Neural Network Based Predictions for the Aerodynamic Performance of Flapping Wings

by

Olivia Chamberland

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements for the degree of Master of Applied Science in Aerospace Engineering Department of Mechanical and Aerospace Engineering Carleton University Ottawa, Ontario, Canada December 2021

Copyright ©
2021 - Olivia Chamberland
The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the Thesis

Neural Network Based Predictions for the Aerodynamic
Performance of Flapping Wings

Submitted by Olivia Chamberland
in partial fulfilment of the requirements for the degree of
Master of Applied Science

______________________________
Dr. J. Laliberté, Co-Supervisor

______________________________
Dr. F. Khouli, Co-Supervisor

______________________________
Dr. J. Dawson, Co-Supervisor

______________________________
Dr. R. Miller, Department Chair

Carleton University
2021
Abstract

There is unrealized potential in using Neural Network approaches in aerospace, particularly in assisting with aircraft design. The objective of this thesis was to determine if neural networks could predict the morphology, kinematics, and performance of flapping wings. A neural network was developed in Python and trained using a small dataset of biological insect data from literature. The model was then tested for biological data (small and large insects) as well as micro-aerial vehicles. The small insect test dataset performed best likely due to similarities with the training set. A larger dataset is needed to validate the use of neural networks in flapping wing micro-aerial vehicle design. This work is ground-breaking in the field of flapping wing micro-aerial vehicle design since it provides the foundation for a quick and accurate alternative to lengthy experiments and simulations.
Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisors Dr. Jeremy Laliberté, Dr. Fidel Khouli and Dr. Jeff W. Dawson for their continuous support. It is thanks to their patience, motivation, enthusiasm, and immense knowledge that I was able to complete this work. Thank you for taking a chance on some little physicist and turning her into the engineer she is today.

Thank you to my friends and family who have supported me throughout my graduate studies. A special thank you to my parents who have infused me with the passion and drive to accomplish anything I set my mind to. Although, you were sometimes a bit harsh on me, I know you have always wanted what was best for me.

When I first started grad school, I attended the teaching assistant orientation session. At this session they told me that the members of my office would become my strongest support system and they were absolutely right. To all the members of Minto 3041 past and present, thank you for providing me a shoulder to cry on. You have all played an important role in my time here at Carleton. I would especially like to thank Brendan Ooi for long nights spent in the office and Mila Kanevsky for the excursions and sanity checks.
Table of Contents

Abstract iii

Acknowledgments iv

Table of Contents v

List of Tables viii

List of Figures ix

1 Introduction 1
  1.1 Background ................................................. 1
    1.1.1 Micro-Aerial Vehicle Design .......................... 1
    1.1.2 Flapping Wing Aerodynamics .......................... 2
    1.1.3 Neural Networks ...................................... 6
  1.2 Research Objectives ........................................... 12
  1.3 Thesis Organization .......................................... 12

2 Literature Review 14
  2.1 Aerodynamics of Flapping Wings .............................. 14
    2.1.1 High Lift Mechanisms ................................ 15
  2.2 Machine Learning Applications ................................ 21
## 3 Methodology 23

3.1 Developing the Neural Network ........................................... 23
3.2 Data Collection .................................................................. 25
3.3 Data Preparation ................................................................. 27
3.4 Convergence Study ................................................................. 28
3.5 Biological Predictions and Case Studies ................................. 29
3.5.1 Statistical Analysis and Presentation of Results ................. 30
3.6 Impact of Dataset Size ............................................................ 31

## 4 Results and Discussion 34

4.1 Determining the Number of Iterations ................................. 34
4.2 Predictions ................................................................. 38
  4.2.1 Small Insects ................................................................. 38
  4.2.2 Training Loss Curves of Small Insects .............................. 46
  4.2.3 Large Insects ................................................................. 48
  4.2.4 Training Loss Curves of Large Insects .............................. 54
  4.2.5 Comparison Between the Smaller and Larger Insect Results 56
  4.2.6 Case Studies ................................................................. 56
  4.2.7 Training Loss Curves of MAVs ....................................... 64
4.3 Linear and Non-Linear Data ................................................... 66
4.4 Key Observations ................................................................. 67

## 5 Concluding Remarks and Future Work 69

5.1 Conclusions ................................................................. 69
5.2 Recommendations ............................................................. 69
5.3 Contributions ................................................................. 71

List of References 72
# List of Tables

4.1 Predicted Values of Small Insects ........................................ 40  
4.2 Comparison of Including and Excluding Negative Values from Predictions 42  
4.3 Predicted Values of Large Insects ........................................ 50  
4.4 Predicted Values of Case Studies on MAV ................................. 60  
4.5 The Effects of Datasets ..................................................... 67  
B.1 Small Insect Dataset ....................................................... 92  
B.1 Small Insect Dataset Continued ........................................... 93  
B.2 Large Insect Dataset ....................................................... 94  
B.3 Dataset for the Case Study on MAVs ..................................... 94
List of Figures

1.1 Terms used to define the geometry of a wing .......................... 3
1.2 Movement and camber changes of wings during one flapping cycle of MAV prototype wings (Zhang et al., 2019) .................. 4
1.3 The stroke plane (Ellington, 1984c) and stroke amplitude (Ellington, 1984c) of an insect where \(x, y, z\) is the local coordinate system and \(X, Y, Z\) is the global coordinate system. The local and global coordinate systems are set relative to the wing and body of the insect respectively. \(\beta\) denotes the angle at which the stroke plane is located from the horizon and \(\phi\) is the stroke amplitude .................. 4
1.4 Cross section of a leading edge vortex during the downstroke ......... 5
1.5 Supervised leaning ............................................................ 7
1.6 Rectified Linear Activation Function .................................... 8
1.7 A simple neural network in comparison to a deep neural network ... 9
1.8 The different types of data fitting ........................................ 11
1.9 Example of a training loss curve ......................................... 11
3.1 The path of data through the neural network ............................ 26
3.2 Example figure for prediction results .................................... 30
3.3 Three training functions used to determine the effects of data set size 32
4.1 Computational time as the number of iterations is increased ............ 35
4.2 The convergence of percent error(%) as the number of iterations increases 36
4.3 The change in percent error as the number of iterations increases. . . 37
4.4 Predicted values of area in comparison to their original values. . . . . 39
4.5 Predicted values of wingspan in comparison to their original values. . 39
4.6 Removing negative area predictions. ........................................ 41
4.7 Number of times a specific predicted value of area occurred. ........ 42
4.8 Predicted values of frequency in comparison to their original values. 44
4.9 Predicted values of stroke amplitude in comparison to their original values. ................................................................. 44
4.10 Predicted values of $C_L$ in comparison to their original values. ...... 45
4.11 Predicted values of mass in comparison to their original values. ...... 46
4.12 The training loss curves of small insects. .................................. 47
4.13 Predicted values of area in comparison to their original values. ...... 49
4.14 Predicted values of wingspan in comparison to their original values. 49
4.15 Predicted values of frequency in comparison to their original values. 51
4.16 Predicted values of stroke amplitude in comparison to their original values. ................................................................. 52
4.17 Predicted values of $C_L$ in comparison to their original values. ...... 53
4.18 Predicted values of mass in comparison to their original values. ...... 54
4.19 The training loss curves of large insects. .................................. 55
4.20 Northwestern Polytechnical University Dove (Yang et al., 2018). .... 57
4.21 Delft University micro-aerial vehicles. .................................... 57
4.22 Harvard University Robobee (Wood, 2008). ............................. 58
4.23 Predicted values of area for MAV in comparison to their original values. 59
4.24 Predicted values of wingspan for MAV in comparison to their original values. ................................................................. 59
4.25 Predicted values of frequency for MAV in comparison to their original values. ................................................................. 62
4.26 Predicted values of stroke amplitude for MAV in comparison to their
original values. .................................................. 62
4.27 Predicted values of $C_L$ for MAV in comparison to their original values. 63
4.28 Predicted values of mass for MAV in comparison to their original values. 64
4.29 The training loss curves of MAV. ........................................... 65
Chapter 1

Introduction

Neural Networks (NN) are useful for anything from object identification (Qiang et al., 2020) to predicting the dynamic stall of a helicopter (Faller and Schreck, 1996). This thesis describes their use for flapping wing aerodynamics and Micro-Aerial Vehicle (MAV) design. With the use of a NN, various parameters of flapping flight were predicted and compared to their true value. In addition, case studies were also performed to determine their feasibility for MAV design.

1.1 Background

1.1.1 Micro-Aerial Vehicle Design

Remotely Piloted Aircraft Systems (RPAS) consist of the aircraft itself and its control station. In Canada, RPAS are classified by their weight. When determining the weight of a Remotely Piloted Aircraft (RPA), the weight of the control station is not included, but does include any payload such as a camera. RPAS that weigh less than 250 g are called micro Remotely Piloted Aircraft (mRPA or MAV), and those weighing between 250 g and 25 kg are called small Remotely Piloted Aircraft (small RPA) (Canada, 2020).
RPA have exhibited all means of flight (fixed wing, rotary wing and flapping wing); however, for the purpose of this thesis, the focus will be on flapping wing MAVs. Currently, flapping wing MAVs are not commercially available unlike many other RPA, instead they are still confined to the realm of academic research and sometimes as a pastime for hobbyists. One of the famous examples is the RoboBee which was developed by Harvard University in 2007 (Wood, 2008). Other examples include the Nano Hummingbird (Keennon et al., 2012), the KU-Beetle (Phan et al., 2019) and the RoboFly (James et al., 2018).

Flapping wing MAVs have many potential applications other than being an academic curiosity, which includes surveillance, monitoring and intelligence gathering. Due to their biomimicry aspects, flapping wing MAVs are stealthy and less likely to be detected. They have the ability to be in closer proximity to the subject of surveillance than typical RPA.

1.1.2 Flapping Wing Aerodynamics

Birds and insects have evolved and used flapping wing systems for flight for millions of years. Studies show that flapping wing flight in nature is more efficient than traditional methods of flight (Shyy et al., 2016). Therefore, understanding the underlying aerodynamic mechanisms responsible for flapping wing flight is of primary importance. In recent years, the development of micro-aerial vehicles has renewed interest in developing a flapping wing flyer.

The principles underlying the aerodynamics of flapping wing flight are not fully understood. For example, various properties of the wing play an important role in flapping wing flight. Figure 1.1 depicts a planform view of a wing. The two commonly used terms which describe the dimensions of a wing are: wingspan and chord. The wingspan describes the length of the wing whereas the chord describes the width of the wing.
In addition to the wing geometry, the aeroelasticity of the wing also contributes to its aerodynamic performance. Aeroelasticity is a term that broadly refers to the complex reactions of materials and structures (e.g. wings) in the context of fluid interactions with flexible and solid bodies (Čečrdle, 2015). For the purpose of this research, aeroelasticity is the amount the wing deforms throughout one stroke of the flapping cycle (Figure 1.2). Figure 1.2, illustrates the top view of a wing deforming due to aerodynamic and inertial forces. The wing adopts a positive and negative camber during the upstroke and downstroke respectively. The amount a wing deforms depends on the aeroelastic properties of the wing (Wehmann et al., 2019): the elastic modulus, the moment of inertia, the torsion modulus and the shear modulus.

Two critical kinematic parameters for describing a flapping flight are stroke amplitude and frequency. The stroke amplitude is denoted by an angle (Figure 1.3). It is used to define the angle swept during the downstroke or upstroke. The stroke plane of flapping wings can either be nearly horizontal or vertical but must contain some angle to generate lift. Frequency is defined as the number of full flapping cycles completed per unit of time. It is usually conveyed in Hertz (Hz).
Figure 1.2: Movement and camber changes of wings during one flapping cycle of MAV prototype wings (Zhang et al., 2019).

Figure 1.3: The stroke plane (Ellington, 1984c) and stroke amplitude (Ellington, 1984c) of an insect where \( x, y, z \) is the local coordinate system and \( X, Y, Z \) is the global coordinate system. The local and global coordinate systems are set relative to the wing and body of the insect respectively. \( \beta \) denotes the angle at which the stroke plane is located from the horizon and \( \phi \) is the stroke amplitude.

In flapping flight, lift can be generated during both the upstroke and the down-stroke depending on the stroke plane (Alexander, 2002). Kinematics such as the clap-fling mechanism (Weis-Fogh, 1973; Ellington, 1984c), the attached Leading-Edge
Vortex (LEV) (Ellington, 1999; Dickinson and Götz, 1993; Dickinson et al., 1999), wing-wake interaction or wake capture (Dickinson et al., 1999; Shyy and Liu, 2007), added mass and rotational circulation have been used to explain how additional aerodynamic forces can be generated using unsteady aerodynamic mechanisms.

The complex airflow around a flapping wing is characterized primarily by the LEV (Phillips et al., 2015) stabilized by a span-wise flow along the leading edge of the wing. LEVs are known to form on thin wings with moderate aspect ratios (approximately 3), flapping at high angles of attack and low Reynolds numbers (order: $10^4$ or smaller) (Nabawy and Crowther, 2017). A LEV remains near the leading edge and does not grow with time. This phenomena causes the flow over the upper surface of the wing to separate at the leading edge but then reattach before the trailing edge. Since there is no vorticity generated at the trailing edge, the wings satisfy the Kutta condition for high angles of attack at which conventional stall would occur (Sane, 2003). Figure 1.4 illustrates a leading edge vortex.

![Figure 1.4: Cross section of a leading edge vortex during the downstroke.](image)

The clap-fling mechanism was first described by Weis-Fogh (1973) studying the hovering kinematics of *Encarsia formosa* (wasp). It is a flapping mechanism where the leading edge of the wings clap together at the end of the upstroke and remain in contact for a period of time. The wings are then flung open with the trailing edges still in contact. This generates circulation about each wing creating lift even at very
low Reynolds numbers. Most tiny insects such as wasps (Weis-Fogh, 1973; Ellington et al., 1996), diptera (Ellington, 1984c; Ennos, 1989) and thirps (Ellington, 1980; Santhanakrishnan et al., 2014) use the clap-fling mechanism during flight. Larger insects typically do not use the clap-fling mechanism except during take-off or when carrying a load (Marden, 1987).

The aeroelasticity of wings coupled with the varying kinematics is what makes flapping flight much more complex than traditional fixed wing and rotary wing flight. This area of research is rapidly expanding, but currently there are no equations that govern flapping flight without major simplifications (Dickinson et al., 1999; Wang et al., 2004; Taha et al., 2014; Lindhe Norberg, 2006; Burgers, 2016; Weis-Fogh, 1973). The only method to obtain accurate values for lift or coefficient of lift ($C_L$) of flapping wing flight is by empirical measurement or computational fluid dynamics (CFD).

1.1.3 Neural Networks

Neural Networks (NN) are inspired by the biological mechanisms contained within the human nervous system (Aggarwal, 2018). The computational units (neurons) in neural networks are connected to each other through weights, which serve a similar purpose as the strength of synaptic connections in biological organisms. Each neuron is scaled with a weight which ultimately affects the computed function of that neuron. Learning occurs by updating the weights for connecting neurons.

For learning, biological organisms require an external stimulus. In neural networks, the external stimulus is provided by a training set which can contain either images or a table of data. Although, training sets must contain both input and output parameters. Training sets are then fed through the neural network where learning occurs. There are three different types of learning: supervised, unsupervised and reinforcement learning (Sathya and Abraham, 2013). Supervised learning
(Figure 1.5) occurs after the data is passed through the network and the prediction is compared to the expected value. The error between the two is then passed back through the neural network to generate the local gradient for each neuron in each layer so that the weights can be updated. This computation is repeated until the network converges which is defined as the point at which the network has learned how to properly respond to a set of training patterns within some margin of error. Each combination of forward and backward pass is defined as an epoch. In contrast, the error between prediction and expected values is not calculated in unsupervised NNs. Instead, unsupervised NNs identify the pattern within the data and corrects accordingly. Reinforcement learning learns through trial-and-error interactions. In other words, the network learns from the result of an action. Reinforcement learning is commonly used for games and autonomous cars.

![Figure 1.5: Supervised leaning.](image)

In supervised learning, the adjustment of weights and biases is referred to as back-propagation. Back-propagation is the process in which NN feed the error back through the neural network denoted by the bottom half of the loop in Figure 1.5.
Bias is an additional term which helps shift the activation function to the left or right (Ganesh, 2020). Activation functions are used in neural networks to recognize complex relationships between data (Sharma et al., 2020), they are simply transfer functions used to obtain the outputs. Without activation functions, NN would only be linear regression models capable of representing one-degree polynomials. There are many types of activation functions: Linear, Sigmoid, Tanh, ReLU, Leaky ReLU, SoftMax to name a few. The Rectified Linear Activation Function (ReLU) is a non-linear function that looks and behaves like a linear function (Brownlee, 2019). Non-linear activation functions allow complex relationships in data to be determined. The rectified linear activation function is a combination of two linear functions which make the function non-linear (Figure 1.6). It also has the added benefit that all negative values return zero which allows neurons to be stimulated at different times rather than all at once. Neurons can be switched on or off increasing computation efficiency in comparison to the commonly used Sigmoid and Tanh functions (Goodfellow et al., 2016).

There are various layers to a neural network. All NNs consist of an input layer and an output layer, but only some possess hidden layers. Neural networks that possess hidden layers, which are classified as any additional layer between the input
and outputs layers are considered deep neural networks (Figure 1.7 (b)).

Figure 1.7: A simple neural network in comparison to a deep neural network.

Neural networks fall into two distinctive categories: regression and classification problems. Regression problems aim to determine the relationship between one or more
independent variables and a dependent variable. Neural networks have revolutionized regression problems and have enabled the ability to predict a numerical value for multi-variable regression. The intention of classification problems is to determine if an image, video, or value is a certain object or feature. Classification problems often require many hidden layers due to their complexity in comparison to regression.

A major challenge in neural networks is overtraining. Overtraining occurs when the network is too powerful for the current problem, which means that the model will overfit the data. Overfitting (Figure 1.8 (c)) is defined as the point at which the NN does not recognize the underlying trend in the data, but instead memorizes the data including the noise. This greatly reduces the performance of the test set. One method to avoid overtraining is by implementing early stopping. Early stopping is implemented by stopping training at the point when performance starts to degrade. Figure 1.8 (a) illustrates underfitting which occurs when the NN cannot recognize a pattern in data. Underfitting is typically the result of a model that is too simple. To see if a model is overfitting or underfitting, one can plot the training loss curves for the training and testing sets. Figure 1.9 depicts an example of a training loss curve, where the orange line represents the training set and the blue represents the test set. The dashed grey line denotes the point at which the network starts overtraining the test set. It is at this point that one would implement early stopping.
Figure 1.8: The different types of data fitting.

Figure 1.9: Example of a training loss curve.
Another important parameter that can affect the performance of a neural network is the learning rate. The learning rate is defined as the speed at which the weights and biases are updated. If they are updated too rapidly, the model can converge to a sub-optimal solution, but if they are updated too slowly the model may not converge. It also plays an important role in computation time; the lower the learning rate the more time it takes. A solution to setting a learning rate is to use an adaptive learning rate optimization algorithm such as Adam. Adam combines the best properties of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propogation (RMSProp) to provide an optimization algorithm that can handle sparse gradients on noisy problems. The algorithm calculates an exponential moving average of the gradient and the squared gradient to update the learning rate. Adam is well suited for problems with a lot of data and/or parameters (Kingma and Ba, 2017).

1.2 Research Objectives

Design work for MAV is still widely researched in academia due to the unknowns of flapping flight. Since there are no equations that govern flapping flight, researchers must compute lengthy simulations and/or experimental procedures to check the feasibility of their design. This research evaluates the use of a neural network to predict various parameters used in MAV design. It is meant to serve as a quick and accurate alternative to CFD and experimental work.

1.3 Thesis Organization

This thesis is organized into five chapters; a description of each chapter is provided in this section.

Chapter 1: Introduction - This chapter is intended to provide the reader with
an introduction to the information related to the topics covered in this thesis. This includes micro-aerial vehicles, the aerodynamics of flapping flight and neural networks. Additionally, this chapter outlines the motivations and objectives of the work performed as well as an overview of each chapter.

Chapter 2: Literature Review - This chapter presents the previous work conducted on flapping wing aerodynamics and the application of neural networks to predict flapping flight.

Chapter 3: Methodology - This chapter details the numerical experiments conducted for this research. It discusses the generation of datasets, the construction of the neural network used for this research and the studies performed. Additionally, it outlines the statistical analysis used and the presentation of data.

Chapter 4: Results and Discussion - The predictions from the neural network are summarized and discussed within this chapter. This includes the results from small insect, large insect, and MAV predictions. Additionally, it includes a convergence study as well as a study of the impact of dataset sizing and complexity.

Chapter 5: Concluding Remarks and Future Work - This chapter summarizes the work completed to date related to the use of neural networks for MAV design and proposes suggestions for future work.
Chapter 2

Literature Review

This chapter explores the previous studies conducted on the aerodynamics of flapping flight as well as the use of neural networks to predict aerodynamics. It includes both the experimental and simulation research conducted over the years.

2.1 Aerodynamics of Flapping Wings

The work of Ellington (1984a,b,c,d,e,f) propelled the research surrounding the aerodynamics of flapping wings. Before Ellington (1984a,b,c,d,e,f), Weis-Fogh and Jensen (1956) thought that most insects hovered according to quasi-steady principles. Their results were based on inadequate data and often proved that quasi-steady mechanics were not feasible but the margin of error on their results was far too large to be accepted as a conclusion. Although, through a series of papers, they were able to produce the most complete study of a desert locust, *Schistocerca gregaria*. They determined the kinematics of fast forward flight and the aerodynamic characteristics of the wings in steady motion (Weis-Fogh, 1956; Jensen, 1956). The results were then combined and compared to a locust flying in a wind tunnel (Jensen, 1956). In this case the results agreed well, but Cloupeau et al. (1979) studied the cyclic force measurements for *Schistocerca* and showed that the quasi-steady principles were not
suitable for this genus.

Ellington (1984a,b,c,d,e,f), re-examined the aerodynamics of hovering flight and found results that disagreed with Weis-Fogh and Jensen (1956). Ellington (1984b,c) presents full data sets for a variety of insects. Ellington (1984d) examined the different flight mechanisms which cannot be incorporated into the blade element theory presented in Ellington (1984a). A more generalized vortex theory which calculates mean lift for quasi-steady mechanisms in addition to unsteady mechanisms is then described by Ellington (1984e). Ellington (1984f) then combined the information presented in this series of papers to analyze flight mechanisms, their power requirements and other mechanical considerations of hovering flight. It is with this combination that he was able to show that quasi-steady assumptions failed to accurately predict the aerodynamic forces required for flight. Since this revelation, there has been great progress in the understanding of flapping wing flight.

### 2.1.1 High Lift Mechanisms

There are a few mechanisms that have been shown to generate high lift in the unsteady environments of flapping flight. Mechanisms such as the attachment of the Leading-Edge Vortex (LEV) (Ellington, 1999; Dickinson and Götz, 1993; Dickinson et al., 1999), delayed stall of the LEV and the interaction between the wing and the separated LEVs from a previous stroke (Dickinson et al., 1999; Shyy and Liu, 2007) and the clap-fling mechanism (Weis-Fogh, 1973; Ellington, 1984c) have extensively been researched. The following provides a brief summary of the results of this research and some of the factors that play a role in these mechanisms.

#### Leading-Edge Vortices

Dickinson and Götz (1993) showed that lift was enhanced by the presence of leading-edge vortices (LEV). They studied the aerodynamic forces of an airfoil started rapidly
at high angles of attack for Reynolds numbers (Re) within the range of a fruit fly (75-225). A slow decrease in $C_L$ after approximately 2-3 chord lengths of travel was observed. This led them to believe that the shedding of the LEV was slow at such low Re. They believed that this stall-delaying behaviour was more appropriate than the steady-state approximations since the wing only moved 2-4 chord lengths each half-stroke.

Ellington et al. (1996); van den Berg and Ellington (1997) demonstrated that the LEV did not shed during the downstroke. They studied model hawkmoth (*Manduca sexta*) wings in tethered forward flight using flow-visualization techniques. It was noted that the LEV did not shed in the transitional phases of the half-stroke and that there was a spanwise flow directed from the wing root to wing tip. They were also able to show that the LEV could produce enough lift for the insect weight by analyzing the momentum imparted to the fluid by the vortex wake. Liu and Kawachi (1998) preformed a study of the flow surrounding a hawkmoth’s wing using computational fluid dynamics (CFD) which validated the results of Ellington et al. (1996) and van den Berg and Ellington (1997).

The studies done by Dickinson and Götz (1993), Ellington et al. (1996) and van den Berg and Ellington (1997) indicated that delayed stall was the mechanism responsible for high-lift generation of some small and large insects. Usherwood and Ellington (2002a,b) studied various revolving wings from insects to quails. The authors concluded that the aerodynamic forces were maintained by the attachment of the LEV.

Dickinson et al. (1999) and Sane and Dickinson (2001) showed that the translational velocity varied in accordance to a trapezoidal function while studying the aerodynamic forces on a mechanical fruit fly wing. They observed large forces during the transition phase of the half-stroke, with a spike near the beginning and end of the half-stroke. Using CFD, Sun and Tang (2001); Ramamurti and Sandberg (2002)
achieved similar results to the experiment conducted by Dickinson et al. (1999). The authors suggested that the large peak force at the beginning of the half-stroke was caused by the interaction between the rapid translational acceleration of the wing and the wake left by previous strokes (Dickinson et al., 1999; Sun and Tang, 2001; Birch and Dickinson, 2001). Additionally, (Dickinson et al., 1999; Sun and Tang, 2001) suggested that the peak near the end of the stroke was due to wing rotation.

The results from these studies indicate that LEVs are a contributing factor to the high lift generation in flapping flight. It is not only the LEV itself, but the wing-wake interaction and prolonged attachment of LEV.

**Low Reynolds Number**

Reynolds number is defined as the ratio between inertial and viscous forces (Eqn. 2.1). A low Reynolds number is typically on the order of one whereas a high Reynolds number is on the order of ten thousand. For high Reynolds number, the viscous forces are small, and the flow is essentially inviscid. For low Reynolds numbers, the viscous forces within the fluid and acting on the body are significant and must be accounted for by

\[
Re = \frac{\rho u L}{\mu} = \frac{\text{Inertial Forces}}{\text{Viscous Forces}}
\]

where \(\rho\) is the density of the fluid, \(u\) is the flow speed, \(L\) is the characteristic linear dimension and \(\mu\) is the dynamic viscosity of the fluid.

Tang et al. (2008) studied the effects of Reynolds number (\(Re\)) in the range of ten to one thousand on the aerodynamics of hovering 2-D airfoils. At low Reynolds numbers, the authors found that the viscosity dissipated the vortex structures rapidly. This leads the forward and backward strokes to exhibit symmetric flow structures and aerodynamic forces. Asymmetric patterns were seen at high Reynolds numbers.
Liu and Aono (2009) examined the interactions between LEV, tip vortices and vortex ring structures. They studied model hawkmoths ($Re=6.3\times10^3$, $k=0.30$), honeybees ($Re=1.1\times10^3$, $k=0.24$), fruit flies ($Re=1.3\times10^2$, $k=0.21$) and thrips ($Re=1.2\times10^1$, $k=0.24$) in hover. Stronger LEVs were created by higher Reynolds numbers, but lower Reynolds numbers were able to maintain the attachment of the LEV for longer periods of time. Shyy and Liu (2007) produced similar numerical results. All these studies imply that the Reynolds number is a dominant parameter of the LEV structure.

**Wing Deformation**

The flapping wings of insects are known to be flexible (Wootton, 1981). The deformation of flexible flapping wings is shown to have a great impact on the aerodynamics of flapping wings although many of the above studies have assumed that the wings act as rigid flat plates rather than flexible plates subject to deformation. Young et al. (2009) conducted a study using high-speed photogrammetry data published by Walker et al. (2009) which showed that camber and twist can enhance the aerodynamic efficiency of forward flight locusts. Similar results were achieved for beetles and butterflies (Le et al., 2013; Zheng et al., 2013). Additionally, Du and Sun (2010) showed that flexible wings produce approximately 10% more lift than rigid wings with approximately 5% reduced power requirements while studying numerical models of hovering flies. Vanella et al. (2009) obtained similar results while observing a chordwise deformation of a two-dimensional wing. They found that the lift-to-drag ratio increased by 28% in comparison to a rigid wing.

More recently, Nakata and Liu (2012) observed the fluid-structure interaction of flapping wings and noted that wing deformation prolongs the life of LEVs. This adjusts the phase of rotation allowing the generation of higher aerodynamic forces in comparison to rigid, flat wings. Nakata et al. (2018b) continued these studies by observing the effects of fluid-structure interactions under varying vacuum conditions.
They found that the density does not play a significant role; it is with the adjustment of wing kinematics that aerodynamic forces and efficiency can be enhanced. Nakata et al. (2018a) also noted that excessive deformation can reduce the aerodynamic efficiency significantly.

Other research surrounding wing deformation stems from the idea that Wootton (1991) believed that the vein pattern, curvature and joints plays a role in the camber of the wing under aerodynamic loading. Wootton et al. (2000) explored the deflections in the hindwing of a locust in three sections. Combes and Daniel (2003) then refined these results and demonstrated the flexural stiffness patterns amongst 16 insect species by applying point forces on the wings in both the spanwise and chordwise directions. The spanwise flexural stiffness in all tested species ranged from $10^{-4}$ to $10^{-6}$ Nm$^2$ and $10^{-5}$ to $10^{-7}$ Nm$^2$ for the chordwise flexural stiffness.

These studies show the impact of wing deformation in flapping flight. Typically, wing deformation aids lift, but naturally there is a point at which the amount of deformation becomes disadvantageous. A balance between stiffness and deformation must be achieved. Additionally, it is important to note that deformation can prolong the attachment of the LEV which has previously been shown to generate high lift.

**Aerodynamic Modelling**

More recent research on flapping wings has focused on generating models and including the added complexity of deformation. Some efforts have developed theoretical models for the fluid-structure interactions of flapping wings in hopes of understanding the underlying physics without the time consumption of solvers. Alben (2008, 2009) made use of potential flow theory by coupling a vortex sheet wake with a structural beam. Moore (2014, 2015, 2017) suggested a model where the coupling is based on pressure instead of vorticity which was less computationally expensive. Flexible wings were shown to reach maximum thrust at their resonant frequency, although a
higher drag (relative to lift) was exhibited once above their resonant frequency. These results agreed with the numerical simulations done by Michelin and Llewellyn Smith (2009). More recently, Medina and Kang (2018) developed an analytical solution for lift, thrust and required input power of an elastic plate and Alon Tzezana and Breuer (2019) extended the work of Moore (2014, 2015, 2017). Alon Tzezana and Breuer (2019) explained that thrust reaches a maximum at resonant frequency for relatively rigid membranes, but flexible membranes may exhibit drag at resonant frequency. The transition from thrust to drag is dependent on natural frequency which is characterized by the peak in maximum deflection and the aerodynamic coefficient.

Simultaneously, an analytical model for chordwise flexible flapping in hover was developed. Kang and Shyy (2014) suggested a model that accounts for acceleration and aerodynamic damping forces by incorporating the Morison equation. The result was an instantaneous wing deformation and lift model. Kodali and Kang (2016) noticed the lack of a model for chordwise flexible flapping wings in forward flight and were able to produce such a model. The model was obtained under the assumption that the passive pitch due to wing deformation is an effective angle of attack. Kodali et al. (2017) then produced an analytical model for spanwise flexibility in forward flight. These results were compared to the numerical results of Kang et al. (2011); Aono et al. (2009); Gordnier et al. (2010) and agreed well.

Although these models have aided in our understanding of fluid-structure interactions such as the aerodynamic efficiency surrounding resonant frequencies, there is currently no model that accounts for the various aerodynamic coefficients in terms of wing geometry.
2.2 Machine Learning Applications

Over the years researchers modelling flapping wing aerodynamics have moved from quasi-steady assumptions to full Navier-Stokes simulations. These simulations are rather time consuming and restricted due to the difficulties modelling the complex three-dimensional flow field and low Reynolds number regime. More recently, the study of machine learning has emerged and offered a more rapid solution to the complexity of these problems. Machine learning has been used to reduce computational time of CFD simulations (Kochkov et al., 2021). One application of particular interest in machine learning is the use of Neural Networks (NN).

Previous research implemented a NN to interpolate aerodynamic coefficients of an airfoil (dos Santos et al., 2008). A Multilayer Perceptron (MLP) network was created to predict drag polar curves for varying Reynolds number at a given Mach number. Other examples include the work of Steck et al. (1997); Soltani et al. (2007); Kurtulus (2009). Steck et al. (1997) predicted the forces and moments of various dynamic systems whereas Soltani et al. (2007) predicted unsteady aerodynamic loading. However, Kurtulus (2009) studied the unsteady aerodynamic forces of a low Reynolds number flapping airfoil. A three-layer feed forward network which included one input layer, one hidden layer and one output layer was implemented. In order to train the NN, the Levenberg Marquardt back-propagation algorithm was used. Inputs included the flapping airfoil displacement, translational velocity, angle of attack and angular velocity. The resulting outputs were the coefficients of lift and drag with a determination coefficient greater than 0.99. Additionally, Adique et al. (2010) proposed the use of NN for low Re airfoils flapping in a figure eight motion. They were able to successfully predict the lift coefficients with sufficient accuracy.

There is no doubt that NN are credible algorithms when trained correctly. Although considerable research has been conducted on the abilities of NN to predict
aerodynamics, there is still a gap in research. Many of the above studies do not incorporate flapping airfoils and those that do (Kurtulus, 2009; Adique et al., 2010), make use of rigid airfoils which have been shown to differ significantly from traditional flexible wings. Additionally, their results are based off CFD simulations and not experimental values. This thesis proposes the use of NN to predict various parameters (wingspan, wing area, wingbeat frequency, wing stroke amplitude, $C_L$ and mass) of flapping flight using previously published experimental data. The data therefore includes the underlying relationships of deformation, which the other studies do not incorporate.
Chapter 3

Methodology

This chapter outlines the experimental procedure used for this research. It explains
the construction of the NN, how datasets were generated and various tests that were
conducted using the datasets and NN.

3.1 Developing the Neural Network

Originally, the NN was created in MATLAB 9.7, but was later transitioned to Python
3.7 due to limitations associated with MATLAB. MATLAB offers a machine learning
toolbox at an added cost while Python offers free packages such as TensorFlow and
Keras. TensorFlow is an open-source library for various machine learning tasks where
as Keras is a neural network library that was built atop of TensorFlow2. Another
commonly used Python library that offers machine learning is scikit-learn which was
built on NumPy, SciPy and matplotlib. Although scikit-learn offers machine learning,
their regression NN does not support multivariable regression. It is for this reason
that TensorFlow and Keras were implemented for the purpose of this research. Keras
was used to generate the neural network itself whereas TensorFlow was sourced to
define some of the parameters such as the learning rate and the number of epochs.
The Python code of the NN used in this research is presented in Appendix A. It is a
two-layer neural network that uses one output and six input variables. The following paragraphs explain the rational behind the specifications used in the neural network; unless otherwise specified they were determined by trial and error.

The activation function chosen was the Rectified Linear Activation Function (ReLU). The main advantage of using ReLU over a sigmoid function is that it does not activate all the neurons at the same time. Neurons are deactivated when the output layer of the linear transformation is less than zero. Since not all neurons are activated at the same time, ReLU is much more computationally efficient than other activation functions such as the sigmoid and tanh functions (Goodfellow et al., 2016). Additionally, ReLU is a non-linear activation function that consists of two linear segments. Linear activation functions are easily trained but incapable of predicting complex relationships. The addition of the two linear functions allows for the complexity while also maintaining the ease of training that accompanies linearity. It is for these reasons that ReLU was chosen opposed to the commonly used sigmoid function.

To further increase computational efficiency, a technique called early stopping was implemented. Early stopping stops the number of epochs (training) if there is no improvement of a given value after a certain number of iterations. For the purpose of this research, early stopping was defined so that if the loss did not decrease by 0.001 for 3 iterations, the epoch would stop. These values were determined through trial and error. The ability to stop epochs when no improvement is made not only reduces unnecessary computation, but also reduces overtraining (Goswami, 2020).

Another key feature of this neural network is the use of Adam as an optimizer. Adam is an adaptive learning rate method that combines the best properties of the Adaptive Gradient Algorithm (AdaGrad) and the Root Mean Square Propagation (RMSProp). AdaGrad uses a separate learning rate for each variable which performs well with sparse gradients whereas RMSProp also has a learning rate for each variable
but adapts them based on the average gradients of the weights which performs well on noisy data. Adam computes an exponential moving average of the gradient and the squared gradient to update the learning rate. It is known to be particularly useful for NNs containing a lot of data and/or parameters (Kingma and Ba, 2017) which is why it was chosen for this network. Containing only 43 datum, the NN used in the present research lacks a large training set. The author would have ideally obtained empirical data for training the NN but this was not possible due to the lack of laboratory access because of the COVID-19 pandemic.

With the use of the NN outlined in this section, predictions can be made for various parameters of flapping flight. In order to make predictions, data must first be collected, prepared and then passed through the NN (Figure 3.1). Afterwards, it must be unscaled and then statistical analysis can be performed.

### 3.2 Data Collection

Datasets were generated after an extensive literature review. Before the COVID-19 pandemic hit, experiments were supposed to be conducted to obtain a dataset, but due to limited lab access during the pandemic, experiments could not be conducted so instead previously published biological data was used. During the literature review, both MAV and biological data were considered. There was not enough published data on MAV to be used as a training set during this study. Biological data was used as the training set, although both biological and MAV data was used as testing sets.

Published biological data was categorized based on insect size: small and large insects. Insects were further classified by their flight characteristics and size. Only 43 small insects datum and 4 large insects datum met our criteria which can be seen in Appendix A. To be used for the purpose of this research, data had to have the following parameters: coefficient of lift ($C_L$), wing stroke amplitude, wingbeat
Figure 3.1: The path of data through the neural network.
frequency, wingspan, and mass. Other parameters such as the number of wings and area were also sought, but not necessary because they could usually be inferred.

The datasets shown in Appendix A are the first of their kind, it is the largest compilation of wing geometry and lift data to currently exist. Additionally, the data sourced was scarce since not much new experimental data on wing geometry and lift has been generated. Researchers have been reusing the same datasets so in order to generate the datasets in Appendix A, data had to be traced back to it’s original source.

3.3 Data Preparation

The performance of the NN improves when the data is scaled to lie between 0 and 1 (Stöttner, 2019). Input variables often have different scales, but by scaling the data, they can be transformed to the same scale. Without scaling there is an increase in complexity which often renders the NN unstable (Brownlee, 2020a). Before the data was passed through the neural network it was normalized using the min-max scaling method (Patro and sahu, 2015). This technique transforms the minimum value to 0 and the maximum value to 1. The rest of the data falling between the scaled minimum and maximum. This approach was used on each input feature separately to maintain the relationships between input variables. They were calculated as follows:

$$S(x_i) = \frac{(x_i - x_{\text{min}})}{(x_{\text{max}} - x_{\text{min}})}$$ (3.1)

where, $S(x_i)$ is the scaled value for the $i$th value of the input variable $x$, $x_i$ is the unscaled value for the $i$th value of the input variable $x$, $x_{\text{min}}$ is the minimum value of input variable $x$ and $x_{\text{max}}$ is the maximum value of the input variable $x$.

Since the data was scaled before being passed through the neural network, it must also be unscaled before conclusions are drawn and statistical analysis is performed.
This is accomplished through Eqn. 3.2 such as,

\[ U_{p_i} = p_i(x_{\text{max}} - x_{\text{min}}) + x_{\text{min}} \]  

(3.2)

where \( U_{p_i} \) is the transformed prediction, \( p_i \) is the predicted value produced by the neural network, \( x_{\text{max}} \) is the maximum value of the prediction variable prior to passing through the neural network and \( x_{\text{min}} \) is the minimum value of the prediction variable prior to passing through the neural network.

### 3.4 Convergence Study

Before running various prediction trials, a convergence study was performed to determine a sufficient number of iterations. A sufficient number of iterations was defined as the point at which the NN converged. Although this is not standard practice as the number of iterations is generally a preference, it was performed to decrease computation time.

Choosing the coefficient of lift as a test output variable, three test points were selected. The test points were chosen so that the full range of data was encompassed. Therefore, three coefficient of lift data points were selected that consist of the highest, the lowest and the middle from the data set.

Once the test points were selected, the number of iterations was varied. For each trial, graphs showing the predicted values and confidence intervals were produced. Additionally, the percent errors were calculated and the total time to run a given number of iterations was recorded. With this data, plots of the computation time versus the number of iterations and the percent error versus the number of iterations could be created. The percent error was then further analyzed to determine the change in percent error per iteration (slope) using the following:
\[
\text{slope} = \frac{y_2 - y_1}{x_2 - x_1}
\]

(3.3)

where, \(y_2\) and \(y_1\) are the percent errors and \(x_2\) and \(x_1\) are the number of iterations.

The number of iterations was then plotted against the slope. This analysis was repeated for the frequency and area. Subsequently, all the plots in Section 4.1 were analyzed, and a number of iterations was selected such that the slope of the percent error had converged and time was kept to a minimum.

Coefficient of lift, frequency and area were chosen as the predicted variables since they each fall under a different category of wing parameters. Coefficient of lift was chosen for performance, frequency for kinematics and area for wing geometry. This was to ensure convergence across the different categories of parameters.

### 3.5 Biological Predictions and Case Studies

With the use of the data collected, biological and MAV data were predicted. Firstly, six data points were randomly selected from the small insect dataset. These randomly selected data points were then predicted using the small insect dataset as a training set (with the test points omitted). The wingspan, area, stroke amplitude, frequency, \(C_L\) and mass were each predicted independently. The entire small insect dataset was then used to predict the same parameters using the large insect and MAV datasets as testing sets. This resulted in a total of 18 predictive trials. Additionally, training loss curves were plotted for each of the 18 trials to ensure that overtraining was not occurring.
3.5.1 Statistical Analysis and Presentation of Results

The results for the predictions were then analyzed to confirm their accuracy. Typically, NNs use precision, recall and F1 to calculate their performance, but this is only useful for classification networks. Regression NNs must resort to other means to monitor their performance. For the purpose of this thesis, percent error was chosen to determine the accuracy of predictions.

The percent error ($PE$) was calculated using the following:

$$PE = \left( \frac{\overline{P}_i - x_i}{x_i} \right) \times 100$$  \hspace{1cm} (3.4)

where $\overline{P}_i$ is the mean of the unscaled predicted value and $x_i$ is the original/true value.

![Figure 3.2: Example figure for prediction results.](image)

Plots for each of the 18 predictive trials were generated to convey the results. An example of a plot is above below in Figure 3.2. Two predictions are being made for the stroke amplitude (output variable). The grey points represent the predictions.
from each of the 600 iterations whereas the orange points represents the mean of the predictions. It is important to note that the mean of the 600 iterations was chosen as the predicted value for the purpose of this thesis. Additionally, the error bars shown in orange convey the 95% confidence interval and the blue line represents the 1:1 ratio. In other words, a predicted point that falls directly on the blue line is equal to the original value. For results that were closely clustered together, additional insert plots were added.

The 95% confidence interval was calculated using scipy.stats 1.7.0 in Python. It is a statistical package designed by SciPy which allows the user to perform statistical analysis of distributions. For the purpose of this thesis, spicy.stats was not only used to calculate the 95% confidence interval, but also to plot histograms (Figure 4.7) and calculate skewness and kurtosis. This was executed to explain the asymmetrical error bars displayed in some of the results.

### 3.6 Impact of Dataset Size

It is established that the size of the dataset plays an important role in the accuracy of the results generated by the neural networks (Bataineh and Marler, 2017; Smyczyńska et al., 2016). Neural networks are not only dependent on dataset size, but also the number of parameters within the dataset and the amount of noise. The proposed numerical study removes noise and maintains a consistent amount of parameters in order to view the true impact of the size of a training set. Additionally, the relationship between the complexity of a dataset and its accuracy are observed by testing functions of increasing nonlinearity.

To examine the effect of data complexities, the performance of the NN was tested using three known mathematical functions with increasing level of nonlinearity. Starting from a linear data set, the linear plane equation presented in Eqn. 3.5 and plotted
in Figure 3.3 (a) is tested first,

\[ y = x + z \]  \hspace{1cm} (3.5)

Figure 3.3: Three training functions used to determine the effects of data set size.

Subsequently, two non-linear data sets were generated; the first being a polynomial (Figure 3.3 (b)) expressed in Eq 3.6 and the second being a more complex rational
algebraic function (Figure 3.3 (c)) expressed in Eq 3.7.

\[ y = x^2 + z^2 \]  \hspace{1cm} (3.6)

\[ y = \frac{(x^2 - 1)(z^2 - 4) + x^2 + z^2 - 5}{(x^2 + z^2 + 1)^2} \]  \hspace{1cm} (3.7)

The datasets were randomly split into testing and training sets during each trial. The program was written such that 20% of the total dataset was used for testing and the other 80% was used for training. These values were selected as they are commonly accepted across neural networks, but it is generally a user preference (Brownlee, 2020b). The datasets were then passed through the neural network separately with varying dataset sizes while maintaining a constant number of iterations. The average number of epochs \( \overline{\text{Epoch}} \) and percent error \( \overline{\text{PE}} \) was calculated for each trial.

Since there is a limited amount of data that exists for flapping flyers, this study was performed to highlight the need for more experimental data to improve the results.
Chapter 4

Results and Discussion

This chapter discusses the results obtained through the use of the NN. It explores the promising results from the convergence study, the use of biological data and MAV data, the effects of dataset size and then proceeds to analyze those results. All biological and MAV predictions are made using the small insect data as a training set due to limitations in data availability. Additionally, it illustrates the versatility of using NNs for future flapping wing flight studies.

4.1 Determining the Number of Iterations

A convergence study was performed to determine the ideal number of iterations to be used for this analysis. Although the number of iterations is usually a user preference, the study was still performed in order to gain a deeper understanding of how increasing the amount of data affects the predictions of the NN. This study explores the relationships between the number of iterations, percent error and time.
Figures 4.1 (a), (b) and (c) denote the relationship between time and the number of iterations for coefficient of lift, frequency and area respectively. Expectedly, the processing time increases linearly with the number of iterations. All three output variables show a linear increase in computational time as the number of iterations increases.
Figure 4.2: The convergence of percent error(\%) as the number of iterations increases.

The above Figures 4.2 (a), (b) and (c) show the relationship between the number of iterations and percent error for coefficient of lift, frequency, and area respectively. The legend provided within each plot shows the original value of the predicted variable. All test points demonstrate convergence with numerical noise, which warranted further investigation. Additionally, it should be noted that the extreme values display a wider range of percent errors and take longer to converge in comparison to the middle values.
Figures 4.3 (a), (b) and (c) convey the secondary analysis performed on the percent error for coefficient of lift, frequency, and area respectively. These plots are the result of calculating the slope between the percent errors as described in Section 3.4. The plots for both coefficient of lift and frequency resemble that of a damped harmonic response. The results for area (0.63 mm) are much more inconsistent and have a larger change in percent error. All examples converged around a slope of zero indicating little to no change.

After close inspection of all figures within Section 4.1; the number of iterations was determined to be 600 since most of the convergence occurs prior to 600 iterations. Although more iterations lead to an increase in precision, it also comes with additional
computational time. For the purpose of this research, such increased precision was deemed unnecessary.

4.2 Predictions

For this numerical experiment, published data were categorized into two categories, smaller and larger insects. Using this data, two numerical experiments were performed. The first uses a subset of the data shown in Table B.1 as a testing set. The second uses the full Table B.1 as a training set and Table B.2 as a testing set. Additionally, MAV data (Table B.3) was also tested using the small insect data as a training set.

4.2.1 Small Insects

In this section, six test points were selected at random from Table B.1. They were then passed through the NN so that predictions could be made for a variety of variables. These include predictions of wing geometry parameters, wing kinematic parameters, performance values and mass.

Wing Geometry

Two wing geometry parameters were investigated: the planform wing area and the wingspan. It is important to note that when discussing the results the variable being discussed is the output variable and all other parameters are input variables for the trial. The results for the area predictions can be seen below in Figure 4.4. This figure depicts the predicted values in comparison to the true values (as determined from literature). The mean value of the 600 predicted values has been adopted as the predicted value for statistical analysis. Table 4.1 denotes the predicted values and their percent errors from each of the trials in Section 4.2.1. For wing area, the
two extremety points have the largest percent error. The rest of the error values fall under 6%.

Figure 4.4: Predicted values of area in comparison to their original values.

Figure 4.5: Predicted values of wingspan in comparison to their original values.
Table 4.1: Predicted Values of Small Insects

<table>
<thead>
<tr>
<th>Area</th>
<th>Wingspan</th>
<th>Frequency</th>
<th>Stroke Amplitude</th>
<th>$C_L$</th>
<th>Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$U_p$ PE</td>
<td>$x$</td>
<td>$U_p$ PE</td>
<td>$x$</td>
<td>$U_p$ PE</td>
</tr>
<tr>
<td>(mm$^2$)</td>
<td>(mm$^2$) (%)</td>
<td>(mm)</td>
<td>(mm) (%)</td>
<td>(Hz)</td>
<td>(Hz) (%)</td>
</tr>
<tr>
<td>4.26</td>
<td>4.43 3.99</td>
<td>2.44 3.15 29.1</td>
<td>188 190 1.06</td>
<td>128 132 3.13</td>
<td>1.55 1.58 1.94</td>
</tr>
<tr>
<td>4.21</td>
<td>4.29 1.90</td>
<td>2.41 3.05 26.6</td>
<td>194 193 0.515</td>
<td>133 134 0.752</td>
<td>1.5 1.57 4.67</td>
</tr>
<tr>
<td>4.35</td>
<td>4.32 0.690</td>
<td>2.49 2.98 19.7</td>
<td>186 197 5.91</td>
<td>144 139 3.47</td>
<td>1.57 1.61 2.55</td>
</tr>
<tr>
<td>1.36</td>
<td>2.29 68.4</td>
<td>2.02 2.72 34.7</td>
<td>254 222 12.6</td>
<td>150 147 2.00</td>
<td>1.59 1.66 4.40</td>
</tr>
<tr>
<td>4.33</td>
<td>4.57 5.54</td>
<td>2.48 2.96 19.4</td>
<td>212 199 6.13</td>
<td>147 139 5.44</td>
<td>1.3 1.52 16.9</td>
</tr>
<tr>
<td>27.7</td>
<td>33.6 20.9</td>
<td>13.7 12.4 9.49</td>
<td>145 168 15.9</td>
<td>104 112 7.69</td>
<td>2.08 1.82 12.5</td>
</tr>
</tbody>
</table>
The results for wingspan can be seen in Figure 4.5 and Table 4.1. The largest wingspan had the smallest percent error (9.49%) and the smallest wingspan had the largest percent error (34.7%). The predictions could be improved by altering the dataset. The definition of wingspan varies throughout literature, some measure the wingspan from tip to tip and others measure it from root to tip. During the construction of the datasets, these factors were taken into consideration, and although attempts were made to minimize such errors, some discrepancy in the data was inevitable. The error bars associated with these results contained the predicted value but were non-symmetrical.

![Figure 4.6: Removing negative area predictions.](image)

It is important to note that negative wingspan and area values are not physically possible. They are the result of pre-scaling the dataset. Since the data is scaled between 0-1, the NN believes it is predicting only positive values, but since the data must be unscaled before it is usable after passing through the NN, negative values can occur. When negative values are removed (Figure 4.6), it shifts the mean of the prediction upwards. This results in worse predictions for overpredicted values.
and better predictions for underpredicted values (Table 4.2). For the purpose of this research the negative values were not removed to demonstrate the true predicting capabilities of the NN.

Table 4.2: Comparison of Including and Excluding Negative Values from Predictions

<table>
<thead>
<tr>
<th>Original Area (mm$^2$)</th>
<th>Predicted Area (mm$^2$) Including Negative Values</th>
<th>Predicted Area (mm$^2$) Excluding Negative Values</th>
<th>Percent Error (%) Including Negative Values</th>
<th>Percent Error (%) Excluding Negative Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.26</td>
<td>4.43</td>
<td>4.47</td>
<td>3.99</td>
<td>4.93</td>
</tr>
<tr>
<td>4.21</td>
<td>4.29</td>
<td>4.31</td>
<td>1.90</td>
<td>2.38</td>
</tr>
<tr>
<td>4.35</td>
<td>4.32</td>
<td>4.34</td>
<td>0.690</td>
<td>0.230</td>
</tr>
<tr>
<td>1.36</td>
<td>2.29</td>
<td>2.79</td>
<td>68.4</td>
<td>105</td>
</tr>
<tr>
<td>4.33</td>
<td>4.57</td>
<td>4.86</td>
<td>5.54</td>
<td>12.2</td>
</tr>
<tr>
<td>27.7</td>
<td>33.5</td>
<td>33.5</td>
<td>20.9</td>
<td>20.9</td>
</tr>
</tbody>
</table>

Additionally, using the predicted values for area (excluding negative values), histograms were plotted for Area = 1.36 mm$^2$ and Area = 27.7 mm$^2$ (Figure 4.7). These two particular values were chosen due to their error bars in Figure 4.6; one has symmetric error bars (Area = 27.7 mm$^2$) and the other does not (Area = 1.36 mm$^2$).

![Histograms](image)

Figure 4.7: Number of times a specific predicted value of area occured.

The skewness and kurtosis were calculated for both histograms. The skewness and kurtosis for Area = 1.36 mm$^2$ were 0.83 and 1.0 respectively. Similarly, the
results for Area = 27.7 mm$^2$ were -0.16 (skewness) and 0.28 (kurtosis). The kurtosis value of 0.28 indicates a normal distribution, whereas the 1.0 denotes a slightly more leptokurtic distribution. A skewness of 0.83 represents positively skewed data. In other words, the data is clustered near the left side of the histogram. In contrast, negative skewness (-0.16) indicates that the data is clustered near the right side of the histogram. The magnitude of skewness effects the symmetry of the error bars, which is why the Area = 1.36 mm$^2$ results are less symmetric than the Area = 27.7 mm$^2$ results. The sign of skewness indicates which direction the mean will be shifted.

**Wing Kinematics**

Both the frequency and stroke amplitude were analyzed. The results for frequency can be seen in Figure 4.8 and Table 4.1. The largest percent errors correspond with the extreme test points (Frequency: 145 Hz (15.9%) and Frequency: 254 Hz (12.6%)). Additionally, it is important to note that most of the error bars are non-symmetric and are skewed towards the expected values. Points near the middle of the dataset have a smaller range of predictions in comparison to those farther away.

The stroke amplitude (Figure 4.9), demonstrated the same pattern where extreme values had a wider range of predictions. All percent errors fell within 7.69% (stroke amplitude= 104 deg) and 0.752% (stroke amplitude =133 deg). Additionally, the majority of the 95% confidence intervals include the corresponding expected value.
Figure 4.8: Predicted values of frequency in comparison to their original values.

Figure 4.9: Predicted values of stroke amplitude in comparison to their original values.
Performance

The $C_L$ predictions can be viewed in Figure 4.10 and 4.1. The largest percent errors correspond with extreme values ($C_L = 1.3 (16.9\%)$ and $C_L = 1.82 (12.5\%)$). The rest of the percent errors are 4.67% and less, which indicates a good prediction. Additionally, most of errors bars include the true values and are non-symmetrical. Most of the means are skewed towards the true value indicating that the majority of the predictions are close to the expected values.

![Figure 4.10: Predicted values of $C_L$ in comparison to their original values.](image)

Mass

The mass predictions (Figure 4.11) do not follow any trend. Due to the scale of Figure 4.11, the results appear have relatively small percent errors, but upon close inspection of the resulting percent errors they do not. The largest percent error is 1100% (mass = 0.72 mg) which is likely the result of the dataset. Since the data was often recorded for different individuals of the same species, the resulting dataset contained clusters of data points. The effects on mass are more noticeable than other parameters because...
the mass had smaller variation across individual data points. Therefore, it is expected that the predictions of mass underperform.

Figure 4.11: Predicted values of mass in comparison to their original values.

4.2.2 Training Loss Curves of Small Insects

The training loss curves for each of the above numerical experiments are shown in Figure 4.12. None of the training loss curves display overtraining.
Figure 4.12: The training loss curves of small insects.
4.2.3 Large Insects

For the purpose of this experiment, the data seen in Table B.1 was used as the training set and the Table B.2 was used as the testing set. The original plan was to also perform this test with the large insect data as the testing set, but due to the scarcity of data in literature it could not be conducted. Only 4 useable points were found in literature. Following the same procedure as the previous section, the 4 points were passed through the NN to predict various wing parameters.

Wing Geometry

The results for area and wingspan can be seen in Figures 4.13, 4.14 respectively and their corresponding percent errors can be seen in Table 4.3. The error bars of the area predictions do not cross the expected value. This likely because the large insect wing areas do not fall within the small insect data. As the wing area diverges from the small insect dataset, the percent error increases. If data were provided that were smaller than the wing areas within the small dataset, we would expect to see a similar result in the opposite direction. The area predictions were roughly six to seven times smaller than the true values.

The percent errors for wingspan appears to follow the same trend, although the middle test points are flipped. The two points are within 0.46 mm of each other, and their percent errors differ only by 0.32%. Due to their proximity, it can be assumed that this is the result of numerical noise (random variation in data). The percent errors range from 40.63% (wingspan= 17.48 mm) to 39.46% (wingspan= 15.97 mm). It is important to note that the error bars include the expected value but are skewed in the opposite direction. This could be an instance where removing negative values improve the predictions.
Figure 4.13: Predicted values of area in comparison to their original values.

Figure 4.14: Predicted values of wingspan in comparison to their original values.
Table 4.3: Predicted Values of Large Insects

<table>
<thead>
<tr>
<th>Area</th>
<th>Wingspan</th>
<th>Frequency</th>
<th>Stroke Amplitude</th>
<th>$C_L$</th>
<th>Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$\overline{U_p}$</td>
<td>$PE$</td>
<td>$x$</td>
<td>$\overline{U_p}$</td>
<td>$PE$</td>
</tr>
<tr>
<td>(mm$^2$)</td>
<td>(mm$^2$)</td>
<td>(%)</td>
<td>(mm)</td>
<td>(mm)</td>
<td>(%)</td>
</tr>
<tr>
<td>327.87</td>
<td>50.653</td>
<td>84.551</td>
<td>27.85</td>
<td>16.77</td>
<td>39.78</td>
</tr>
<tr>
<td>312.38</td>
<td>48.726</td>
<td>84.402</td>
<td>27.23</td>
<td>16.31</td>
<td>40.10</td>
</tr>
<tr>
<td>292.59</td>
<td>47.725</td>
<td>83.689</td>
<td>26.38</td>
<td>15.97</td>
<td>39.46</td>
</tr>
<tr>
<td>357.44</td>
<td>53.292</td>
<td>85.091</td>
<td>29.44</td>
<td>17.48</td>
<td>40.63</td>
</tr>
</tbody>
</table>
The results of both the area and wingspan behave as expected. Both show the direct relationship between the original values and percent errors. Additionally, the errors are much larger than those of the smaller insects shown in Section 4.2.1.

**Wing Kinematics**

Two wing kinematic parameters were explored: wing beat frequency and stroke amplitude. Figure 4.15 depicts the prediction results for frequency. Their associated percent errors can be seen in Table 4.3. The percent errors ranged from 200% (frequency= 40 Hz) to 270% (frequency= 35 Hz). The lower frequencies had larger percent errors as they were on the lower extreme of the dataset. Physically, this makes sense because frequency increases as size decreases (Dudley, 2000). Additionally, it is important to note that the error bars do not cross the point at which the prediction is equal to the true value.

![Figure 4.15](image)

Figure 4.15: Predicted values of frequency in comparison to their original values.

The results for stroke amplitude can be seen in Figure 4.16 and Table 4.3. The percent errors range from 0% (stroke amplitude= 101 deg) to 34% (stroke amplitude=
91 deg). The magnitudes of the percent errors were smaller than those of frequency since the stroke amplitude values were similar to those of the small insect dataset. All the error bars crossed the 1:1 ratio line denoting that the 95% confidence interval encompasses the expected value. The error bars are skewed towards the 1:1 ratio line indicating that the predictions were clustered near the expected value.

Figure 4.16: Predicted values of stroke amplitude in comparison to their original values.

The results from the frequency and stroke amplitude trials denote the relationship between percent error and the similarity of the data in comparison to the training set. Since only four data were used, more studies would have to be conducted before a conclusion can be drawn. The results for stroke amplitude were more promising than those of frequency. All the error bars passed through the expected value were as none of the frequency error bars did.
Performance

The results for $C_L$ can be seen in Figure 4.17 and Table 4.3. The percent errors ranged from 55% ($C_L = 0.92$) to 71% ($C_L = 0.84$). The highest percent error corresponds with the lowest $C_L$ value. Once again, this is expected since the $C_L$ values fall towards to lower extreme of the dataset. All error bars pass through and are slightly skewed towards the expected value. These results would have greatly benefited from a larger training set.

![Figure 4.17](image-url)  
Figure 4.17: Predicted values of $C_L$ in comparison to their original values.

Mass

Figure 4.18 denotes the mass predictions of the large insects. The associated percent errors can be viewed in Table 4.3. They ranged from 142.7% (mass= 133.0 mg) to 179.2% (mass= 111.5 mg). It is important to note that the smaller the mass, the smaller the percent error. This trend is a direct result of the classification of datasets. Since datasets were classified by insect size, the larger insects would not be in the range of the training set. This is another example where the results would have
benefited from more data. Since the error bars are non-symmetrical and the mean is shifted towards the expected value, most of the predictions do fall near the expected value.

Figure 4.18: Predicted values of mass in comparison to their original values.

4.2.4 Training Loss Curves of Large Insects

The associated training loss curves for the numerical experiments of large insects are shown in Figure 4.19. Like the training loss curves of the small insects, the large insects did not overtrain.
Figure 4.19: The training loss curves of large insects.
4.2.5 Comparison Between the Smaller and Larger Insect Results

Most of the percent errors for the small insect predictions with the exception of mass are all 35% or lower. Most of which have a percent error of 5% or lower. The mass has the most variability in percent errors, followed by the wingspan. The lack of agreement is likely the result of a deficit of variability in mass and consistency in wingspan within the literature. The available dataset is rather small in comparison to many NN datasets. Although the results were deemed acceptable for the purpose of this numerical experiment, the analysis would benefit from additional training and testing data.

In regard to the large insect predictions, the percent errors are larger in comparison to the small insect predictions. The large insect data was not within the range of the small insect data which it was being predicted from. If additional data was available, an increase in accuracy would be expected.

4.2.6 Case Studies

In this section, micro-aerial vehicle data were predicted and examined. For the purpose of this numerical experiment, Table B.3 was used as a testing set and Table B.1 was used as a training set. The four MAV used in this numerical experiment were the Dove, the Delfly-micro, the Delfly-2 and the Robobee.

The Dove is a large flapping wing MAV developed by the Northwestern Polytechnical University (Yang et al., 2018). With a wingspan of 50 cm and a mass of 220 g, the dove is capable of a half hour flight duration (Yang et al., 2018). Additionally, it can fly fully autonomously and transmit live stabilized colour video to a ground station over 4 km away (Yang et al., 2018).
Both the Delfly-micro (Figure 4.21 (a)) and Delfly-2 (Figure 4.21 (b)) were developed by the Delft University of Technology (de Croon et al., 2009). With a wingspan of 10 cm and weighing only 3 g, the Delfly-micro is the smaller of the two (de Croon et al., 2009). The Delfly-2 has a wingspan of 28 cm and weighs 16 g (de Croon et al., 2012). It is capable of forward flight at a speed of 7 m/s, backwards flight at 1 m/s and hovering (de Croon et al., 2012). The Delfly-micro and Delfly-2 are unique from
traditional flapping wing MAV because they consist of 2 pairs of wings, one on top of the other. Each pair of wings performs a similar motion to the clap-fling mechanism.

Wood (2007) designed the Robobee at Harvard in 2007 (Figure 4.22). With a mass of 80 mg and a wingspan of 16 mm (Wood, 2007), is the smallest MAV among the presented case studies. It is also capable of the largest stroke amplitude and frequency. Given the nature of this MAV, overall, it should produce lower percent errors than the other MAV.

**Wing Geometry**

The results for wing area of the case study can be seen in Figure 4.23 and Table 4.4. The percent errors were extremely large, some of these predictions would have greatly benefited from the removal of negative values. The scale associated with the axis in this figure varies widely due to the large and small predictions. Three of the four 95% confidence intervals pass through the true value, the only one that does not
is the Dove.

Figure 4.23: Predicted values of area for MAV in comparison to their original values.

Figure 4.24: Predicted values of wingspan for MAV in comparison to their original values.
Table 4.4: Predicted Values of Case Studies on MAV

<table>
<thead>
<tr>
<th>MAV</th>
<th>Area $x$</th>
<th>$\overline{U_p}$</th>
<th>$PE_x$</th>
<th>Wingspan $x$</th>
<th>$\overline{U_p}$</th>
<th>$PE_x$</th>
<th>Frequency $x$</th>
<th>$\overline{U_p}$</th>
<th>$PE_x$</th>
<th>Stroke Amplitude $x$</th>
<th>$\overline{U_p}$</th>
<th>$PE_x$</th>
<th>$C_L$ $x$</th>
<th>Mass $x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dove</td>
<td>340000000</td>
<td>1200000</td>
<td>100</td>
<td>500</td>
<td>14.6</td>
<td>97.1</td>
<td>9</td>
<td>141</td>
<td>1467</td>
<td>70</td>
<td>105</td>
<td>50</td>
<td>1.0</td>
<td>1.6</td>
</tr>
<tr>
<td>Delfly-Micro</td>
<td>1310</td>
<td>-2640000</td>
<td>201627</td>
<td>50.5</td>
<td>7.74</td>
<td>84.7</td>
<td>37.5</td>
<td>129</td>
<td>244</td>
<td>25.5</td>
<td>85.1</td>
<td>234</td>
<td>1.81</td>
<td>1.49</td>
</tr>
<tr>
<td>Delfly-2</td>
<td>10300</td>
<td>-1250000</td>
<td>12236</td>
<td>140</td>
<td>8.4</td>
<td>94</td>
<td>11.0</td>
<td>111</td>
<td>909</td>
<td>22.0</td>
<td>77.6</td>
<td>253</td>
<td>1.36</td>
<td>1.45</td>
</tr>
<tr>
<td>Robobee</td>
<td>75</td>
<td>1400000</td>
<td>1866567</td>
<td>16</td>
<td>5.5</td>
<td>65</td>
<td>105</td>
<td>126</td>
<td>20.0</td>
<td>100</td>
<td>93.9</td>
<td>6</td>
<td>0.63</td>
<td>1.2</td>
</tr>
</tbody>
</table>
For wingspan, only the Robobee’s wingpsan (16mm) fell within the training set. It produced the smallest percent error (65%). The dove has the largest wingspan (500 mm) which resulted in the worst percent error (97.1%). The Delfly-2 had the second largest wingspan (140 mm) with the second worst percent error (94%), followed by the Delfly-micro (wingspan = 50.5 mm, percent error = 84.7%). The magnitude of the wingspan correlates with the percent error; as wingspan increased, percent error increased.

**Wing Kinematics**

The results for frequency and stroke amplitude are shown in Figures 4.25 and 4.26 and Table 4.4. The percent errors for frequency range from 20% to 1467%. The Robobee frequency was the highest whereas the Dove was the lowest which corresponds with their percent errors. In this numerical experiment, larger frequencies were associated with smaller percent errors since they had a greater difference from the small insect dataset. The 95% confidence intervals included the expected values except for the Delfly-micro.

For stroke amplitude, the Delfly-2 had the largest percent error (253%). Physically, the Delfly-2 is different; it has two pairs of wings where one wing is on top if the other. The wings are unable to flap large distances because they are constrained by the other wing. Similarly, the Delfly-micro produced a percent error of 234%. With a percent error of 6% the Robobee performed the best in this numerical study. The Robobee is fundamentally similar to the small insect data and therefore should have produced this result.
Figure 4.25: Predicted values of frequency for MAV in comparison to their original values.

Figure 4.26: Predicted values of stroke amplitude for MAV in comparison to their original values.
Performance

The results for $C_L$ had the lowest range of percent errors in comparison to all the other parameters presented in this numerical study. The percent errors ranged from 6.62% (Delfly-2) to 90% (Robobee). Due to their similarities with the training set, it makes sense that $C_L$ performed better as a whole. In regard to their 95% confidence interval, three of the four test points passed through the expected value, only the Robobee did not. The Robobee had a $C_L$ of 0.63 which was not included in the range of the small insect data. It is also possible that there is a discrepancy in the lift recorded for the purpose of this numerical experiment since the data was spread throughout a few literature sources.

![Figure 4.27: Predicted values of $C_L$ for MAV in comparison to their original values.](image)

Mass

All the MAV mass values were larger than the mass values provided in the training set, with the exception of the Robobee. Although the Robobee was within the range of the training set, it was still towards the high end of the values. The Robobee produces
the highest percent error (250%), which was not expected since it fell within the range of the testing set. With a percent error of 133.2% the Delfly-2 performed the best. Similarly, to the biological predictions of mass, there was no apparent trend in mass predictions.

Figure 4.28: Predicted values of mass for MAV in comparison to their original values.

The results for mass behaved as the biological predictions; they showed no trend in results. The stroke amplitude, wingspan and area behaved as expected showing trends between percent error and magnitude of parameters. Overall, the $C_L$ predictions performed the best, but still do not compare to the results of the small insect predictions.

4.2.7 Training Loss Curves of MAVs

Figure 4.29 displays the training loss curves for MAVs; none show overtraining.
When comparing the MAV, the Robobee performed the best out of the four. It maintained the lowest percent errors in wingspan, frequency, and stroke amplitude.
This is due to its compatibility with the small insect dataset. The other three MAV are fundamentally different than the training set. For instance, the Delfly-micro and the Delfly-2 have 2 pair of wings that each perform a similar motion to the clap-fling mechanism. The Dove on the other hand, is modeled after a real Dove and although it flaps similarly to insects, the wings are much larger and flap at a slower frequency than insects.

4.3 Linear and Non-Linear Data

To gain a greater understanding of the impact of dataset size and complexity, linear and non-linear datasets were generated. Functions were used so that there was a known relationship between variables that could be easily calculated. The number of points in the datasets were varied to determine the effects of the size of a training set. The changes in the number of epochs and percent error were explored. Results can be seen in Table 4.5.

The average percent error and average number of epochs decreased as the size of the dataset increased. Additionally, the complexity of the dataset played a role in the percent errors and number of epochs. The linear dataset had the lowest values whereas the non-linear results were the largest. Although the percent errors are much larger than those associated with the biological data it is not a concern. Complex NNs tend to under-perform with simple relationships (Aggarwal, 2018). They are best used for complicated datasets with many variables as opposed to known relationships with only two variables. Regardless of the magnitudes of the percent errors, these results are still crucial to our understanding of how the neural network is performing with the biological data. An increase in dataset size results in a decrease with percent error.
Table 4.5: The Effects of Datasets

<table>
<thead>
<tr>
<th>Dataset Size</th>
<th>Planar</th>
<th>Paraboloid</th>
<th>Non-Linear</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Epoch</td>
<td>PE (%)</td>
<td>Epoch</td>
</tr>
<tr>
<td>35</td>
<td>92</td>
<td>33.52</td>
<td>102</td>
</tr>
<tr>
<td>64</td>
<td>55</td>
<td>43.36</td>
<td>75</td>
</tr>
<tr>
<td>100</td>
<td>43</td>
<td>35.86</td>
<td>56</td>
</tr>
<tr>
<td>400</td>
<td>26</td>
<td>26.93</td>
<td>45</td>
</tr>
<tr>
<td>900</td>
<td>19</td>
<td>19.33</td>
<td>51</td>
</tr>
</tbody>
</table>

### 4.4 Key Observations

If additional training and testing data were provided, the NN predictions would improve. With supplemental data, we would expect to see most of the predictions improve if:

1. they fell within the training set; and

2. they were physically similar to the training set.

When the underlying physics differ from the training set, the NN cannot predict the data properly. The physics must be considered when generating datasets, which is supported by the results of the Delfly-micro and Delfly-2. The small insect predictions show that NNs perform better as the test point approaches the mean of the training set. This can also be inferred from the large insect data as well as the MAV data,
but since there were limitations in data, it is hard to draw conclusions with only 4 data points.
Chapter 5

Concluding Remarks and Future Work

5.1 Conclusions

The results of this research have demonstrated that neural networks can successfully predict various parameters of flapping wings using existing datasets for training. This is of particular use to engineers and hobbyists designing flapping wing MAVs. To this day, there are no lift equations without major simplifications that effectively describe flexible flapping wing flight. With the use of this NN, individuals can input data to generate a value for lift. In other words, engineers can quantify the lift of a flapping wing which would not be possible without lengthy manufacturing and experimental procedures or Computational Fluid Dynamics (CFD). It serves as a quick alternative to testing. Additionally, it can be used in reverse, where a lift is given, and various wing parameters are predicted.

5.2 Recommendations

Although the results from this research are promising, a larger dataset would be needed to further improve results. Larger datasets have been shown to improve predictions (Sagina, 2018), as demonstrated in Section 4.3. It is recommended that
a large database of datasets be generated for the various categories of insects. For example, in this research the datasets are divided by insect size, but they should also be divided by the number of wings they possess. The separate datasets should then be tested independently and combined to determine if the various separation categories have a significant impact on the results. Additionally, a database of MAVs should be generated and tested. Unfortunately, only a few lift datum for pre-existing MAVs have been published. It would be interesting to see how MAV predictions behave when the training set is comprised of entirely MAV data.

Results could be improved, by writing a few lines of code which remove negative values from predictions. This only works for underpredicted values whose true values are known. Removing negative predictions from overpredicted values shifts the mean upwards and away from the true value. The code could be as simple as an `if, elif` statement which differentiates between underpredicted and overpredicted values. This is only useful when the true value of the prediction is known. For this reason, negative values were kept even though they are non-feasible.

It is also imperative that future data parameters be defined in the same manner. For example, the tests conducted on wingspan have larger percent errors due to the inconsistency in how wingspan was defined in literature. Future data must have a consistent definition of parameters across datasets. This not only helps with the predictions of a particular dataset, but it allows the datasets to be interchangeable with each other. Datasets could be interchanged to determine relationships between different categories without the added factor of inconsistency.

Future datasets could include other parameters, such as elasticity. Elasticity is a key factor in flexible flapping wing performance which has not yet directly been accounted for within this research. Most of the literature included in Tables B.1 and B.2 did not include elasticity values. Although some literature containing elasticity values exists (Lehmann et al., 2011; Sivasankaran et al., 2017; Truong et al., 2017),
they are very limited and are often incomplete.

Aside from the generation of more data, future work should include predictions without mass. The mass predictions did not follow a trend and often included the highest percent errors. Although mass is an important parameter in the design of flapping wings, it would be interesting to see if the predictions of other variables improve when mass is removed.

5.3 Contributions

This research provides a promising step forward in the knowledge concerning flapping flight, but more research must be conducted in this area. The work showcased in this research provides a new twist to the application of NNs for flapping flight. It uses biological data to predict the performance of flapping wings rather than the rigid wings and CFD models used in previous work. It includes the underlying deformation found within true flapping flight of flexible wings. Additionally, this thesis proposes a new strategy for MAV design. Engineers and hobbyists can predict the lift generated by flapping wings without the need for lengthy experiments or simulations. In fact, the Capstone Micro Flapping-Wing Flyer project at Carleton will be making use of this work to design a flapping wing MAV.

In conclusion, the results of this research are encouraging, but this work would greatly benefit from a larger dataset. More data must be generated in order to truly visualize the potential.
List of References


Ganesh, S. (2020, July). What’s The Role Of Weights And Bias In a Neural Network? *Towards Data Science*.


Le, T. Q., T. V. Truong, S. H. Park, T. Quang Truong, J. H. Ko, H. C. Park, and


Phan, H. V., S. Aurecianus, T. Kang, and H. C. Park (2019, January). KUBeetle-S:


Appendix A

Neural Network Code

Appendix A displays the code for the neural network used in this thesis. In this excerpt, the predicted variable (or output variable) is stroke amplitude. Only one dataset is being used and the two predictions are being made. As you can see below, this code can easily be manipulated to add or remove variables as well as change the predicted variable. Additional predictions can also be conducted easily. This code could be used for a range of applications aside from flapping flight.

```python
import matplotlib
matplotlib.use("TkAgg")
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import tensorflow as tf
import scipy.stats as st
import time
import csv

# Starting timer
t0 = time.time()
```
# Reading in dataset
df = pd.read_csv('newdata.csv')

# Scaling dataset
# Note: the title in quotations was the header used in the dataset
df_prescaled = df.copy()
maxPrice = df['Stroke Amplitude'].max()
minPrice = df['Stroke Amplitude'].min()
Y = (df['Stroke Amplitude'] - minPrice) / (maxPrice - minPrice)

max_x1 = df['Frequency (Hz)'].max()
min_x1 = df['Frequency (Hz)'].min()
x1 = (df['Frequency (Hz)'] - min_x1) / (max_x1 - min_x1)

max_x2 = df['2 wing / 4 wing'].max()
min_x2 = df['2 wing / 4 wing'].min()
x2 = (df['2 wing / 4 wing'] - min_x2) / (max_x2 - min_x2)

max_x3 = df['CL'].max()
min_x3 = df['CL'].min()
x3 = (df['CL'] - min_x3) / (max_x3 - min_x3)

max_x4 = df['Wingspan (mm)'].max()
min_x4 = df['Wingspan (mm)'].min()
x4 = (df['Wingspan (mm)'] - min_x4) / (max_x4 - min_x4)

max_x5 = df['m (mg - total mass)'].max()
min_x5 = df['m (mg - total mass)'].min()
x5 = (df['m (mg - total mass)'] - min_x5) / (max_x5 - min_x5)

max_x6 = df['Area (mm\^2)'].max()
min_x6 = df['Area (mm\^2)'].min()
x6 = (df['Area (mm\^2)'] - min_x6) / (max_x6 - min_x6)

df_new = pd.concat([x1, x2, x3, x4, x5, x6, Y], axis=1)
df = df_new.copy()

# Spliting input and output variables
X = df.loc[:, df.columns != 'Y']

# Spliting testing and training set
num_test = 6
X_train = X.iloc[0:35]
Y_train = Y.iloc[0:35]

X_test = X.iloc[36:42]
Y_test = Y.iloc[36:42]

# Setting the number of iterations
num_trials = 600

# Building the neural network
P = []
for i in range(0, num_trials):
    model = Sequential()
    model.add(Dense(20, input_dim=X_train.shape[1],
                    activation="relu"))
    model.add(Dense(1))

    # Implement early epoch stopping
    epochs = 6000
    callback = tf.keras.callbacks.EarlyStopping(monitor='loss',
                                                min_delta=0.001,
                                                patience=3)

    # Note: will stop epochs if there's no improvement of
    # more than min_delta after 3 iterations

    # learning rate changing with epochs
    learning_rate = 0.00001
    decay_rate = learning_rate / epochs
    tf.keras.optimizers.Adam(learning_rate=learning_rate,\n                             decay=decay_rate, name='Adam')

    # Fit and save models
    model.compile(loss='mse', optimizer='adam',
                  metrics=['accuracy'])
    history = model.fit(X_train, Y_train, epochs=epochs,
                         callbacks=[callback])
    #filename = '/Users/liv/PycharmProjects/thesis4/Saved
    #Models/model1.1.h5'
    #model.save(filename)

test_pred =model.predict(X_test)
test_pred.tolist()
P.append(test_pred)

new_P = np.stack(P, axis=0)
new_P.flatten()
#print(new_P)

test = Y_test * np.ones(Y_test.shape[0])

# Unscaling input test variables
unscaling_Y_test = np.vstack((Y_test * (maxPrice - minPrice)) + minPrice)
#print(unscaling_Y_test)

# Unscaling predictions and calculating mean
mean = np.mean(P, axis=0)
unscaled_mean = (mean * (maxPrice - minPrice)) + minPrice

# Stop timer
t1 = time.time()
total = t1 - t0
print(total)
Appendix B

Datasets

Appendix B conveys the various datasets used for the experiments conducted in this thesis. It includes the small and large insect data as well as the MAV data used for the case studies. The “*” displayed in the small insect data denotes the six random test points used for the small insect predictions in Section 4.2.1. For the experiments conducted in Sections 4.2.3 and 4.2.6 used the entire small insect dataset (Table B.1) as the testing set. Tables B.2 and B.3 were used at the testing sets for Sections 4.2.3 and 4.2.6 respectively.
Table B.1: Small Insect Dataset

<table>
<thead>
<tr>
<th>Reference</th>
<th>Scientific Name</th>
<th>Common Name</th>
<th>Area (mm²)</th>
<th>Wingspan (mm)</th>
<th>Frequency (Hz)</th>
<th>Stroke Amplitude (deg)</th>
<th>C_L</th>
<th>Mass (mg)</th>
<th>Number of Wings</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Eanes and Wootton, 1989)*</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>1.36</td>
<td>2.02</td>
<td>254</td>
<td>150</td>
<td>1.59</td>
<td>0.72</td>
<td>2</td>
</tr>
<tr>
<td>(Ellington, 1984b,c)</td>
<td>Tipula obsoleta</td>
<td>Cranefly</td>
<td>30.18</td>
<td>12.7</td>
<td>45.5</td>
<td>123</td>
<td>1.26</td>
<td>11.4</td>
<td>2</td>
</tr>
<tr>
<td>(Ellington, 1984b,c)</td>
<td>Episyrphus balteatus</td>
<td>Marmalade Hoverfly</td>
<td>20.48</td>
<td>9.3</td>
<td>160</td>
<td>90</td>
<td>1.48</td>
<td>27.3</td>
<td>2</td>
</tr>
<tr>
<td>(Ellington, 1984b,c)</td>
<td>Eristalis tenax</td>
<td>Common Dronefly</td>
<td>36.89</td>
<td>11.4</td>
<td>157</td>
<td>109</td>
<td>1.05</td>
<td>68.4</td>
<td>2</td>
</tr>
<tr>
<td>(Ellington, 1984b,c)</td>
<td>Apis mellifera</td>
<td>Honey bee</td>
<td>30.14</td>
<td>9.8</td>
<td>197</td>
<td>131</td>
<td>1.08</td>
<td>101.9</td>
<td>4</td>
</tr>
<tr>
<td>(Dudley and Ellington, 1990a)</td>
<td>Bombus hortorum</td>
<td>Bumble bee</td>
<td>54.9</td>
<td>13.2</td>
<td>155</td>
<td>116</td>
<td>1.17</td>
<td>175</td>
<td>4</td>
</tr>
<tr>
<td>(Fry et al., 2005)*</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.33</td>
<td>2.48</td>
<td>212</td>
<td>147</td>
<td>1.3</td>
<td>1.2</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.14</td>
<td>2.37</td>
<td>211</td>
<td>139</td>
<td>1.71</td>
<td>1.11</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.04</td>
<td>2.31</td>
<td>225</td>
<td>143</td>
<td>1.01</td>
<td>0.61</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.37</td>
<td>2.5</td>
<td>211</td>
<td>142</td>
<td>1.35</td>
<td>1.06</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.19</td>
<td>2.4</td>
<td>227</td>
<td>151</td>
<td>1.29</td>
<td>1.22</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.02</td>
<td>2.3</td>
<td>221</td>
<td>120</td>
<td>1.49</td>
<td>0.57</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.30</td>
<td>2.46</td>
<td>189</td>
<td>135</td>
<td>1.48</td>
<td>0.63</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.35</td>
<td>2.49</td>
<td>186</td>
<td>144</td>
<td>1.57</td>
<td>0.81</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)*</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.26</td>
<td>2.44</td>
<td>188</td>
<td>128</td>
<td>1.55</td>
<td>0.58</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)*</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.21</td>
<td>2.41</td>
<td>194</td>
<td>133</td>
<td>1.5</td>
<td>0.56</td>
<td>2</td>
</tr>
<tr>
<td>(Fry et al., 2005)</td>
<td>Drosophila melanogaster</td>
<td>Common Fruitfly</td>
<td>4.56</td>
<td>2.61</td>
<td>195</td>
<td>106</td>
<td>1.46</td>
<td>0.43</td>
<td>2</td>
</tr>
<tr>
<td>(Hocking, 1935)</td>
<td>Tabanus affinis</td>
<td>Horsefly</td>
<td>57.4</td>
<td>14.3</td>
<td>120</td>
<td>114</td>
<td>1.2</td>
<td>180</td>
<td>2</td>
</tr>
<tr>
<td>(Hocking, 1935)</td>
<td>Aedes nearcticus</td>
<td>Mosquito</td>
<td>3.6</td>
<td>3.8</td>
<td>320</td>
<td>114</td>
<td>1.2</td>
<td>3.5</td>
<td>2</td>
</tr>
<tr>
<td>(Cheng and Sun, 2016)</td>
<td>Liriomyza sativae</td>
<td>Vegetable leaf miner</td>
<td>0.61</td>
<td>1.44</td>
<td>260.8</td>
<td>182.4</td>
<td>1.99</td>
<td>0.25</td>
<td>2</td>
</tr>
<tr>
<td>(Cheng and Sun, 2016)</td>
<td>Liriomyza sativae</td>
<td>Vegetable leaf miner</td>
<td>0.63</td>
<td>1.46</td>
<td>270.4</td>
<td>174.1</td>
<td>1.87</td>
<td>0.32</td>
<td>2</td>
</tr>
<tr>
<td>(Cheng and Sun, 2016)</td>
<td>Liriomyza sativae</td>
<td>Vegetable leaf miner</td>
<td>0.67</td>
<td>1.49</td>
<td>259.3</td>
<td>181.7</td>
<td>1.96</td>
<td>0.29</td>
<td>2</td>
</tr>
<tr>
<td>(Cheng and Sun, 2016)</td>
<td>Liriomyza sativae</td>
<td>Vegetable leaf miner</td>
<td>0.6</td>
<td>1.45</td>
<td>188.4</td>
<td>172.4</td>
<td>1.79</td>
<td>0.31</td>
<td>2</td>
</tr>
<tr>
<td>(Cheng and Sun, 2016)</td>
<td>Liriomyza sativae</td>
<td>Vegetable leaf miner</td>
<td>0.52</td>
<td>1.31</td>
<td>267.5</td>
<td>182.0</td>
<td>2.16</td>
<td>0.17</td>
<td>2</td>
</tr>
<tr>
<td>(Meng and Sun, 2015)</td>
<td>Drosophila virilis</td>
<td>Fruitfly</td>
<td>2.27</td>
<td>2.9</td>
<td>161</td>
<td>159</td>
<td>1.65</td>
<td>1.18</td>
<td>2</td>
</tr>
<tr>
<td>Reference</td>
<td>Scientific Name</td>
<td>Common Name</td>
<td>Area (mm²)</td>
<td>Wingspan (mm)</td>
<td>Frequency (Hz)</td>
<td>Stroke Amplitude (deg)</td>
<td>$C_L$</td>
<td>Mass (mg)</td>
<td>Number of Wings</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------</td>
<td>---------------</td>
<td>------------</td>
<td>----------------</td>
<td>-----------------</td>
<td>------------------------</td>
<td>-------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>(Meng and Sun, 2015)</td>
<td><em>Drosophila virilis</em></td>
<td>Fruitfly</td>
<td>2.45</td>
<td>2.88</td>
<td>178</td>
<td>136</td>
<td>1.82</td>
<td>1.13</td>
<td>2</td>
</tr>
<tr>
<td>(Meng and Sun, 2015)</td>
<td><em>Drosophila virilis</em></td>
<td>Fruitfly</td>
<td>2.7</td>
<td>2.96</td>
<td>171</td>
<td>147</td>
<td>1.78</td>
<td>1.38</td>
<td>2</td>
</tr>
<tr>
<td>(Meng and Sun, 2015)</td>
<td><em>Drosophila virilis</em></td>
<td>Fruitfly</td>
<td>2.4</td>
<td>2.92</td>
<td>208</td>
<td>134</td>
<td>1.79</td>
<td>1.66</td>
<td>2</td>
</tr>
<tr>
<td>(Meng and Sun, 2015)</td>
<td><em>Drosophila virilis</em></td>
<td>Fruitfly</td>
<td>2.78</td>
<td>3.15</td>
<td>184</td>
<td>147</td>
<td>1.60</td>
<td>1.78</td>
<td>2</td>
</tr>
<tr>
<td>(Meng and Sun, 2015)</td>
<td><em>Drosophila virilis</em></td>
<td>Fruitfly</td>
<td>2.61</td>
<td>3.06</td>
<td>198</td>
<td>142</td>
<td>1.74</td>
<td>1.88</td>
<td>2</td>
</tr>
<tr>
<td>(Meng and Sun, 2015)</td>
<td><em>Drosophila virilis</em></td>
<td>Fruitfly</td>
<td>2.27</td>
<td>2.87</td>
<td>213</td>
<td>153</td>
<td>2.05</td>
<td>1.97</td>
<td>2</td>
</tr>
<tr>
<td>(Dudley and Ellington, 1990a,b)</td>
<td><em>Bombus terrestris</em></td>
<td>Bumblebee</td>
<td>26.6</td>
<td>13.2</td>
<td>155</td>
<td>118</td>
<td>1.19</td>
<td>180</td>
<td>4</td>
</tr>
<tr>
<td>(Dudley and Ellington, 1990a,b)*</td>
<td><em>Bombus terrestris</em></td>
<td>Bumblebee</td>
<td>27.7</td>
<td>13.7</td>
<td>145</td>
<td>104</td>
<td>2.08</td>
<td>595</td>
<td>4</td>
</tr>
<tr>
<td>(Dudley and Ellington, 1990a,b)</td>
<td><em>Bombus terrestris</em></td>
<td>Bumblebee</td>
<td>41.0</td>
<td>15.4</td>
<td>166</td>
<td>130</td>
<td>2.65</td>
<td>115.9</td>
<td>4</td>
</tr>
<tr>
<td>(Buchwald and Dudley, 2010)</td>
<td><em>Bombus impatiens</em></td>
<td>Bumblebee</td>
<td>70.3</td>
<td>10.9</td>
<td>181</td>
<td>144</td>
<td>2</td>
<td>200.1</td>
<td>4</td>
</tr>
<tr>
<td>(Dudley, 1995)</td>
<td><em>Euglossa dissimula</em></td>
<td>Orchid bee</td>
<td>10.6</td>
<td>8.8</td>
<td>214</td>
<td>103</td>
<td>2.08</td>
<td>91.0</td>
<td>4</td>
</tr>
<tr>
<td>(Dudley, 1995)</td>
<td><em>Euglossa imperialis</em></td>
<td>Orchid bee</td>
<td>1.5</td>
<td>10.9</td>
<td>189</td>
<td>105</td>
<td>1.84</td>
<td>151.1</td>
<td>4</td>
</tr>
<tr>
<td>(Dudley, 1995)</td>
<td><em>Euglossa meriana</em></td>
<td>Orchid bee</td>
<td>71.8</td>
<td>21.1</td>
<td>115</td>
<td>105</td>
<td>1.87</td>
<td>819.6</td>
<td>4</td>
</tr>
<tr>
<td>(Mon et al., 2011)</td>
<td><em>Episyrphus balteatus</em></td>
<td>Hoverfly</td>
<td>21.88</td>
<td>9.7</td>
<td>162</td>
<td>65.9</td>
<td>1.63</td>
<td>22.08</td>
<td>2</td>
</tr>
<tr>
<td>(Mon et al., 2011)</td>
<td><em>Episyrphus balteatus</em></td>
<td>Hoverfly</td>
<td>19.96</td>
<td>9.68</td>
<td>183</td>
<td>66.2</td>
<td>1.85</td>
<td>23.63</td>
<td>2</td>
</tr>
<tr>
<td>(Mon et al., 2011)</td>
<td><em>Episyrphus balteatus</em></td>
<td>Hoverfly</td>
<td>11.87</td>
<td>6.93</td>
<td>186</td>
<td>86.1</td>
<td>1.7</td>
<td>10.39</td>
<td>2</td>
</tr>
<tr>
<td>(Mon et al., 2011)</td>
<td><em>Episyrphus balteatus</em></td>
<td>Hoverfly</td>
<td>12.17</td>
<td>7.13</td>
<td>164</td>
<td>84.2</td>
<td>1.71</td>
<td>10.41</td>
<td>2</td>
</tr>
</tbody>
</table>
## Table B.2: Large Insect Dataset

<table>
<thead>
<tr>
<th>Reference</th>
<th>Scientific Name</th>
<th>Common Name</th>
<th>Area (mm²)</th>
<th>Wingspan (mm)</th>
<th>Frequency (Hz)</th>
<th>Stroke Amplitude (deg)</th>
<th>$C_L$</th>
<th>Mass (mg)</th>
<th>Number of Wings</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Wakeling and Ellington, 1997a,b)</td>
<td><em>Sympetrum sanguineum</em></td>
<td>Muller dragonfly</td>
<td>327.87</td>
<td>26.9</td>
<td>36</td>
<td>92</td>
<td>0.92</td>
<td>121.9</td>
<td>4</td>
</tr>
<tr>
<td>(Wakeling and Ellington, 1997a,b)</td>
<td><em>Sympetrum sanguineum</em></td>
<td>Muller dragonfly</td>
<td>312.38</td>
<td>26.17</td>
<td>40</td>
<td>96</td>
<td>0.92</td>
<td>133.0</td>
<td>4</td>
</tr>
<tr>
<td>(Wakeling and Ellington, 1997a,b)</td>
<td><em>Sympetrum sanguineum</em></td>
<td>Muller dragonfly</td>
<td>292.59</td>
<td>25.49</td>
<td>40</td>
<td>115</td>
<td>0.84</td>
<td>111.5</td>
<td>4</td>
</tr>
<tr>
<td>(Wakeling and Ellington, 1997a,b)</td>
<td><em>Sympetrum sanguineum</em></td>
<td>Muller dragonfly</td>
<td>357.44</td>
<td>28.54</td>
<td>35</td>
<td>105</td>
<td>0.96</td>
<td>139.3</td>
<td>4</td>
</tr>
</tbody>
</table>

## Table B.3: Dataset for the Case Study on MAVs

<table>
<thead>
<tr>
<th>Reference</th>
<th>Name</th>
<th>Area (mm²)</th>
<th>Wingspan (mm)</th>
<th>Frequency (Hz)</th>
<th>Stroke Amplitude (deg)</th>
<th>$C_L$</th>
<th>Mass (mg)</th>
<th>Number of Wings</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Yang et al., 2018)</td>
<td>Dove</td>
<td>3400000000</td>
<td>500</td>
<td>9</td>
<td>70</td>
<td>1.0</td>
<td>220000</td>
<td>2</td>
</tr>
<tr>
<td>(Tay et al., 2015)</td>
<td>Delfly-Micro</td>
<td>1310</td>
<td>50.5</td>
<td>37.5</td>
<td>25.5</td>
<td>1.81</td>
<td>3000</td>
<td>4</td>
</tr>
<tr>
<td>(Tay et al., 2014, 2015)</td>
<td>Delfly-2</td>
<td>10300</td>
<td>140</td>
<td>11.0</td>
<td>22.0</td>
<td>1.36</td>
<td>16070</td>
<td>4</td>
</tr>
<tr>
<td>(Wood, 2007; Chen et al., 2016; Bhushan, 2016)</td>
<td>Robobee</td>
<td>75</td>
<td>16</td>
<td>105</td>
<td>100</td>
<td>0.63</td>
<td>80.00</td>
<td>2</td>
</tr>
</tbody>
</table>