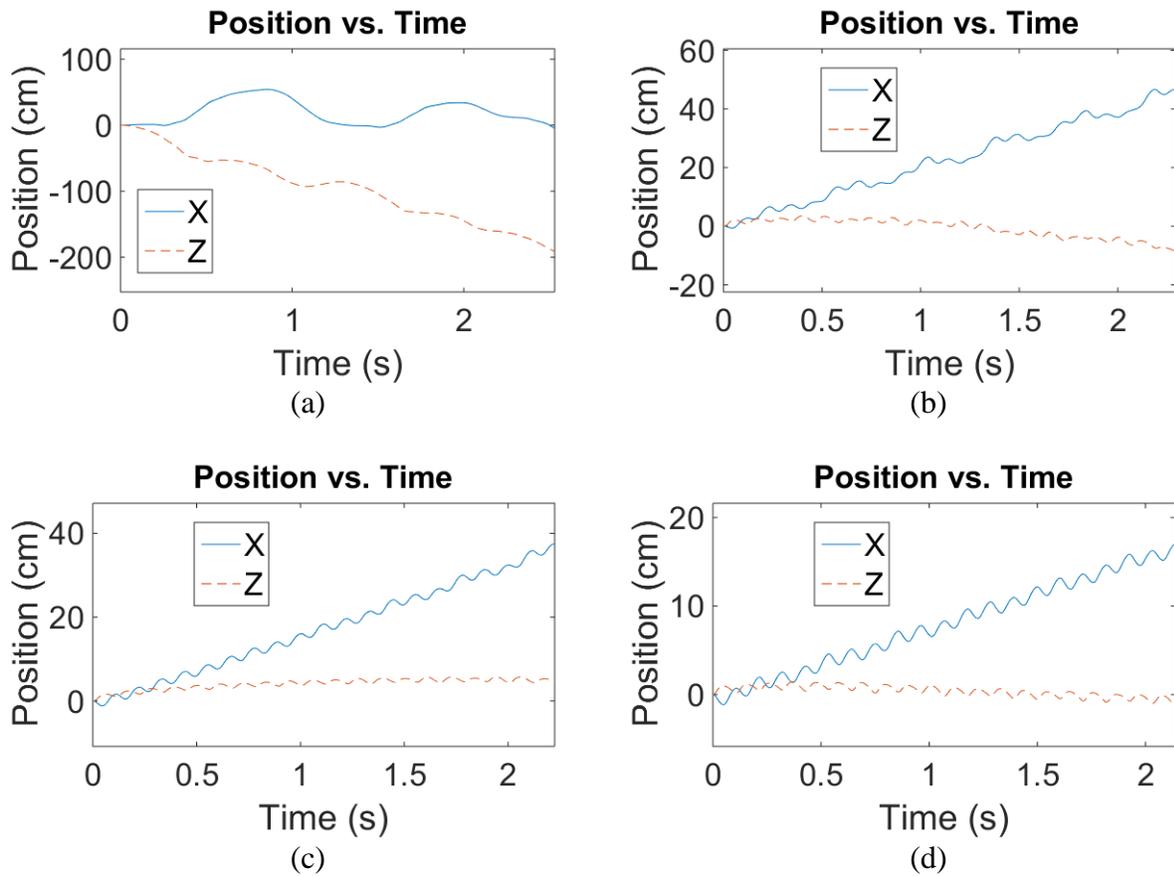


Figure 5.1: Positions over time for (a) a randomly generated parameter set, (b) a parameter set optimized at a pitch angle of 10° , (c) another set optimized at 30° , and (d) a set optimized at 50° .

The randomly generated parameter set does not reach a steady flight, and continually loses elevation. In the X-direction, it oscillates between approximately 0 and 50 cm with a net

velocity of almost 0 cm/s. In the Z-direction, it loses altitude relatively steadily at a mean descent rate of -75.5 cm/s.

The parameter set optimized at an angle of 10° loses altitude more slowly than the random solution and moves forward in a significantly more constant velocity. It does not exhibit regular body oscillations, however, and loses altitude at a rate almost 1/3 of its forward velocity. In the X-direction, the butterfly flies forward at a mean rate of 20.67 cm/s with body oscillations generally falling below distances of 6 cm. In the Z-direction, vertical velocity begins positive but reaches a relatively steady descent of about -6.5 cm/s after 1 s of flight.

The parameter set optimized at an angle of 30° displays more regular, oscillating overall forward movement. Instead of losing altitude, it actually rises slightly in the Z-direction at a mean rate of 2.04 cm/s with oscillations of about 1.2 cm. In the X-direction, it moves forward at a mean rate of 16.85 cm/s with vertical oscillations of about 3.5 cm.

The parameter set optimized at an angle of 50° behaves very similarly to the set at 30° , but flies forward about half as slowly. It also loses altitude slowly. In the X-direction, it moves forward at a mean rate of 7.96 cm/s with oscillations of about 2 cm, which are larger than those in the 30° set. In the Z-direction, it reaches a steady decline of about -1 cm/s after 0.5 s and oscillates vertically approximately 2 cm.

Figure 5.2 shows the pitches over time of each parameter set. The random set oscillates slowly between 0 and almost 300 degrees twice and then seems to only oscillate to 100 degrees. The 10° set pitches erratically between 0 and 100 degrees with oscillations of about 50 degrees. The 30° set pitches in a more even sinusoidal manner from about 25 to 96 degrees with oscillations of 70 degrees. The 50° set pitches similarly to the 30° set in a relatively even sinusoidal pattern between about 45 to 110 degrees with oscillations of approximately 60 degrees.

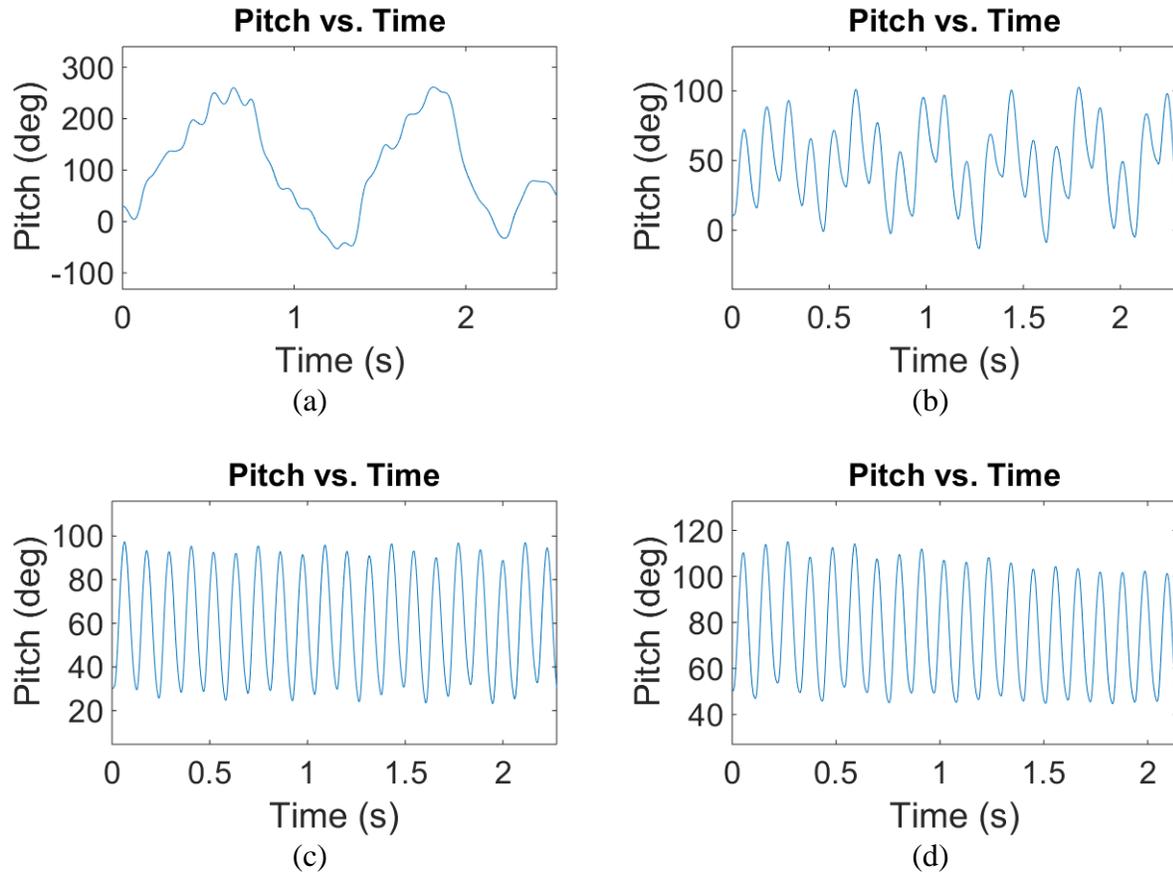


Figure 5.2: Pitch over time for (a) a randomly generated parameter set, (b) a parameter set optimized at a pitch angle of 10° , (c) another set optimized at 30° , and (d) a set optimized at 50° .

Comparing the randomly generated parameter set to those that were optimized using the genetic algorithm, it is apparent that optimization delivers improved results. Comparison of the flights of the parameters sets optimized at 10° , 30° , and 50° indicates that the model behaves in a more regular and steady manner when optimized at 30° and 50° . The flight of the 10° set does continue to fly in the same direction for at least 50 flaps, as can be seen in Figure 5.3, but its pitching during flight is erratic and is unlikely to represent realistic butterfly flight.

In contrast, the flights of the 30° and 50° sets exhibit behavior that more closely fits the expectation of steady flapping. While the model cannot account for a butterfly changing its kinematic parameters mid-flight, the results of the optimization could represent a butterfly in the

short-term. An inverse correlation was produced between forward velocity and pitch angle in the 10°, 30°, and 50° sets, resulting in slower forward flight for higher angles. This fits the prediction made in Section 2.3 that by pitching the butterfly forward, the lift vector will act more in the X-direction.

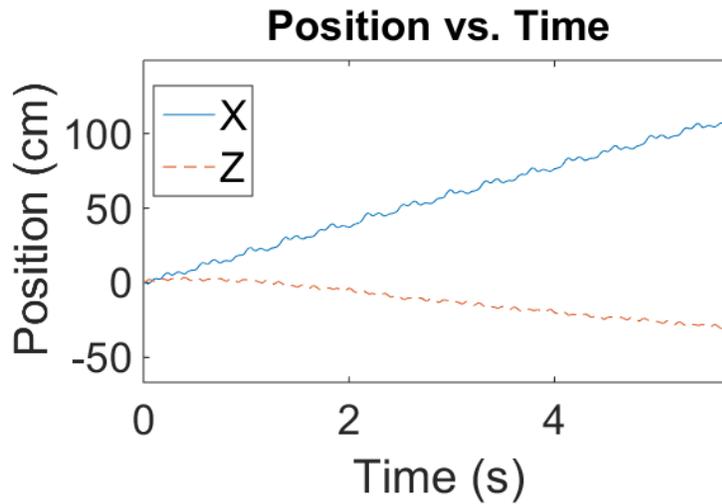


Figure 5.3: Position history of a parameter set optimized at a pitch angle of 10° for 50 flaps

Comparison of the three optimized parameter sets in Table 5.1 shows a trend in the flapping frequencies that is the opposite of what was predicted to occur. When pitching forward, the model is optimized to flap more slowly, which should produce less lift. The forward velocity, however, did not exhibit a trend that would be consistent with a decrease in thrust. In fact, pitching from 50° to 30° resulted in an increased velocity, implying that thrust was increased to account for its pitching forward. This trend could be due to the high mechanical power cost of hovering and the energetically favorable conditions slow flight features by reducing the need for continual balancing of up and down strokes [15].

5.2 Modified Flapping Frequency and Mass

5.2.1 Flapping Frequency

Figure 5.4 shows the effects of increasing and decreasing flapping frequency of a steadily flying parameter set. Flapping at 8.8 Hz resulted in a mean descent rate of -2.29 cm/s and a stable flight. Flapping at 9.0 Hz produced results almost identical to those in Section 5.1, which was expected. Flapping at a faster 9.2 Hz resulted in a flight that was no longer stable after about 1.8 seconds, but ascended at a mean rate of 8.10 cm/s.

The results of this experiment seem to imply that increased flapping frequency increases the aerodynamic lift of the butterfly, due to the direct relationship between flapping frequency and ascent rate. As frequency increased, however, the model became unstable. This is an indication of the complicated nature of the optimization problem being solved by the genetic algorithm. Any one solution for a given objective function is dependent on all 14 input parameters and manipulating just one without adjusting the other 13 accordingly can result in instability.

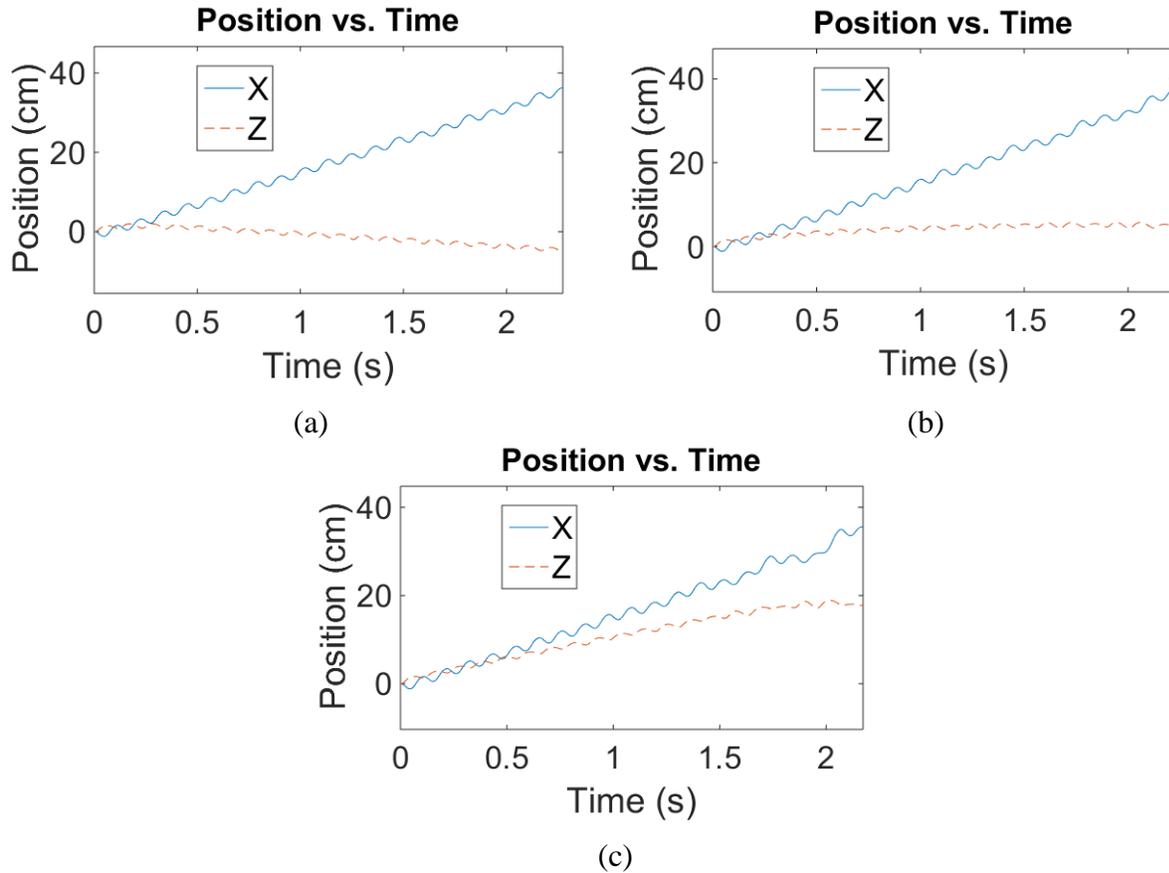


Figure 5.4: Positions over time of the set optimized at a pitch of 30° with (a) a flapping frequency of 8.8 Hz, (b) a flapping frequency of 9 Hz, and (c) a flapping frequency of 9.2 Hz

5.2.2 Mass

Figure 5.5 shows the effects of increasing and decreasing the body mass of the butterfly for the same parameter set used in Section 5.2.1. A butterfly with a mass scale factor of 0.95 flies at an ascent rate of 7.32 cm/s, but becomes unstable at 2 seconds. The unscaled butterfly with a factor of 1.00 flew as expected with an ascent rate of 2.04 cm/s. Increasing mass scaling factor to 1.10 resulted in a descent rate of -6.11 cm/s, and there was no loss in stability. These results show that increasing mass does make it more difficult for the butterfly to maintain altitude.

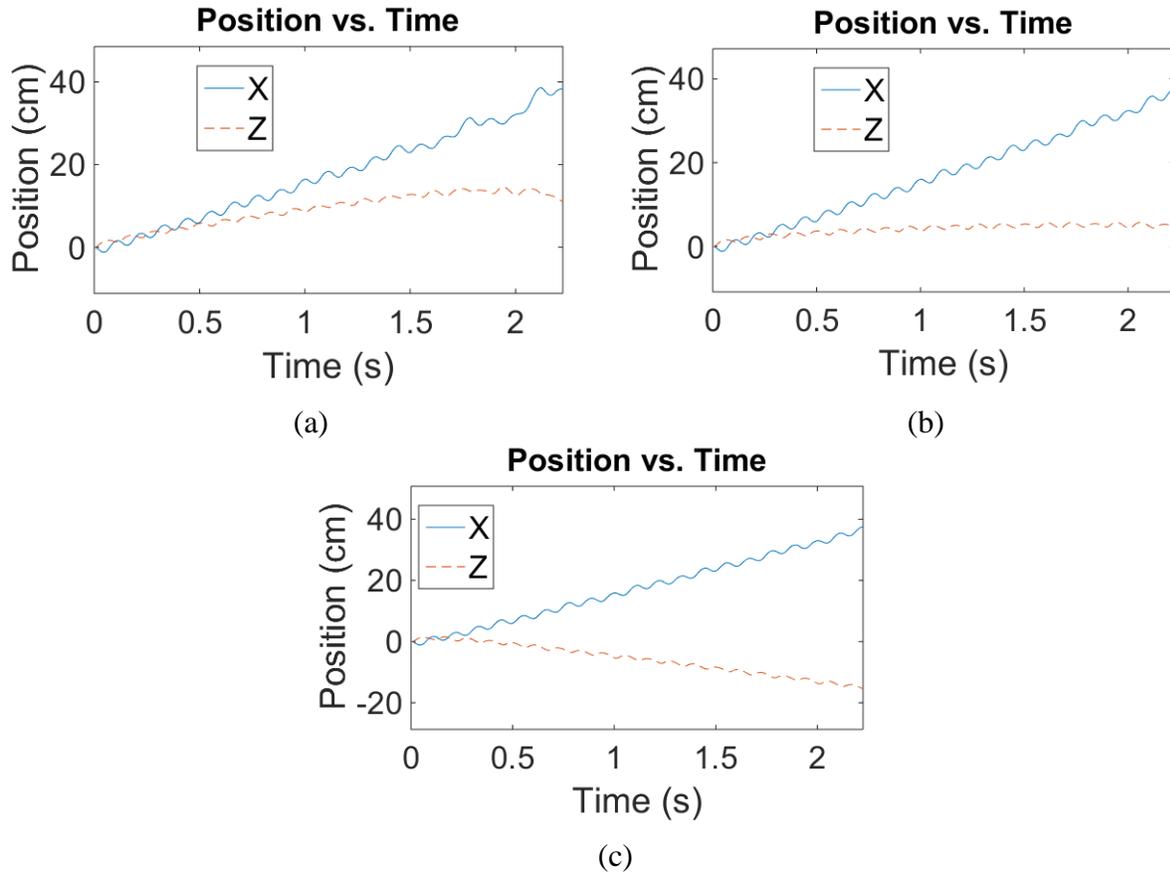


Figure 5.5: Positions over time of the set optimized at a pitch of 30° with (a) a mass scaling of 0.95, (b) a mass scaling of 1.00, and (c) a mass scaling of 1.10.

Chapter 6: Conclusions

This thesis sought to review and improve the numerical model and genetic algorithm optimization routine developed during previous work with the goal of producing stable forward flight in the butterfly simulation. The effects of pitch angle, flapping frequency, and butterfly mass on the stability and direction of forward flight in the numerical model were also studied.

Firstly, the numerical model was updated to a more user friendly state. This involved searching for and modifying older versions of files to replace missing ones, combining blocks of code from multiple versions of the model, and rewriting other parts. The optimization routine was improved to increase the rate at which results could be produced. The first improvement made was to implement parallel processing into the optimization process, allowing multiple, up to the number of logical cores available to MATLAB, parameter sets to be evaluated for their fitness concurrently, thereby reducing optimization times to almost half on a dual-core workstation and one third on a quad-core system. Next, the process of collecting results from the optimization routine into the actual simulation was streamlined by producing automatically generated results of the fittest individual of each successive generation. As part of this improvement to the transcription process, precision of each parameter set was increased from 5 places to the right of the decimal to 24, mitigating stacking errors in the ODE solver. Finally, because of the prohibitive length of optimization times, a method to continue or improve upon older optimizations was developed. This feature enabled the user to specify a parameter set as founder and include it as part of the population of the first generation of the current optimization.

Stable forward flight of the numerical model was produced by pitching the butterfly forward, then using the genetic algorithm to optimize for altitude and pitch angle maintenance. This resulted in flight patterns that oscillated, but moved forward at constant velocities over time.

Three pitch angles, 10° , 30° , and 50° , were tested. Flight at 10° exhibited the highest forward velocity at 20.67 cm/s, while the sets at 30° and 50° , which had forward velocities of 16.85 cm/s and 7.96 cm/s, respectively, flew progressively more slowly with increasing pitch. The set at 10° exhibited a different mode of flight that involved more irregular pitching patterns and a less regular forward oscillation in X-position. Results at 30° and 50° displayed more regular and sinusoidal movement, where the set at 30° moved forward more quickly, fitting predictions based on pitch angle. Flapping frequency did not increase with decreasing angle, however. The underlying mechanism behind this may be the power vs speed relationship observed in bats and other flapping fliers. While hovering and very fast forward flight are energetically unfavorable, forward flight at a moderate speed presents a minimum in the power vs forward velocity relationship [15].

To explore mechanisms of mitigating descent in the Z-direction during forward flight, flapping frequency and body mass were manipulated. A direct relationship between flapping frequency and upward velocity was found, although large deviations produced very unstable results. Increasing flapping frequency from 8.8 to 9.2 Hz increased ascent rate from -2.29 cm/s to 8.10 cm/s. Conversely, an indirect relationship was found between mass and ascent rate with the same behavior of instability at larger deviations from optimized values. Increasing the mass scaling factor from 0.95 to 1.10 increased ascent rate from -6.11 cm/s to 7.32 cm/s.

Chapter 7: Future Work

Recommendations for the continuation of this work are focused on the optimization process and the numerical model. Beginning with the optimization algorithm, the process of changing the objective function could be streamlined. Firstly, developing a user interface for the initial position, velocity, and environmental conditions would significantly improve the usability of the optimization routine. Additionally, a lot of commented code, or vestigial code, exists within the numerical model and optimization algorithm. A complete code refactor would make future edits simpler and reduce the learning curve of using the code.

To improve the numerical model, it is suggested to implement a method of decreasing downstroke speed to simulate forward velocity without pitch angle changes as opposed to tilting to increase forward velocity. Expanding on the dynamic parameters, the ability to change from one flight mode to another would be a valuable addition to the model. This would involve flapping with one parameter set for a time span and switching to another to simulate in-air maneuvering.

To improve the genetic algorithm, it is recommended to perform a study on its settings. There is no evidence that the current configuration is the most efficient method of converging upon results. Modifying settings of the genetic algorithm could significantly reduce optimization times or potentially the quality of results by preventing premature convergence or convergence upon local minima [16]. In addition, exploring the concept of linked genes, specifically feathering and flapping angles, within the context of the genetic algorithm could help the model more accurately simulate butterfly flight. Butterflies passively control their feathering angles, so changing the model to reflect this would be a huge improvement [17].

Bibliography

- [1] T. Wilson, *Theoretical modeling and experimental analysis of the flight mechanics of Lepidoptera*, Corvallis, Or: Oregon State University, 2013.
- [2] D. A. Reay, *The History of Man-Powered Flight*, Burlington: Elsevier Science, 2014.
- [3] A. Panaras, *Aerodynamic Principles of Flight Vehicles*, Reston: American Institute of Aeronautics and Astronautics AIAA, 1900.
- [4] M. Dickinson, "Insect flight," *Current Biology*, vol. 16, no. 9, pp. R309-R314, 2006.
- [5] S. Sane, "The Aerodynamics of Insect Flight," *Journal of Experimental Biology*, vol. 206(23), pp. 4191-4208, 2003.
- [6] D. Kolomenskiy, H. K. Moffatt, M. Farge and K. Schneider, "The Lighthill–Weis-Fogh clap–fling–sweep mechanism revisited," *Journal of Fluid Mechanics*, vol. 676, pp. 572-606, 2011.
- [7] L. A. Miller and C. S. Peskin, "A computational fluid dynamics of 'clap and fling' in the smallest insects.," *Journal of Experimental Biology*, vol. 208(2), p. 195, 2005.
- [8] S. Ho, H. Nassef, N. Pornsinsirak, Y.-C. Tai and C.-M. Ho, "Unsteady aerodynamics and flow control for flapping wing flyers," *Progress in Aerospace Sciences*, vol. 39(8), pp. 635-681, 2003.
- [9] S. P. Sane and M. H. Dickinson, "The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight," *Journal of Experimental Biology*, vol. 205(8), p. 1087, 2002.
- [10] J. Watkinson, *Art of the Helicopter*, Burlington: Elsevier Science, 2003.

- [11] S. Vogel, "Flight in *Drosophila*," *Journal of Experimental Biology*, vol. 44, pp. 567-578, 1966 .
- [12] X.-S. Yang, *Nature-inspired optimization algorithms*, Amsterdam: Elsevier, 2014.
- [13] R. Albertani, M. Goettl and T. Wilson, "A Wind Tunnel Investigation of Lepidopterae Flight in Cross-Wind Conditions," *51st AIAA Aerospace Sciences Meeting*, Grapevine (Dallas/Ft. Worth Region), Texas.
- [14] MathWorks, Inc., *MATLAB and Statistics Toolbox Release 2014b*, Natick, Massachusetts, United States.
- [15] U. M. Norberg and J. M. V. Rayner, "Ecological Morphology and Flight in Bats (Mammalia; Chiroptera): Wing Adaptations, Flight Performance, Foraging Strategy and Echolocation," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences (1934-1990)*, vol. 316, no. 1179, pp. 335-427, 1987.
- [16] M. Affenzeller, S. Wagner, S. Winkler and A. Beham, *Genetic Algorithms and Genetic Programming Modern Concepts and Practical Applications*, Hoboken: CRC Press, 2009.
- [17] N. Yokoyama, S. Kei, I. Makoto and H. Norio, "Aerodynamic forces and vortical structures in flapping butterfly's forward flight," *Physics of Fluids*, vol. 25, no. 021902, 2013.

