



**University of London**

# **6CCS3PRJ Final Year Project**

## **Passive Dynamics of a Robotic Bat Wing Flapping using an Anisotropic Membrane**

Final Project Report

Author: Michael Zheng

Supervisor: Dr Thrishantha Nanayakkara

Student ID: 1016630

Programme of Study: BSc Computer Science

April 22, 2014

## **Abstract**

This project focuses on the anisotropic stiffness distribution of the bat wing, and its application in the field of robotics and biologically inspired systems. Nature is vastly complex. It serves as a invaluable source of knowledge and inspiration, particularly in the field of Computer Science. The project seeks to understand the role of the anisotropic stiffness distribution of the bat wing, and how it reduces the necessity for active control during flight, thus conserving energy.

The project presents the design, implementation and subsequent analysis of a robotic bat wing with aero-elastic wing membrane properties. A number of different approaches are considered, both from a hardware and a software perspective, and analytical software is developed in order to draw conclusions regarding the effectiveness of these approaches. Hardware solutions include 3D printing the bat wing skeleton, and using a latex membrane with an elastic network to mimic the anisotropic behaviour of the bat wing. Energy usage during different phases in the wing-beat cycle is analysed, and long exposure photographs capture the extending-collapsing behaviour of the robotic bat wing.

### **Originality Avowal**

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Michael Zheng

April 22, 2014

Word count: 15405

### **Acknowledgements**

I would like to thank my supervisor Dr. Thrishantha Nanayakkara for his guidance and support throughout this project.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Literature Review . . . . .	5
2.2	Control Mechanism . . . . .	12
2.3	Actuating mechanism . . . . .	13
2.4	Measurement . . . . .	16
2.5	Interface . . . . .	18
<b>3</b>	<b>Specification</b>	<b>20</b>
3.1	User Requirements . . . . .	20
3.2	Functional Requirements . . . . .	21
3.3	Hardware Requirements . . . . .	21
3.4	Other Requirements . . . . .	22
<b>4</b>	<b>Design</b>	<b>23</b>
4.1	Interface . . . . .	23
4.2	Control Mechanism . . . . .	24
4.3	Measurement . . . . .	24
4.4	Actuating Mechanism . . . . .	25
4.5	Wing Design . . . . .	27
4.6	Software . . . . .	32
<b>5</b>	<b>Implementation</b>	<b>37</b>
5.1	Hardware . . . . .	37
5.2	Software . . . . .	42
<b>6</b>	<b>Professional and Ethical Issues</b>	<b>47</b>
<b>7</b>	<b>Experimental Results and Testing</b>	<b>48</b>
7.1	Experimental Results . . . . .	48
7.2	Testing . . . . .	53

<b>8 Evaluation</b>	<b>55</b>
8.1 Project Evaluation . . . . .	55
8.2 Evaluation Against Requirements . . . . .	57
<b>9 Conclusion and Future Work</b>	<b>59</b>
Bibliography . . . . .	64
<b>A Bat Wing Skeleton Models</b>	<b>65</b>
A.1 Body . . . . .	65
A.2 Shoulder . . . . .	66
A.3 Humerus . . . . .	67
A.4 Radioulna . . . . .	68
A.5 Digits I and V . . . . .	69
A.6 Digits II and III . . . . .	70
A.7 Digit IV . . . . .	71
A.8 Leg . . . . .	72
<b>B Diagrams</b>	<b>74</b>
B.1 Circuit Diagram . . . . .	74
B.2 Flowcharts . . . . .	75
B.3 State Diagrams . . . . .	78
B.4 Dependency Diagrams . . . . .	78
<b>C Test Plan</b>	<b>79</b>
<b>D User Guide</b>	<b>81</b>
D.1 BatWing.ino . . . . .	81
D.2 SerialConnection.py . . . . .	83
D.3 plotData.m . . . . .	88
<b>E Source Code</b>	<b>91</b>
E.1 BatWing.ino . . . . .	93
E.2 SerialConnection.py . . . . .	97
E.3 plotData.m . . . . .	102

# Chapter 1

## Introduction

Bats have the unique characteristic of being an airborne mammal. Their wings are constructed differently from other flying creatures. This allows them to perform impressive manoeuvres, exhibiting a high degree of complexity in their flight [16]. The bat wing has an anisotropic stiffness distribution. This distribution exhibits high stiffness strength in the chordwise direction, and a high failure strain in the spanwise direction [17]. This distribution prevents high shear forces from developing between the bones and the wing membrane. It also aids the bat in flight by storing energy during the downstroke. This important property opens up the possibility of applications in designing folding wing Unmanned Aerial Vehicles (UAVs). By mimicking the properties of the bat wing, it may be possible to increase the efficiency of UAVs, an exciting prospect from an engineering perspective. This could also have an indirect effect on the kinematics of robotic flight. Developments in this field of study would prove useful for use in law enforcement, indoor surveillance, and studies into the natural behaviour of bats and other animals to name a few. A robotic wing enables further study into the flight of bats, as it is easy to collect a range of data from a robotic wing, data that would be difficult or even infeasible to obtain when using a live specimen. Recent advancements in engineering materials, actuators and controls have allowed researchers to consider mimicking flight in the natural world [20]. There has also been a strong interest in devel-

oping drones for use in courier services, potentially decreasing delivery times significantly [4, 11].

When considering flapping UAVs, particularly micro-UAVs, there is a significant need for energy efficiency. Rather than develop flexible wings that require actuation by artificial muscles, the aeroelastic properties of the bat wing can be applied to UAVs in order to contribute to efficient flight. Wing folding reduces the energetic cost on the upstroke of a wing in flapping flight [13]. However, this reduction in the energetic cost may be surpassed by the extra load of the the actuating mechanism required in order to realise this motion. The passive properties of the bat wing membrane opens the possibilities to reduce the number of motors required to actuate the wing, thus reducing the weight of the UAV.

There have been a number of advancements in the study of bat flight, most notably the design and characterisation of a 3D printed articulated robotic bat wing [3]. This study highlights the benefits of wing-folding, a useful technique in energy conservation. The study also addresses the role that wing-folding plays in the aerodynamic qualities of the wing.

Different species of bats have different properties, such as their wing bone structure [19]. The project will focus on the *Cynopterus brachyotis*, a species that has been used in a number of studies [3, 14, 19, 20]. The effect of altering the oscillating frequency of the wing and its effect on the wing-folding properties of the robotic wing will be investigated.

A prototype of a biologically inspired UAV imitating the hovering flight of humming birds and dragonflies shows great potential in developing a high performance flapping micro-UAV [5]. There have been previous attempts to develop a bat inspired UAV, such as BATMAV [6]. Here the authors aim to develop flexible wings actuated by artificial muscles, rather than exploiting the aero- elastic properties of the bat wing to improve the efficiency of flight.

# Chapter 2

## Background

### 2.1 Literature Review

The field of biomimetics seeks to discover ways to apply properties and behaviours found in nature to the engineering and science disciplines. The bat wing is an intricately complicated structure [15]. For example, the wingbeat of a bat is far more articulated than one of a bird or an insect [20].

There is much to be learnt from flying animals. However, in regards to the flight of hummingbirds, the mechanics of flight are not understood very well [1]. The same can be said about bats. Factors affecting flight are not limited to the wing as a whole but to specific parts of the wing, from the skeletal design of the wing [18] to the wing membrane [17].

#### 2.1.1 Forces

“The moment produced by a uniformly distributed pressure load” that is needed to sustain the weight of a flying animal is given as:

$$M_{fluid} = m_b g L/4 \quad (2.1)$$

where  $m_b$  is the “mass of the body”,  $g$  is the “earth’s gravitational acceleration”, and  $L$  is the “length of the wing” [8]. Symbol definitions for the following

equations are given in Table 2.1.

When calculating the total moment to flap a wing through a gaseous fluid of a given density  $p_w$ , the following formula can be used:

$$\begin{aligned} M_{oscil} &= \int l^2 A dl p_w \Theta \omega^2 \\ &= p_w \Theta \omega^2 A L^3 / 3 \\ &= m_w \Theta \omega^2 L^2 / 3 \end{aligned} \quad (2.2)$$

where  $l$  is the position along the wing,  $A$  is the area,  $\Theta$  is the stroke amplitude,  $\omega$  is the frequency,  $L$  is the wing length, and  $m_w$  is the total mass of the wing [8].

Therefore they conclude that the flapping moment to sustain the weight of a given flying animal is:

$$R_w = (m_w/m_b) 4 \Theta \omega^2 L / 3g \quad (2.3)$$

where  $m_w$  is the total mass of the wing,  $m_b$  is the mass of the body,  $\Theta$  is the stroke amplitude,  $\omega$  is the frequency,  $L$  is the wing length, and  $g$  is gravitational acceleration [8].

As the experiments will be carried out on a stationary body with an oscillating wing, it is not necessary to consider equations 2.1 or 2.3. However equation 2.2 can be used when calculating the moment applied during the upstroke and down-stroke of the wing.

It may be desirable to determine forces during wing oscillation at any one location along the wing at a given time. Providing the motion is sinusoidal, it is possible to find the flapping moment at a given position along the wing [8]. This is achieved using:

$$\begin{aligned} dM_{oscil}(l, t) &= l A(l) dl p_w a(t) \\ &= l^2 A(l) dl p_w \Theta \omega^2 \sin(\omega t) \end{aligned} \quad (2.4)$$

where  $l$  is the position along the wing,  $A(l)$  is the local cross-sectional area,  $p_w$  is the wing density,  $a(t)$  is local tangential acceleration,  $\Theta$  is the stroke

amplitude, and  $\omega$  is the frequency [8].

<b>Symbol</b>	<b>Definition</b>	<b>Symbol</b>	<b>Definition</b>
$m_b$	body mass	$m_w$	wing mass
$g$	gravitational acceleration	$L$	wing length
$p_w$	wing density	$l$	position along wing
$a(t)$	local tangential acceleration	$\omega$	frequency
$A(l)$	local cross-sectional area	$\Theta$	stroke amplitude

Table 2.1: Symbol definitions [8]

### 2.1.2 The bat wing skeleton

Researchers at Brown University designed and 3D printed a robotic bat wing, based on the Cynopterus brachyotis. The specification of this wing is detailed in Table 2.2. The bone structure of the wing must be able to withstand the strains and changes in force associated with flapping flight. Shear stresses are “significantly greater” in bats when compared to other mammals, such as dogs [17]. Shear stress is given by:

$$\tau(y) = \frac{F}{A} \quad (2.5)$$

where  $F$  is the force applied, and  $A$  is the cross-sectional area that the force is initially applied over.

In order to resist such stresses, it is important to consider the bone structure when designing the bat wing. This idea is supported by experimental results on a ABS printed robotic bat wing. It was found that wrapping steel cable around the elbow to emulate a ligament was sufficient in preventing the wing from breaking at that particular location [3].

### 2.1.3 The bat wing membrane

The bat wing is made up of a number of regions: the propatagium, uropatagium, plagiopatagium and dactylopatagium. For the purposes of simplification and reducing complexity when considering a robotic bat wing, one continuous sheet of material can be used across all of these regions. This allows the flexibility to customise these individual areas as appropriate based on their

<b>Bone</b>	<b>Length (mm)</b>	<b>Bone</b>	<b>Length (mm)</b>
Humerus	37.2	Digit IV	
Radius	68.1	Metacarpal	42.61
Digit I	10.0	Proximal phalanx	22.1
Digit II		Distal phalanx	24.8
Metacarpal	29.7	Digit V	
Proximal phalanx	6.9	Metacarpal	40.7
Middle phalanx	4.1	Proximal phalanx	20.2
Distal phalanx	2.6	Distal phalanx	22.9
Digit III		Leg	
Metacarpal	39.9	Femur	26.2
Proximal phalanx	29.1	Tibia	31.5
Distal phalanx	37.9		

Table 2.2: Specification of a robotic bat wing [3]

individual properties. This customisation can be achieved by using an elastic network layered on top of this membrane material. When considering material for the base membrane, a number of factors must be considered. One important factor is the thickness of the membrane. Bat skin is remarkably thin, having a thickness of approximately 0.2mm. In previous designs, a wing membrane with a thickness ranging from 0.04mm to 0.2mm was used [7]. They also note that the membrane can “elongate to as much as 400% of the resting length”. Synthetic membranes have proven to be highly susceptible to tearing, especially at the edges, near the body, and at “attachment sites near the joints” [3]. Bat wings are exposed to strong bending forces due to the tension of the membrane [12]. The edges of a bat wing are reinforced using elastin and similar materials [10]. Similarly, a synthetic wing can be reinforced by a synthetic alternative, such as elastic. In order to reduce the shear stress at the joints, bats have loose connective tissue between the membrane and the underlying structure [3]. This allows the skin to have a degree of freedom in movement from the bone. This can be mimicked using a network of elastic fibres between the bone and the membrane, allowing the wing a greater range of motion [3].

When considering a synthetic material for the wing membrane, it is important to consider properties such as elasticity, but also strength. One such material that exhibits both of these properties is latex. Natural latex has a

number of advantageous properties from being extremely stretchable to being able to resist tearing to a considerable degree. In addition to this, it is possible to manufacture latex to a thickness consistent with the thickness of the wing membrane of the bat. However, taking into consideration any potential shortcomings of a synthetic material in comparison to bat skin, and considering previous experimental results using synthetic materials [3], it may be advisable to take a conservative approach to the thickness of the material being used. A fibre-reinforced poly(2-hydroxyethyl methacrylate) (pHEMA) membrane is also a potential candidate. This composite has a higher strength than pure pHEMA [23]. They claim that the maximum strength of these membranes are “17 times that of the pure pHEMA” when reinforced by “G fabrics”, and that the “elongation at break increases by a factor of 11.5” when reinforced with “S fabrics”. These properties are satisfactory for modelling the bat wing membrane. One drawback to using a fibre-reinforced pHEMA membrane is its availability; it is not readily available for purchase, and production would likely be costly. Polydimethylsiloxane is a material that is available in sheet form, and in the required thickness. However, previous experimental results have shown that this material is prone to tearing when used as a bat wing membrane [3]. Therefore, using this material would likely pose a number of challenges.

The bat wing membrane has a large stretch factor. However, it still retains its anisotropic behaviour [7]. In order to emulate the anisotropic properties of the membrane, an elastic network can serve as an intermediary between the skeletal structure and the wing membrane [3]. This elastic network will not be sufficient however, as a separate elastic network will be required to give the wing its anisotropic properties.

The bat wing membrane has an “inherent microstructure” that enables the bat to exhibit its remarkable flight behaviour [9]. This research is a fantastic introduction to the properties of biological materials, in particular materials that display anisotropic behaviour. The authors are able to relate information about biological tissues such as the aortic valve [5] to bat wing tissue in order

to better understand its mechanical properties [9]. The fibres in the elastic network tense during muscle contraction, thus tensing the wing membrane [10]. This observation is a key factor in the investigation, as it shows that this elastic network has a significant effect on the behaviour of the wing membrane. In addition, there is “an additional family of fibers”, a family separate from the dominant fibre family, elastin. These are muscle fibres, which exhibit both “passive and active properties” [22]. The existence of passive properties in fibre families reveals the potential for further research. This research also shows that bat wing tissue exhibits variations in fibre size. This can be taken into consideration when implementing an anisotropic stiffness distribution for a robotic wing. The bat wing has a number of large muscle bundles that “anchor the wing membrane to the bone”, controlling tension [17]. Because of the interaction with the membrane, it is necessary to emulate this behaviour, replacing the large muscle bundles with a passive means to control tension. This can be achieved by making adjustments to the elastic network in order to incorporate some of the properties of the muscle bundles into the elastic network.

#### 2.1.4 Wing folding

It has been proven that wing folding reduces the cost of flapping [3, 13]. It was also shown that the average inertial cost of the upstroke “was roughly equivalent” to the cost of the downstroke [14]. Experimental results conducted on a robotic wing required up to 50% less power during the upstroke, and that keeping the wing extended during upstroke produces both “substantial negative lift and positive thrust” [3]. This folding was programmed as a sinusoidal motion, where maximum folding occurs at mid-upstroke. However, this was not achieved passively.

It has been suggested that “the wing membrane itself may regulate the patterning of wing muscles” [21]. This indicates the importance of the wing membrane towards flight, especially when considering the relationship with the wing muscles. This relationship can be expanded by making the role of the

wing muscles reliant on the wing membrane. For the robotic wing, the elastic network of the wing membrane will serve as pseudo-muscles in the absence of an actuating mechanism. It has been observed that a “combination of elastin and collagen in a fibrous net” should create wing properties that affect flight [10]. There are a number of equations that can be used to estimate the cost of moving a rigid wing [14]. It may be advantageous to use the following equation to calculate the wing angle  $\gamma$  at a particular point in time  $t$ :

$$\gamma(t) = \frac{1}{2}\phi \times \cos(2\pi ft) + \bar{\gamma} \quad (2.6)$$

where  $\phi$  is the arc of amplitude,  $\bar{\gamma}$  is the average angle and  $f$  is the frequency.

Bats that have been studied are able to cut the inertial cost of flapping down to 65% due to the wing folding techniques that they employ, for example, having a span ratio of approximately 0.3 - 0.4 [14].

### 2.1.5 Flapping flight

Studies on two sub-orders of bats have shown that maximum values for wing-beat frequency range between 4 Hz - 13 Hz, and between  $90^\circ$  to  $150^\circ$  for wingbeat amplitude [6]. They also provide equations to predict the wingbeat frequency and amplitude for the flight of a bat. To calculate the wingbeat frequency:

$$f_w = 5.54 - 3.068 \log_{10} m - 2.857 \log_{10} V \quad (2.7)$$

where  $m$  is the mass of the bat in kilograms, and  $V$  is the flight speed in metres per second. To calculate the wingbeat amplitude:

$$\Theta_w = 57.92 + 5.18V + 16.06 \log_{10} S_{REF} \quad (2.8)$$

where  $V$  is the flight speed in metres per second, and  $S_{REF}$  is the wing area in  $m^2$ . A number of options can be considered when determining the best way to control the frequency at which a robotic wing oscillates.

## 2.2 Control Mechanism

### 2.2.1 Pulse-width modulation

A pulse-width modulation motor speed control switch can be used to adjust the flapping frequency of the wing. In contrast to an analogue signal where the value changes continuously, a digital signal is comprised of discrete values. Pulse-width modulation controls the pulse duration of a non-sinusoidal wave in order to control the power sent to the motor. This duration expressed as a percentage is known as a duty cycle. Each pulse wave can be expressed with a Fourier series expansion:

$$f(t) = \frac{\tau}{P} + \sum_{n=1}^{\infty} \left[ \frac{2}{\pi n} \sin\left(\frac{\pi \tau n}{P}\right) \cos\left(\frac{2\pi n}{P}t\right) \right] \quad (2.9)$$

where  $P$  represents the period and  $\tau$  represents the pulse time. This technique enables the speed of the wing to be controlled precisely, and any effects that minute adjustments have on the force output of the wing can be observed, as well as any other factors. Pulse-width modulation units often come as stand alone units with an adjuster for manual control. While this solution is straightforward, it sacrifices some of the precision that is associated with using a fine tunable control system, such as using a piece of software that is capable of controlling the speed of the motor. This may be a suitable option for preliminary testing, especially when developing the wing itself. However, it may be unsuitable when collecting data for analysis unless data can be collected from the unit and the motors simultaneously.

### 2.2.2 Potentiometer

Potentiometers can be compared to implementing two resistors in series. They are used to provide an adjustable way to control the output voltage to a device.

This concept can be explained using Ohm's law:

$$I = \frac{V}{R} \quad (2.10)$$

Thus, for a potentiometer with voltage source  $V$  and resistors  $R_1$  and  $R_2$ :

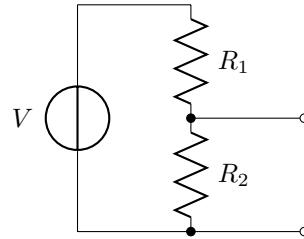


Figure 2.1: Circuit diagram for a potentiometer

The current in amps can be obtained using:

$$I = \frac{VR_2}{R_1 + R_2} \quad (2.11)$$

One limitation to using a potentiometer is that they are not generally used to control any significant levels of power. This may be insufficient when considering a larger motor, requiring more power. However, they are easy to use and easily accessible, and can prove useful during the prototyping phase of development.

### 2.3 Actuating mechanism

There are a number of approaches to actuating the flapping motion of the wing. When considering these approaches, it is necessary to consider equations 2.7 and 2.8 to ensure that every solution can meet these specifications. For a wing shoulder with diameter  $d$ , the circumference of the shoulder can be calculated as  $c = \pi d$ . So, for a actuating crank of length  $c$ , the degree of motion  $d$  of the wing can be determined using:

$$d = \frac{l \times 720}{c} \quad (2.12)$$

Possible approaches involving a motor can be divided into two main techniques, alternating rotation and continuous rotation.

### 2.3.1 Alternating rotation

This is the easiest mechanism to actuate. This is due to the fact that it will not be necessary to develop a mechanism to translate the rotation of the motor into flapping flight. Using this method, a chain or belt can be attached to the motor spindle, which is then attached around the circumference of the wing shoulder. A groove can be cut around the circumference of the shoulder to accommodate the cable. However, this approach requires that the motor is capable of sudden changes in direction, and can achieve this with a high enough precision. Realistically it would be difficult and infeasible to source a motor capable of these demands however.

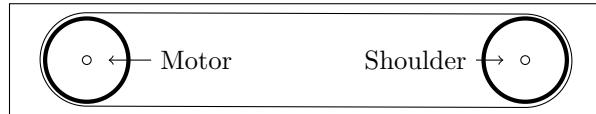


Figure 2.2: A system supporting alternating rotation

### 2.3.2 Continuous rotation

This approach requires a mechanism to translate the continuous rotation of the motor into flapping flight. This can be approached in a number of ways. One method is to attach a spring to the wing that will pull the wing downwards. The wing then rests on a oval disc attached to the rotating motor spindle. The disc pushes the wing upwards during certain periods of the motor's rotation cycle, and then the spring will bring it back down afterwards (Figure 2.3).

One major weakness of this system is that the upstroke relies completely on the properties of the spring. This means that for different flapping frequencies, the wing may exhibit an unnatural upstroke motion.

Another approach is to use a system where the motor controls both the upstroke and downstroke of the wing. A motor powers a disc with a crank

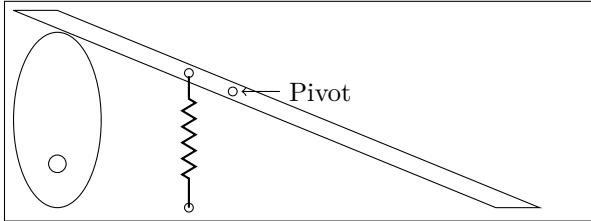


Figure 2.3: A system that uses a spring to actuate the upstroke

attached at an offset to the center. The crank is then connected to the wing (Figure 2.4).

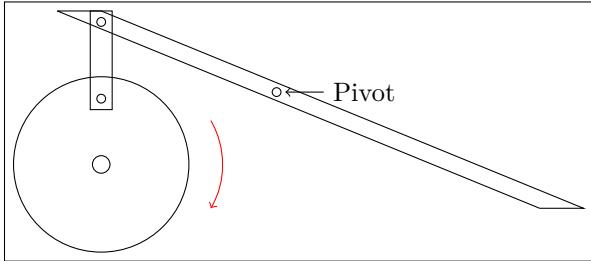


Figure 2.4: A system that uses a crank and disc to actuate the upstroke and the downstroke

The final approach is to employ a crank and cable system. The motor is connected to a gear system so as to have two output shafts. These output shafts are then connected to cranks. A cable is attached around the circumference of the shoulder, and each end is attached to a crank. A variation of this system is to employ two motors, each with one output shaft. One motor will actuate the downstroke, and one will actuate the upstroke of the wing. In either situation, it is vital that these cranks are synchronised to be directly opposite each other in relation to the output shaft of the motor (Figure 2.5).

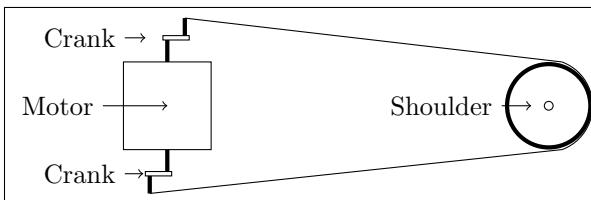


Figure 2.5: A system that uses two synchronised cranks

These approaches require less complex hardware, but also require a suitable

mechanism to be designed. The pivot system and the crank system allow for more control, as the mechanism is in complete control of the entire wing stroke cycle.

## 2.4 Measurement

There are a number of methods that can be used to collect data from the robotic wing. Each approach has its own advantages and disadvantages. It is important to employ a method that enables accurate information to be collected from the robotic bat wing. This data can aid and inspire researchers to develop new engineering techniques and perspectives, so it is imperative that error is reduced to a minimum.

### 2.4.1 Strain gauge

A strain gauge measures the strain on an object. These can be attached to the wing membrane in order to measure the strains exerted on various sections of the wing during different phases in a wing stroke cycle. The gauge has a foil track that alters the resistance across the gauge when it is stretched. The gauge factor  $F$  of a strain gauge is given by:

$$F = \frac{\Delta R/R}{\epsilon} \quad (2.13)$$

where  $R$  is the resistance and  $\epsilon$  is the strain.

Strain gauges are light and unobtrusive. In addition, they provide an accurate means of calculating the strains that different parts of the wing experience during different phases of a wing stroke. One disadvantage is having a measuring device directly on the wing. It can be difficult to predict the effect of the strain gauge on the behaviour of the wing. In addition, while the gauge itself may be unobtrusive, the wiring for the gauge may not be, and may hinder the functionality of the wing.

#### **2.4.2 Encoder**

An encoder can be used to monitor the twisting behaviour of the wing joints during oscillation. An incremental rotary encoder is capable of recording changes in joint angle to an accuracy of a fraction of a degree. This is sufficient for the experiments on the robotic wing. The encoder uses 3 bit reflected binary code to define incremental values. However, accurate encoders are often heavy and could influence the results of the experiments, and so therefore can only be used on relatively stationary parts of the wing where the encoders presence will not affect the wing stroke. One such location could be alongside the motor, where the encoder can monitor the flapping frequency of the wing. However, employing an encoder to the digit joint would likely have a great influence on the behaviour of the wing.

#### **2.4.3 Force/Torque sensor**

A force sensor alters its resistance when a force is applied to it. Such a sensor can be realised by applying a number of strain gauges to the shaft of the motor. a six-axis force sensor is able to measure force and torque along the  $x$ ,  $y$ , and  $z$  axis. It can be configured and calibrated easily, and provides accurate data. Using this sensor may pose a number of difficulties, especially if it is not feasible to incorporate the sensor into the wing flapping mechanism. The sensor requires mounting on a solid surface during operation. For a flapping wing, this is difficult, as the wing will not only move along one axis, but two or three.

#### **2.4.4 Tracing wing dynamics during flapping**

High speed cameras can record fast-moving objects, and are capable of capturing a significantly higher number of photos per second when compared to a conventional camera. This allows the recorded content to be played back in slow motion. This technique has been used to observe the flight behaviour of real bats. In one particular instance, two high speed cameras mounted on the

ground to collect kinematic information as the bats fly through their line of vision [20]. While this approach can reveal much about the distortion of the wing during oscillation, it also requires costly specialised equipment. In addition, it will be harder to derive accurate data from still photos when compared to sensors mounted directly onto the wing. In order to improve the accuracy of the data, reflective markers were placed on strategic locations on the bat [20]. It is possible to use a similar method when conducting tests on the robotic wing. A camera can also be used to capture “three-dimensional wing kinematics” in order to analyse a single wingbeat cycle [14]. A more economical alternative to a high speed camera would be to use a camera capable of HD video recording. A potentially suitable video can be recorded at 50 frames per second with a shutter speed of 1/400. The high shutter speed to frames per second ratio will help to eliminate blur, but will require a well lit environment and a higher aperture or ISO count. This technique is ideal for observing the flapping behaviour of the wing. An alternative to this is to shoot a long exposure with a small aperture and ISO count. This can be used to see the general movement pattern of the wing, but will not be suitable to examine the wing at any point in time.

#### 2.4.5 Current measurement

Using a motor driver such as the L298, it is possible to measure the current used by a motor at any one time. Since the current drawn by a motor can be used as a representation of the working effort of the motor, this measurement can be used to observe the force exerted by the motor at various points in a wing stroke.

### 2.5 Interface

There are a number of different ways to interface with the hardware. Suitable hardware is required to control and monitor the bat wing, and accuracy is of

high importance. The Arduino Uno<sup>1</sup> is a microcontroller<sup>2</sup> board with a wide range of uses. Up to six of its I/O pins can be used as PWM outputs, which makes it an ideal board to control the bat wing with. It is not only possible to write applications for it in C or C++, but also by using MATLAB<sup>3</sup>. In addition to this, Labview<sup>4</sup> has an interface to connect to the Arduino Uno. The Arduino Uno is capable of handling up to 40mA of current, which is insufficient for the motor that will be needed to power the wing. However, it is possible to control an externally powered motor with the Arduino via a motor driver such as the L298. Due to the wide range of support for different programming languages, and readily available documentation, the Arduino is a very versatile choice.

---

<sup>1</sup><http://arduino.cc/en/Main/ArduinoBoardUno>

<sup>2</sup>The Arduino Uno uses the ATmega328 microcontroller

<sup>3</sup><http://www.matlabarduino.org/serial-communication.html>

<sup>4</sup><http://www.ni.com/labview>

# **Chapter 3**

# **Specification**

There are a number of requirements for this project, and these range from user requirements to hardware requirements. Due to the nature of this project, there are varying degrees of appropriateness of using software engineering methods or other techniques to create a specification[2]. These requirements have been separated into a number of sections, each of which would likely be addressed using techniques appropriate to each field.

## **3.1 User Requirements**

It is important to provide a system that is not only efficient, but is easy to use.

The project will be approached with a number of user requirements:

- The system must be easy to use.
- The system must be efficient.
- The system must have a suitable, intuitive interface to operate and collect data from the wing.
- The user should be able to operate the system without any unnecessary distractions from the system.

## 3.2 Functional Requirements

The system should be able to fulfil its purpose. As the system will interact with experimental data, it is also important that the system is not limited to processing on particular data set. Therefore, the project will be approached with the following functional requirements:

- The system must be able to analyse the data collected from the wing and represent the data in an easy to understand manner.
- The system must be able to integrate and organise the data collected from different input sources in order to provide a clear data set on which to draw conclusions.
- The system must be able to present the data in the form of a time-series.
- The system must be able to present the data in the form of a return map.
- The system must be able to account for inaccuracies in the data set.
- The system must be able to process the data in a way that is meaningful to the user.
- The system must be accommodating to a wide range of data sets.

## 3.3 Hardware Requirements

The robotic wing must fulfil a number of requirements, in particular those related to the anisotropic stiffness distribution, and its application in robotics. The relationship between the anisotropic stiffness distribution and the flapping frequency of the wing will also be examined. The project will be approached with the following objectives:

- The robotic wing must be able to expand and contract passively during flapping flight.
- The robotic wing must have a membrane with an anisotropic stiffness distribution.

- The robotic wing must be able to utilise its anisotropic membrane to achieve efficient flight.
- The robotic wing's specification must imitate the specification of a real bat wing.

### 3.4 Other Requirements

In addition to system and hardware requirements, there are also a number of requirements regarding the analysis of the resulting data, where human analysis is more appropriate than automated analysis. These could involve data sets where it is infeasible to use software to draw meaningful conclusions on the data set, or where inconsistencies in the data set would result in human interference in order to make the data set compatible with software. Therefore, the following requirements may be appropriate for either software or human analysis:

- Analyse the wing-folding patterns for different anisotropic stiffness distributions.
- Determine a suitable flapping frequency for the anisotropic stiffness distribution of the robotic wing.
- Analyse all the data collected from various anisotropic stiffness distributions in order to reach a conclusion on their effectiveness.
- Relate conclusions about the applications of the anisotropic stiffness distribution of the bat wing to potential applications in robotics and biomimetics.

# Chapter 4

## Design

### 4.1 Interface

An Arduino Uno microcontroller board will be used to interface with the robotic wing. This allows persistent C++ code to be uploaded. This code will enable the wing to function without being connected to a computer, which may be useful for testing and demonstration purposes. Four of the Arduino's PWM output pins will be used to drive two motors. In order to communicate with the Arduino during operation, a serial connection will be established using a Python application. Python was chosen as it has a simple serial interface package, and it can also be run on multiple platforms. The Arduino does not have a significant amount of on-board memory (1024 bytes of EEPROM), so data will be collected via serial and stored locally on the connected computer. As the Arduino can only accept 40mA of current, an external driver will be used to power the wing, as will be discussed in Section 4.2.

This solution provides a finer grain control over other solutions. The ability to control the wing via software is especially useful, as it allows a higher degree of accuracy during experiments.

## 4.2 Control Mechanism

Pulse-width modulation via an L298 motor driver will be used to control the speed of the wing (Figure 4.1). This is advantageous as not only can the speed be specified precisely (in 1/255 increments), but the driver also allows motor current to be measured in real-time. The L298 motor driver is a high voltage and high current driver capable of accepting two inputs and controlling them independently. It also features two sensor outputs, allowing current usage information to be collected in real-time. This then allows conclusions to be drawn about the workload of the motors at any given point in the wing stroke based on the current readings.

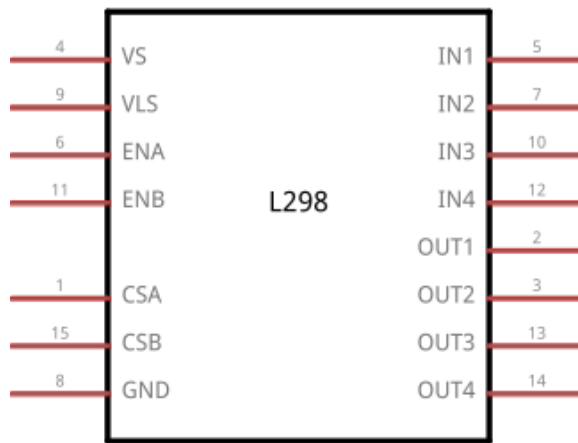


Figure 4.1: The L298 dual full bridge driver <sup>1</sup>

## 4.3 Measurement

In addition to using the sensing capabilities of the L298 motor driver, a digital camera will be used to capture the movement of the wing. Video recorded at 50 frames per second will be used to observe the general behaviour of the wing. Long exposure photographs will aid in observing the folding behaviour of the wing in higher detail. One issue with long exposure photography is capturing a moving object while reducing interference from other light sources. In order to produce an accurate image, a small LED will be attached to the tip of the

wing, allowing the long exposure photograph to be taken in darkness. This way, the path of the LED will be captured in the photograph with minimal interference from its surroundings. Different long exposure photographs of the wing flapping at different speeds will be compared, with a photograph of the wing flapping at a speed low enough so as wing-folding does not occur used as a baseline measurement. Conclusions can then be drawn about the effectiveness of the wing membrane's elastic network using these photographs.

## 4.4 Actuating Mechanism

A continuous rotation method can be employed to actuate the flapping motion of the wing. This solution will use a crank and cable system (Figure 2.5), as it provides a good degree of control. It is also easily adjustable, which will be useful for future variations on the actuating mechanism. The primary motivation for choosing the crank system over the pivot system is the design for the bat wing. There is limited space behind the wing for an extension of the wing to move. In order to accommodate the full range of motion necessary for the wing using this mechanism, it will be necessary to divide the mounting plate into two separate pieces so that the extended arm can move up and down fully.

The shoulder will have a diameter of 29mm. The circumference of the shoulder is  $c = \pi \times 29$ , approximately 91.1mm. Therefore, for a 15mm crank, the degree of motion can be calculated using Equation 2.12 as:

$$d = \frac{15 \times 720}{91.1} \quad (4.1)$$

giving approximately  $118^\circ$  of motion.

Using equation 2.7, it is possible to estimate the maximum revolutions per minute (rpm) that the actuating motor will need to operate at. Given a 50 g

(0.05 kg) bat travelling at 1 m s<sup>-1</sup>:

$$\begin{aligned}
 f_w &= 5.54 - 3.068 \log_{10} 0.5 - 2.857 \log_{10} 5 & (4.2) \\
 &\approx 4.466 \text{ Hz} \\
 &\approx 267.996 \text{ rpm} \\
 &\approx 268 \text{ rpm}
 \end{aligned}$$

Based on previous findings, it is not realistically necessary to test wing-beat frequencies higher than 4 Hz [6]. Using the relationship between rpm and hertz:

$$rpm = Hz \times 60 \quad (4.3)$$

the absolute maximum rpm of the actuating shaft is specified as 240 rpm. However, the wing will be tested at relatively low speeds in order to aid observation, and so it is not necessary to achieve those speeds..

Two 130-size 6v DC brushed motors will power a modified Tamiya 70097<sup>2</sup> dual motor gearbox. This gearbox offers a 58:1 gear ratio. The DC brushed motors are capable of speeds of up to 11500 rpm. At this gear ratio, the effective rpm  $rpm_e$  is sufficiently high:

$$\begin{aligned}
 rpm_e &= \frac{11500}{58} & (4.4) \\
 &\approx 198
 \end{aligned}$$

The two axles will be altered in such a way as to produce two synchronised output shafts. The crank will be attached to the motor shaft, with a nylon bolt on the other end of the shaft. The cable is attached around the circumference of the shoulder, and each end will have a loop which is attached to the bolt on the crank. The loop will be affixed to the bolt so as to be able to freely rotate around the bolt. The actuating wire will be routed through a barrier with holes drilled into it so as to avoid any wires being caught in the mechanism (Figure 4.2).

---

<sup>2</sup><http://www.tamiya.com>

As discussed in Section 2.3, the crank and cable continuous rotation system provides numerous advantages over other potential candidates. Among its advantages is the possibility of control throughout all stages of the wingbeat cycle, as opposed to a system that, for example, uses a spring to actuate part of the cycle.

By using a dual motor gearbox, we can observe the power usage of the upstroke and down-stroke wing motion easily. This will allow us to draw more effective conclusions when analysing experimental data.

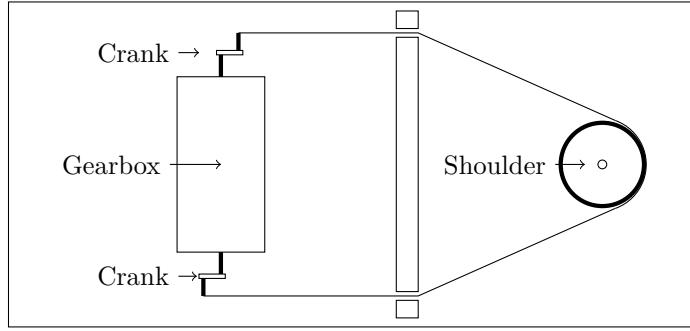


Figure 4.2: A system that uses two synchronised cranks

## 4.5 Wing Design

There were a number of options available in designing a wing skeleton. Two of the most important factors to consider are materials and complexity. Ultimately there must be a trade off between these two factors, but it is imperative that any trade off does not impact the functionality of the design in any significant manner.

### 4.5.1 The Bat Wing Skeleton

The bat wing has been shown to be highly complex [12]. Therefore it is necessary to apply a level of abstraction to the construction of the bat wing skeleton in order to identify the important artefacts for replication. One possible solution is to use plastic tubes with properties similar to the skeleton of a bat wing. However, pre-manufactured plastic tubes do not accommodate for the intricate

structure of a new key bones in the bat wing. Another option is to 3D print the major bones of the wing. As it would be advantageous to create 3D models of the wing before implementing a particular solution, 3D printing is a viable option. This method has been successfully used previously in the design and characterisation of a robotic bat wing [3]. Resources are available to 3D print the wing. An increasing number of commercial companies are now also able to provide 3D printing services. The wing will be modelled on the dimensions of the *Cynopterus brachyotis*, the design is based on a previously 3D printed robotic bat wing [3]. The wing will be designed using Solidworks 2013<sup>3</sup>, and printed using a 3D printer. The material for the bat wing skeleton will be ABS plastic, in line with the previous design. A number of simplifications can be made to the actuating mechanism. The stretching and collapsing behaviour of the bat wing should be achieved passively. So rather than powering the wing using three servo motors, a single actuating mechanism will be used. This mechanism will control only the flapping frequency of the wing. This is because the elastic properties coupled with the anisotropic stiffness distribution of the wing will allow for the wing to extend during the down-stroke, and fold during the upstroke in a completely passive manner. This means that the design that the wing is based on can also be simplified. In the original design, three cables were used to allow the wrist joints to flex synchronously with the elbow, as well as cables to actuate the humerus and radioulna [3]. A single cable will be used to control the movement of the shoulder.

This specification is also used when considering radii for the robotic wing joints. However, there are a number of considerations to factor into the specification, as allowances must be made for manufacturing error. As seen in Table 4.1, 0.5mm clearances have been specified for the joints. This is evident in “Humerus Elbow”, “Radioulna Elbow” and “Radioulna Elbow Clearance”. This is to ensure that the joints have easy movement. A number of minor adjustments have been made to the given specification. A detailed specification of each part of the bat wing can be seen in Appendix A.

---

<sup>3</sup><http://www.solidworks.co.uk>

<b>Joint</b>	<b>Radius (mm)</b>
Outer Shoulder	14
Inner Shoulder	10
Humerus Elbow	6.5
Radius Ulna Elbow	6.0
Radius Ulna Elbow Clearance	7.0
Radius Ulna Digit	9.0
Wrist-digit I and V	8.0
Wrist-digit II and III	6.5
Wrist-digit IV	6.4
Leg	4

Table 4.1: Radii of the robotic bat wing joints

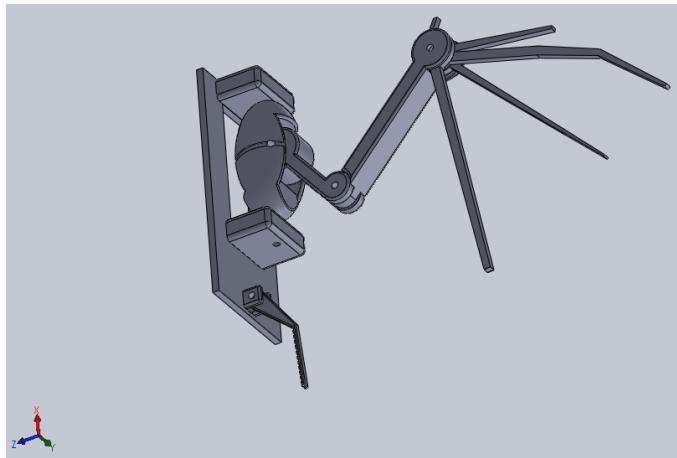


Figure 4.3: Robotic bat wing designed in Solidworks 2013

#### 4.5.2 The Bat Wing Membrane

0.25mm latex sheeting will be used to mimic the elasticity of the bat wing membrane. Latex is readily manufactured in sheet form, and would thus serve as an accurate and reliable material to use. An elastic network will be attached to the membrane to reproduce the anisotropic stiffness distribution of a bat wing. The latex sheeting will not be attached directly to the skeleton, this allows a number of elastic networks with different anisotropic stiffness distributions to be produced and attached to the wing interchangeably with relative ease. This solution is favourable as it is straightforward to use while preserving the quality of the solution. The membrane will also have an intermediary elastic network between itself and the skeleton in order to give the membrane a degree of freedom in movement from the skeleton. In order to

prevent negatively influencing the anisotropic properties of the membrane, it will closely resemble the actual fibre bundles in a bat wing [17]. Figure 4.4 shows an example of such a network. Towards the tip of the wing, the elastic threads converge towards the location of the wrist joint, much like the layout of fibre bundles of a bat wing [17]. In fact, the vertical elastic threads will aid in giving the wing its anisotropic behaviour, as the bat wing membrane exhibits its highest stiffness strength parallel to the fifth digit, in the chordwise direction [17].

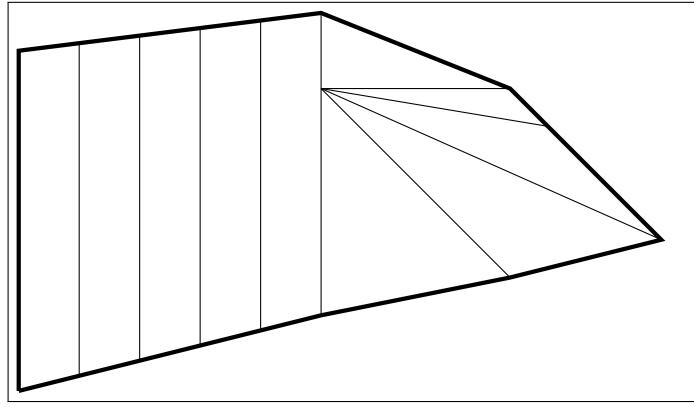


Figure 4.4: An intermediary elastic network between the wing skeleton and the membrane

#### 4.5.3 Wing folding

There are a number of mechanisms that can be employed to realise wing folding. However, not all of these options are completely passive. One such mechanism involves taking advantage of the position of the wing in relation to the stationary body. Elastic from the wing membrane is attached onto the mounting board, wrapping it around the underside of the shoulder. Figure 4.5 shows this configuration. The elastic is fixed to the mounting board so that its elastic potential alters during wing oscillation. The elastic relaxes on the down-stroke, allowing the wing to extend outwards. This allows the wing to maximise its surface area, increasing up-thrust on the down-stroke. The third diagram in Figure 4.5 shows the effects of the elastic configuration on the upstroke. The elastic configuration contracts, forcing the wing to fold inwards. This conserves

energy on the upstroke. However, this mechanism is not passive. Although the elastic potential can be seen as a passive result, the primary motion is an active one.

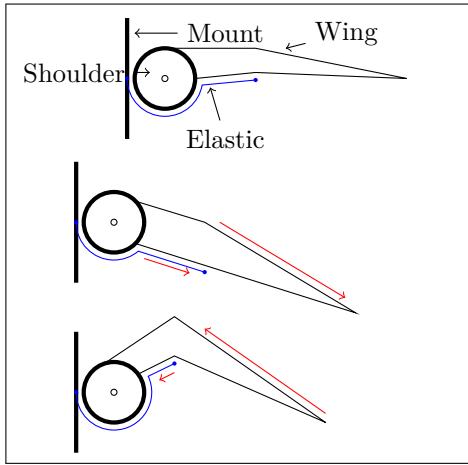


Figure 4.5: The effect of elastic threads mounted to the body on wing-folding at different points in a wing-stroke. This system involves a degree of active wing-folding

A completely passive mechanism relies solely on the up-thrust generated by the down-stroke to expand the fold the wing. Here the elastic network folds the wing when it is stationary. As the wing flaps, the up-thrust on the down-stroke expands the wing outwards. On the upstroke, the elastic potential of the wing membrane will pull the wing inwards, folding the wing. This requires a strong enough down-stroke in order to extend the wing. The advantage of this mechanism is that it is completely passive, independent of any actuating motion. Any elongation and folding of the wing will be a result of up-thrust generated by the wing.

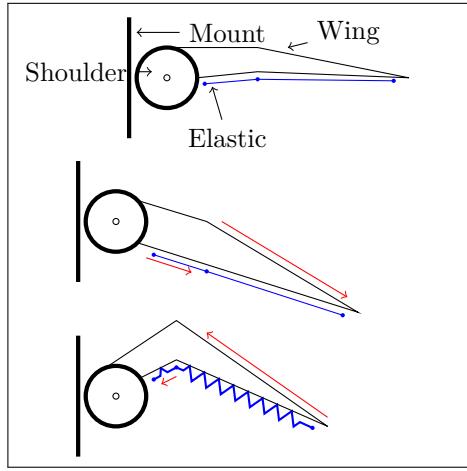


Figure 4.6: The effect of an elastic network on wing-folding at different points in a wing-stroke. The elastic network is attached to the underside of the wing membrane. Contraction of the network causes the wing to fold

## 4.6 Software

### 4.6.1 Programming Language

A number of programming languages were considered for suitability. A decision on languages to use were based upon specific needs and restrictions. Firstly, a programming language is needed to write the application to actuate the wing. C++ is an intermediate level language with both high-level and low-level features. It is supported by a wide range of systems, including the Arduino. As C is a subset of C++, these can be used together. The Arduino Integrated Development Environment (IDE) comes with built in support for C and C++. In reality, the Arduino uses the *Arduino Programming Language*<sup>4</sup>, a language very similar to the C family, and is almost interchangeable, except for a small number of specialised functions and variables. While it is possible to control the Arduino over a serial connection, this would be unnecessary for fixed wing behaviour. Arduino sketches can also be developed on third party IDEs, such as Xcode for Mac OS X or Microsoft Visual Studio for Windows.

Next, a system that will be able to collect data from the Arduino is needed. This application will collect sensor data from the Arduino in 10 millisecond

---

<sup>4</sup><http://arduino.cc/en/Reference/HomePage>

time increments. The Arduino can only persistently store this data in its Electrically Erasable Programmable Read-Only Memory (EEPROM). However, as mentioned earlier, this is limited to 1024 bytes of data, and will degrade the EEPROM. Instead, the Arduino will send the data back to the connected computer via a serial connection. Python was chosen to establish a serial connection with the Arduino in order to collect the data. Python is a high-level programming language. Its main advantage is its simplicity. One major advantage of using Python is that the code used to interface with the Arduino can be expressed in fewer lines of code than a language such as Java. Python has an intuitive module for serial communication, which simplifies the process of establishing a connection with the Arduino. Although primarily a command line language, Python also has a number of graphical user interface (GUI) frameworks, which may make the application easier to use for those less familiar with using the command line. This system will be used to log the data from the Arduino and store this data in data files as a matrix.

Finally a system is needed to analyse the data from the wing. While many programming languages such as Java offer useful tools such as graphs and tables, they are often cumbersome to implement, and lacking in versatility. Therefore a language that is specially tailored towards data analysis was chosen. MATLAB is a numerical computing environment and programming language. This will be used to process and analyse experimental data in order to draw useful conclusions.

All of these systems are designed for a single user, and thus a use case diagram is not necessary.

#### 4.6.2 3D Modelling

There are a number of applications capable to creating 3D models to be printed on a 3D printer. These include Solidworks and AutoCAD. As Solidworks is specifically designed for 3D modelling, and has the ability to create an assembly from a number of parts easily, it was chosen to model the parts for the robotic bat wing. AutoCAD is often used in industry for 2D drawings, such

as architectural blueprints. 3D designs for each part of the bat wing skeleton can be seen in Appendix A.

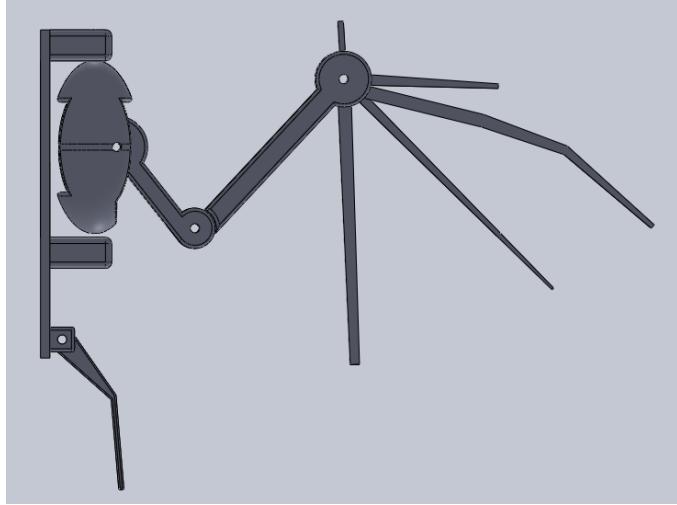


Figure 4.7: Bat wing skeleton designed using Solidworks 2013

#### 4.6.3 Interfacing With Hardware

The primary purposes of the Arduino are:

1. Operating the wing
2. Collecting sensor data from the wing

Due to the nature of its purpose, the code that will run on the Arduino's microprocessor will remain largely unchanged during its execution, and also between executions. We define a flowchart for testing different oscillating frequencies in the form of a flowchart diagram (Appendix B.2). The system increments the oscillating frequency of the wing progressively from  $f_{\min}$  to  $f_{\max}$  by a given incremental value  $i$  until the entire range of values have been tested (Appendix B.3). We also define a flowchart to oscillate the wing at a certain frequency for a given period of time. In this system,  $f$  is a frequency between  $f_{\min}$  and  $f_{\max}$ ,  $t$  is the time limit, and  $x$  is the time to pause in milliseconds.

#### 4.6.4 Serial Connection and Collecting Data

An application is required to establish a serial connection with the Arduino. This application should be accessible via both a command line interface (CLI) and a graphical user interface (GUI). The behaviour of the application in response to user actions while running in the GUI mode is defined using an interaction state machine diagram (Appendix B.5). This diagram shows the interaction possibilities when using the system, as well as navigation paths between various views in the system. The CLI does not allow the user to navigate through the system, but rather restricts the user to single operations that define all the parameters of the system. Therefore it is not necessary to define a state machine diagram for the CLI. The application will be developed using the Model-View-Controller (MVC) design pattern. This not only separates different tiers of the system, but also allows the model to send updated data in the view. The view will send any data changes to the controller, at which point the controller will handle the data appropriately, perhaps sending the data to the model. This is especially important when up-to-date information is critical for the proper running of the system. A dependency diagram (Appendix B.6) is used to show the external frameworks that the application will be dependent upon.

#### 4.6.5 Data Representation and Analysis

An application is needed to perform analysis on the robotic bat wing data. It must be capable of representing the data in a number of different forms. In addition to this, it must be able to manipulate data when necessary, such as removing anomalous readings or producing meaningful data for the user to view. We thus define a flowchart to formalise the behaviour of this application (Appendix B.4). The system will accept as inputs: the data file  $d$ , the start time in milliseconds  $s_{start}$ , the size of the data sample in milliseconds that is desired  $s_{size}$ , the low-pass filter constant  $a$ , and the type of graph  $t$ , where  $t$  can be either a return map, a time series, a graph of the upstroke motor plotted against the down-stroke motor, or a cumulative graph of current usage.

Default values will be used whenever a parameter is not provided. In order to provide flexibility to the user, these parameters can be entered in any arbitrary order, separated by parameter names. For example, a sample input of: “*my-data*, ‘filter’ , *1/5*” will operate on the *my data* matrix, taking in *1/5* as the filter value, and using default values for the other undefined parameters. By allowing the user to specify runtime parameters, the software is able to cater for a wider range of datasets and data representation possibilities. Another approach would be to implement a separate function for every type of graph, however this would result in a high degree of code reuse and would therefore not be a good solution. As this is a single function system, a class diagram or interaction state machine diagram is unnecessary.

# Chapter 5

## Implementation

### 5.1 Hardware

#### 5.1.1 The Bat Wing Skeleton

The wing skeleton is based upon the design for a robotic bat wing from researchers at Brown university [3]. The dimensions for the wing are based on the Cynopterus brachyotis. These designs were replicated using Solidworks 2013 (Appendix A), and 3D printed using ABS plastic. A number of adjustments were made to the specification used in previous research [3]. As the wing-folding mechanism is not actuated, the joint design was simplified. Originally the joint diameters were designed so that the radioulna and digits move in proportion to the movements of the humerus. As there is no longer any actuation on the humerus, this ratio system is not necessary. The actuating mechanisms for the wing were reduced to one gearbox. This gearbox controls the upstroke and down-stroke of the wing. All other wing movements are actuated during oscillation via the anisotropic membrane of the wing. This simplification meant that it was not necessary to create grooves in the joints for cables, allowing simpler and stronger components to be designed. Each component was designed individually, and then a Solidworks assembly was created from the components. This allowed any needed adjustments to the components to be made, and also made it possible to see a preview of the wing

prior to 3D printing. Components were attached together using M2.5 nuts, bolts and washers. After assembly, each joint was tested to ensure that there was sufficient movement and to check whether any adjustments were required in the models.

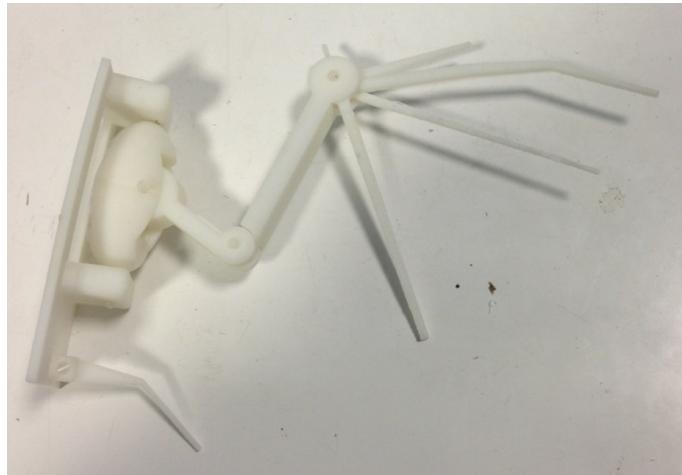
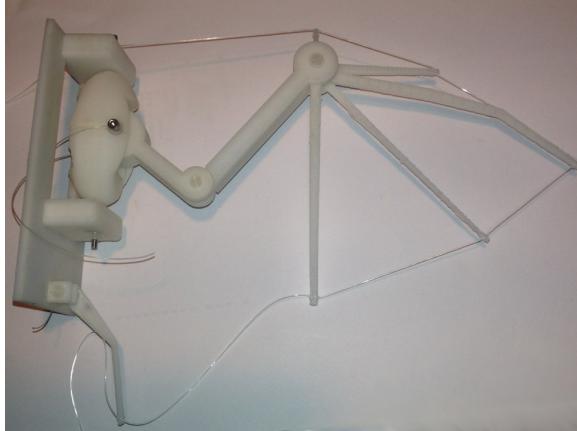


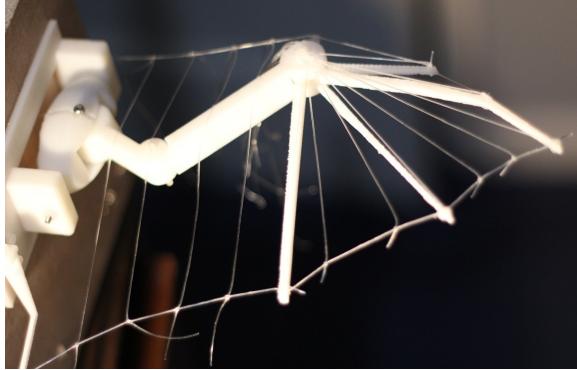
Figure 5.1: 3D printed bat wing skeleton using ABS plastic

### 5.1.2 The Bat Wing Membrane

The elastic network seen in Figure 4.4 was replicated using a network of  $0.5\text{mm}$  elastic threads (Figure 5.2).  $0.25\text{mm}$  latex sheeting was attached onto this network along the elastic threads using cyanoacrylate glue. Polyvinyl acetate glue was applied around the edge of the wing to act as a seal. Prior to implementing this system, varying concentrations of cyanacrylate glue were tested on latex sheeting. It was found that a high concentration of this glue caused the surface of the latex to bubble and deform. Therefore, when implementing the wing membrane, cyanacrylate glue was used sparingly to prevent this problem occurring. The elastic network serves a dual purpose; it not only provides support for the latex sheeting, but it also mimics the high stiffness strength in the chordwise direction of the bat wing membrane.



(a) Reinforcing elastic along the edge of the wing skeleton



(b) Elastic network implementation of the design seen in Figure 4.4

Figure 5.2: Elastic network underneath the bat wing membrane

### 5.1.3 Actuating Mechanism

A number of different prototypes were created when implementing the actuating mechanism. The prototype mechanism used a K'NEX gearbox powered by a 14.4v cordless drill<sup>1</sup> (Figure 5.3). This allowed quick and easy changes to be made to the gearbox design, reducing the time needed to determine the best way to actuate the wing. The prototype gearbox allowed a number of gear ratios to be trialed in order to determine an appropriate ratio for the robotic wing. The K'NEX prototype also served as a proof of concept for the designs that was proposed in Chapter 4 (Figure 4.2). The final mechanism and the prototypes were all tested at different operating voltages. Initial development used batteries as a source of power, ranging from 3V to 9V. The motors and the

---

<sup>1</sup>Site SMB800

sensors were also tested from no load to stall in order to ensure that the sensor readings would be compatible with the Arduino. As a precaution, the sensor leads were also tested with resistors between the L298 motor driver controller board sensor pins and the Arduino.



Figure 5.3: K'NEX Gearbox prototype

The actuating mechanism comprises of a Tamiya 70097 twin-motor gearbox with a 58:1 gear ratio. The gearbox was modified so that the output shafts are synchronised with each other. The original motors provided with this gearbox were 3V 130-size brushed motors. These motors were replaced with stronger 6V 130-size brushed motors, capable of 11500 RPM. These motors can be run at higher voltages without reducing the life-span of the motors dramatically.

The gearbox is controlled by an Arduino Uno and a L298 motor driver controller board. The circuit diagram can be seen in Appendix B.1.  $1\mu F$  capacitors were soldered to the motor terminals in order to reduce the radio interference, preventing potential damage to the components. This also protects against current surges to the motor by storing or releasing charge when necessary.

The gearbox is mounted vertically in order to simplify the pulley system used to actuate the flapping. If it were mounted horizontally, pipes or additional pulleys would be required to reroute the cables before they reached the wing. The gearbox is mounted on a sheet of primed medium-density fibreboard (MDF) (Figure 5.4).

The components are mounted on a sheet of plywood, including the MDF

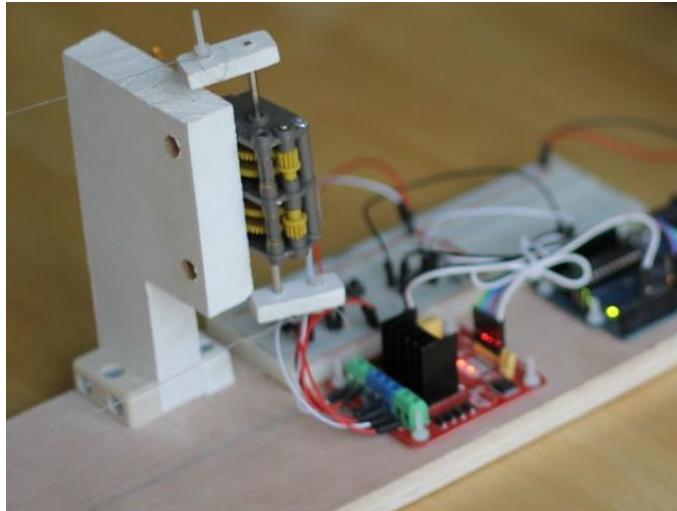


Figure 5.4: Vertically mounted gearbox

mount for the gearbox. Holes were stepped into the plywood so that the bolt heads do not protrude from the surface of the sheet.

The gearbox cranks are attached to the motor shafts. The other end of the cranks are attached to nylon bolts. These bolts attach to loops at either end of the actuating cable. This allows the cable to freely rotate around the bolt during actuation.

#### 5.1.4 Wing Folding

The principal design for the anisotropic membrane is to have elastic threads attached spanwise across the wing such that the elastic thread causes the wing to fold in its resting position. The elastic is sufficiently stretchable so as to expand when force is exerted against it. On the down-stroke, the up-thrust generated by the wing causes the wing to extend outwards, increasing the elastic potential of the wing. The taut elastic network will fold the wing back at the end of the down-stroke. The elastic network (Figure 4.4) implemented underneath the wing membrane contributes to the stiffness along the chordwise direction. This combination produces an membrane with anisotropic properties. Originally, the mechanism depicted in Figure 4.5 was implemented in order to observe the wing-folding behaviour that could be achieved with an

anisotropic membrane. However, as this is not a passive mechanism, and ultimately this technique was not used to achieve wing-folding. The design was then altered so that the elastic network is confined to the wing membrane, thus making the wing-folding behaviour passive.

A number of different elastic configurations were implemented and tests were performed to observe how changes in the elasticity of the membrane affected the wing-folding behaviour of the robotic wing. The effect of using elastic of different strengths on the extent of wing-folding were also investigated. During the experiments, motor data was captured from the wing, and long exposure photographs were taken to observe the passive wing-folding behaviour of the robotic wing, as will be discussed in Chapter 7.

## 5.2 Software

### 5.2.1 BatWing.ino

The code for the Arduino comprises of a single *.ino* file. This file is uploaded to the Arduino via a serial connection and is loaded onto the microcontroller. The flowcharts seen in Appendix B.2 were implemented as algorithms, for which the source code can be seen in Appendix E:

We define two platform dependent algorithms for the platform independent flowchart in Appendix B.3. The Arduino has two predefined functions, `setup()` and `loop()`. The `setup()` function is executed upon start-up, and is used to initialise the system. The `loop()` function then runs indefinitely, or until the Arduino is powered down or reset.

The code was implemented considering the differences between conventional C++ and the Arduino Language. This allowed the program to be written specifically for the Arduino, and so considered issues such as memory management on a microcontroller board. One method the application manages to adhere to these restrictions is by avoiding utilising the heap during execution, such as using multiple `print` and `println` commands as opposed to `cout` and `<<`.

---

**Algorithm 1** Initialising serial connection and environment

---

**Precondition:**  $f$  is the PWM value defining the flapping frequency of the wing, expressed as a fraction of 255

```
1: function SETUP( $f$ )
2:   Initialise serial connection
3:    $motorenable_{L,R} \leftarrow OUTPUT$        $\triangleright$  Set motor enable pins as outputs
4:    $motorpin_{L,R} \leftarrow OUTPUT$          $\triangleright$  Set motor power pins as outputs
5:    $motor_{L,R} \leftarrow ON$                  $\triangleright$  Power  $motor_L$  and  $motor_r$ 
6:    $sense_{L,R} \leftarrow ON$              $\triangleright$  Enable sensor pins
7:    $speed \leftarrow f$ 
8:    $motor_{L,R} \leftarrow speed$ 
9: end function
```

---

**Algorithm 2** Arduino main body loop

---

**Precondition:**  $time$  is the current time, which is initialised to 0 outside of the function.  $limit$  is the length of time the function will execute,  $delay$  is the time to wait between each loop

```
1: function LOOP( $time, limit, delay$ )
2:   if  $finished \neq True$  then
3:     Collect Sensor Readings
4:   end if
5:   if  $time < 0 \parallel time \geq limit$  then     $\triangleright$  Checks for  $time$  integer overflow
6:      $motor_{L,R} \leftarrow OFF$ 
7:      $finished \leftarrow True$ 
8:   end if
9:    $time \leftarrow time + delay$ 
10:  PAUSE  $delay$ 
11: end function
```

---

### 5.2.2 SerialConnection.py

The serial connection application was implemented using the MVC design pattern, based on the interaction state machine diagram specified in Appendix B.5. The EasyGui module served as the view of the application, while the main Python file acted as the controller. There was no need for a model as the application does not need to interface with persistent data, but rather only writes data to a file. Therefore, any required information is stored in variables within the controller, and motor data is written directly to file. The application does not read data from the file. Due to the behaviour of the EasyGui module, changes did not need to be made to the view component of the application in order to allow it to comply with the MVC design pattern. While not being

an event-driven GUI module, it provided more than sufficient functionality for the *SerialConnection.py* application. The CLI interface was implemented using flags and parameters, as opposed to command line prompts (Figure 5.5). This enables concise and easy access to the range of functionality offered by the system.

```
optlist, args = getopt.getopt(argv, 'f:gp:') //get flags and values
for opt, arg in optlist: //for each flag in the list
    if opt == '-f': //parse -f flag if available
    ...
    if opt == '-g': //parse -g flag if available
    ...
    if opt == '-p': //parse -p flag if available
    ...
```

Figure 5.5: CLI flag parsing in the *SerialConnection.py* application

The CLI will parse all flags before executing any functionality, this allows the user to enter flags in any arbitrary order without affecting functionality.

When run in GUI mode, the application will establish a serial connection, read and store motor data from the robotic wing in a separate thread (Figure 5.6). This is to allow the GUI to execute concurrently with the serial connection capabilities of the system. Without this, it would be non-trivial to utilise the GUI to close the serial connection to the robotic wing. A global flag variable `running` is used to close the connection to the robotic wing. This flag variable can be set to `False` using the GUI. Multi-threading is not

```
...
thread.start_new_thread(logData, ()) //Starts a new thread to log data
...
```

Figure 5.6: Multi-threading in the *SerialConnection.py* application

necessary when establishing a connection from the CLI. This is because the CLI is designed to be a single purpose application, and thus terminating the application will also close the serial connection.

In order to reduce the amount of code reuse, the GUI and CLI share the same code for their core functionalities. They differ in the ways they provide an interface to the user.

### 5.2.3 MATLAB

The *plotData* MATLAB function was designed with flexibility and ease of use in mind. As with a number of existing MATLAB functions, such as *plot*, the user is able to provide a list of command line arguments in any arbitrary order, providing that values are passed immediately after their corresponding parameters. For example, in order to set the plot type to a time series, a valid function call would be:

```
plotData(data, 'type', 'Time')
```

The algorithm for parsing function parameters is given in Algorithm 3.

---

**Algorithm 3** Parsing parameters in the *plotData* function

---

**Precondition:** *data* is a compliant data matrix. *args* is an array of arguments. The function expects exactly one value to follow each argument passed

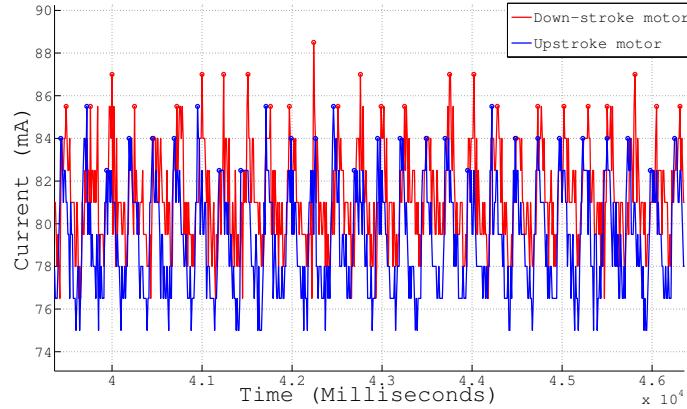
```
1: function PARSER(data, args)
2:   while i < size(args) do
3:     param  $\leftarrow$  args[i]
4:     value  $\leftarrow$  args[i + 1]
5:     if param = 'start' then
6:       start  $\leftarrow$  value
7:     else if param = 'milliseconds' then
8:       milliseconds  $\leftarrow$  value
9:     else if ... then            $\triangleright$  Checks all valid parameters
10:      ...
11:    end if                    $\triangleright$  Assigns value to parameter variable
12:    i  $\leftarrow$  i + 2
13:  end while
14: end function
```

---

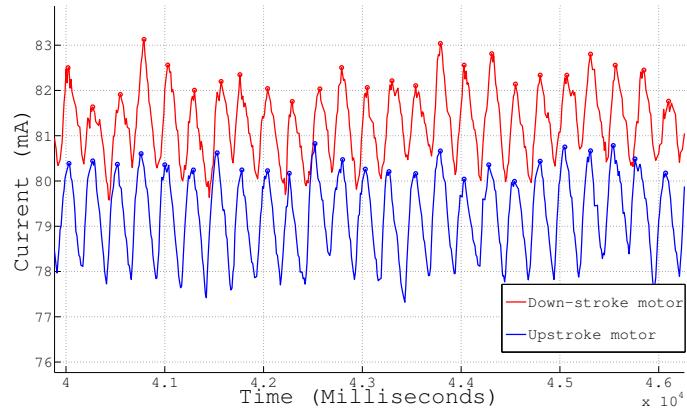
The function will use predefined default values when a parameter value is not specified when the function is called. The function will also perform error checking prior to processing the given data. This is to ensure that the parameters provided are valid, or whether a given parameter may cause unexpected behaviour. An example of an invalid parameter is one that requests that the graph is displayed in grid-position five of a two-by-two grid.

The data is subject to a low pass filter in order to reduce the noise in the motor data. Figure 5.7 shows motor data before and after a low pass filter is applied. Peaks are then detected in the data, while ignoring local optima

between the peaks.



(a) Unfiltered robotic wing motor data



(b) Robotic wing motor data from Figure 5.7a after the application of a low pass filter

Figure 5.7: Comparison between unfiltered and filtered motor data for the robotic wing

The default graph that the function will plot is a time series. In addition to this, the user is able to plot the two motors against each other, a return map, and a cumulative graph of motor data. Due to the nature of the function's design, it is straightforward to add graph plotting functionality, as the core data processing is separate from the graph plotting elements of the function.

## Chapter 6

# Professional and Ethical Issues

There were a number of issues to consider when implementing this project. Of high concern are the ethical issues surrounding unmanned aerial vehicles (UAVs). UAVs are used in a variety of real life scenarios, some of which may have ethical considerations. For example, the use of UAVs in surveillance may become unethical if the surveillance involves sensitive information or a breach of human rights. Situations such as these would not fall into the intended usage of the technology. Ultimately, the user is free to decide what the technology is used to accomplish. However, this project was approached with a number of useful applications in mind, such as surveillance in fire-fighting scenarios, where a flexible membrane would be advantageous. Much consideration has been expended in order to ensure that the end product is designed with ethical applications in mind.

All third party technologies, code and software used are declared in the report, whether commercial or non-commercial. The most relevant research currently being conducted in this field were also considered when developing this project.

# Chapter 7

# Experimental Results and Testing

## 7.1 Experimental Results

### 7.1.1 Long Exposure Photography

An LED was affixed to the tip of the wing, allowing long exposure photographs to be taken in the dark in order to observe the folding behaviour of the robotic wing at different speeds. A baseline measurement was established by flapping the wing at low speed (Figure 7.1a). The wing was then flapped at a number of different speeds in order to observe the differences in the wing-folding motion. It was observed that the wing requires a single wing-beat cycle in order to reach a consistent flapping state, as can be seen in Figure 7.1b. It is evident from this figure that the wing extends upon the down-stroke and then folds mid-upstroke. However, an extension at the top of the upstroke was also observed, similar to the motion of a real bat, albeit more pronounced. This is represented by a figure-of-eight pattern seen in Figure 7.1b. The wing then extends again during the down-stroke. This cycle, and subsequent figure-of-eight pattern appeared to remain constant throughout the experiment, even when the experiment was conducted using a longer exposure time. A possible reason for this figure-of-

eight motion are the elastic properties of the membrane. The effect of the membrane was investigated by testing a number of elastic networks allowing varying spanwise stretch.

The project seeks to discover the effects of using an elastic network that would allow for easier wing extension, as opposed to a less elastic membrane that would reduce the desired passive behaviour of the wing. An elastic network was implemented on the wing membrane that was much more flexible in the spanwise direction than the original elastic network seen in Figure 7.1b. The same long exposure photography experiments were then conducted on this wing. It was observed that the figure-of-eight pattern seen in previous experiments dramatically altered into a pattern resembling a ‘D’ shape (Figure 7.1c). Stronger wing-extension can still be seen on the latter half of the down-stroke, although the wing-extension seen at the top of the upstroke is significantly less profound.

From experimental data, it is suggested that a minimum flapping frequency of  $\sim 199\text{rpm}$  should be maintained in order to take advantage of the anisotropic properties of this bat wing membrane.

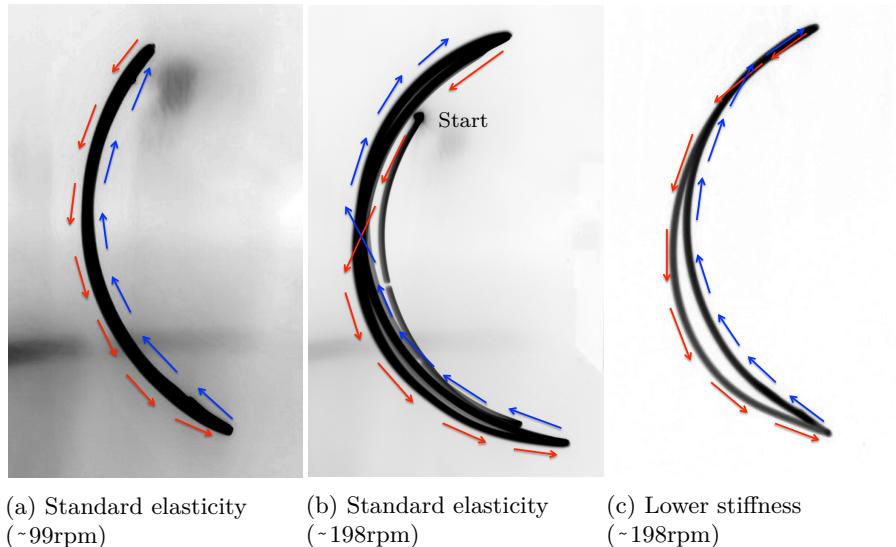


Figure 7.1: Inverted, grayscale, long exposure photographs of robotic wing flapping. a) shows the no wing extension or folding. b) shows wing extension and folding in a figure-of-eight motion. c) shows wing extension and folding closely resembling the hypothesised pattern. Down-stroke ( $\rightarrow$ ) Up-stroke ( $\leftarrow$ )

### 7.1.2 Sensor Readings

The data from the wing was converted into current readings (mA) using Ohm's law. A low pass filter was applied to the data using MATLAB where  $T = 10$  and  $\tau = 100$ . A time-series of both the upstroke (upper) motor  $\phi_U$  and the down-stroke (lower) motor  $\phi_L$  operating during passive wing-folding flight was plotted in order to observe the differences in current data. It was observed that the down-stroke motor  $\phi_L$  uses significantly more current than the upstroke motor  $\phi_U$  (Figure 7.3a). When compared to current data where no passive wing-folding occurs (Figure 7.2), it can be seen that the passive wing-folding behaviour that had been implemented made a significant contribution to efficient flight. These results were consistent across a range of flapping speeds, up to  $\sim 198$  rpm.

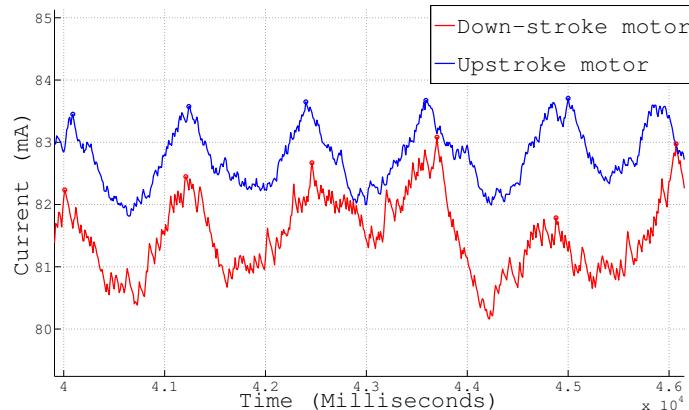
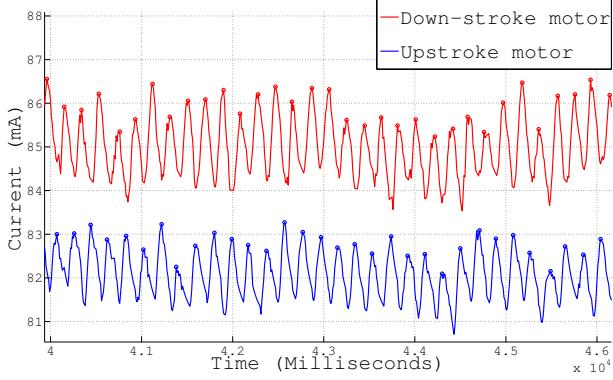


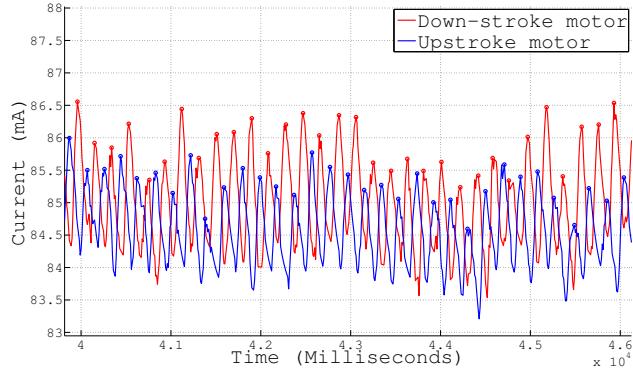
Figure 7.2: Extract of the time-series for current data without passive wing-folding. The upstroke motor generally uses more current than the down-stroke motor, indicating inefficient flight

In order to analyse the limit cycle behaviour of the motors, a return map was plotted for peaks in the data. For peaks 1 to  $n - 1$ ,  $\phi_U^P(k + 1)$  was plotted against  $\phi_U^P(k)$ , and  $\phi_L^P(k + 1)$  against  $\phi_L^P(k)$ , where  $k$  the peak number, and  $P$  represents that peaks in the data form the dataset. In order to observe whether the limit cycle of the motors settle into a steady state, the first and second half of each data set were plotted in different colours

From Figure 7.4, it can be seen that both motors settle into a steady state.



(a) Comparison of current usage between the down-stroke and upstroke motor, running at ~198rpm

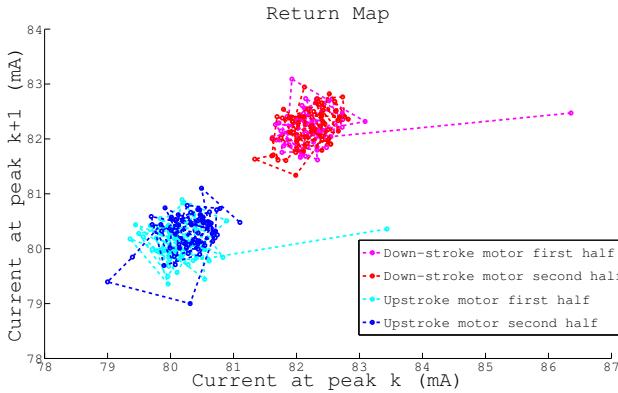


(b) Comparison of motor data from Figure 7.3a, where the upstroke motor readings have been shifted in-line with the down-stroke motor readings in order to compare the current change between peaks for each motor

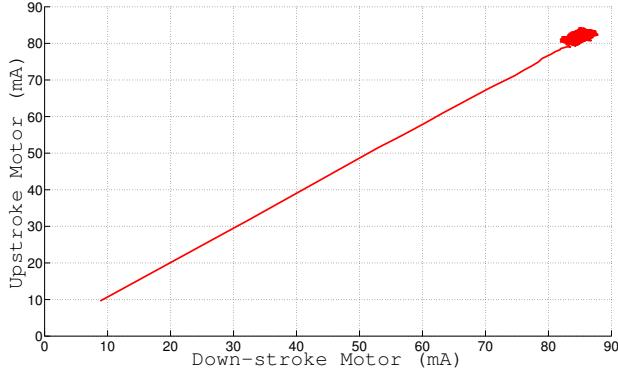
Figure 7.3: Extract of the time-series for current data after the application of a low pass filter for down-stroke and upstroke motors. The upstroke motor uses noticeably less current than the down-stroke motor, and has a lower peak when compared against each other

Figure 7.4a shows that the steady state variability of the two motor's currents. It can be seen that both motors quickly settle at a steady state. In fact, the motor current values aggregate around the steady state almost instantaneously, as can be seen by the large shift in current readings in the first half of each motor's data (— and —). Figure 7.4b shows the motor current readings tending towards the upper right of the graph, supporting the observations derived from Figure 7.4a.

Figure 7.5 shows a cumulative graph of current usage over time, for both the down-stroke and upstroke motors. This graph supports the claim that the



(a) Return map for peaks in the data for down-stroke and upstroke motors. The results show the steady state variability of the motor currents. The motors quickly settle at a steady state, as indicated by the second half of each motor's data (— and —)



(b) The upstroke motor  $\phi_U$  plotted against the down-stroke motor  $\phi_L$ . The motors settle at a steady state quickly, as shown by the congregation of values around the upper right side of the graph

Figure 7.4: Two graphs showing the steady state variability of the robotic bat wing's flapping behaviour

down-stroke motor uses more current than the upstroke motor, and indicating that the implemented passive wing-folding contributes to efficient flight. The graph also shows the this difference in current usage remains consistent across the entire experiment. The difference between the down-stroke motor's current usage and the upstroke motor's usage also appears to stay at a constant ratio. This is also indicated by the current usages depicted in Figure 7.3b and the shape of the plot in Figure 7.4b.

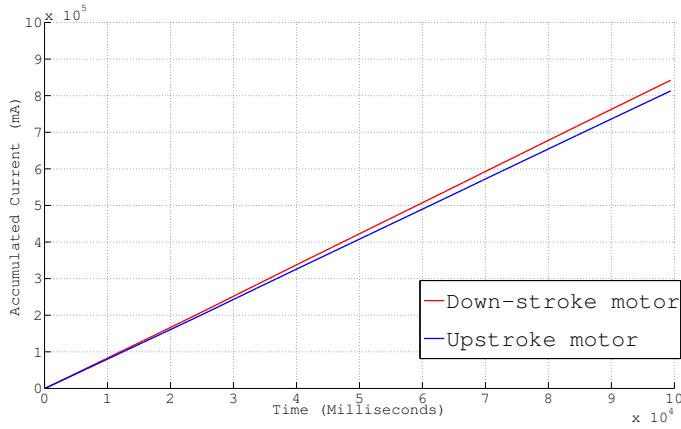


Figure 7.5: Cumulative graph depicting the current usage over time for the down-stroke and upstroke motors. It can be seen that the down-stroke motor consistently uses more current than the upstroke motor

## 7.2 Testing

### 7.2.1 Hardware Testing

The bulk of the testing carried out on the robotic bat wing is described in Section 7.1. As the expected behaviour of the robotic wing was uncertain prior to experimentation, there was not a specific set of requirements that were to be met. Rather, predicted outcomes were used to test the effectiveness of the wing. Testing mainly comprised of evaluating the effectiveness of a number of implementations, and how closely they match the predicted outcome. Then, the final implementation is tested against the original predictions. This was discussed in Section 7.1. An evaluation of the effectiveness of the robotic wing is given in Chapter 8.

### 7.2.2 Software Testing

Software testing was carried out throughout the development cycle via unit testing and user acceptance testing (UAT). Individual functions and methods were tested using white-box testing techniques to ensure that they are able to handle erroneous and extreme test data, in addition to behaving an expected when given valid test data. This helped to uncover a number of bugs, all of

which were quickly addressed. This method of testing also helped to reduce the complexity of debugging the system, especially when searching for stubborn bugs that can only be discovered by a few test cases.

System testing was used to check whether the system behaved according to the original designs, such as the user interaction state diagram (Appendix B). Every transition in the diagram was tested for compliance using, when appropriate, a range of valid and invalid data, including extreme data values. Operating the system using the default parameters were also tested, as it was possible that they were exempt from the standard error checking procedures. Black-box testing was used to ensure that the system operated as expected, without any loss of functionality. The user interface was also tested for ease-of-use, and changes were made accordingly.

# Chapter 8

## Evaluation

### 8.1 Project Evaluation

The main focus on the project was studying the role of the anisotropic stiffness distribution of the bat wing, and how it can contribute to efficient flight in a robotic flapper. Using experimental data, it has been shown that the anisotropic stiffness distribution can indeed play a role in efficient flight, but it requires a minimum flapping speed in order to achieve passive wing collapsing on the upstroke. It proved challenging to design and construct a robotic wing that was capable of emulating the wing-folding kinematics of a real bat wing using one degree of actuated motion. The robotic wing created at Brown University [3] dictated precise wing folding movements using servo motors, thus providing a consistent wing-folding pattern. Power consumption data from the wing allowed wing-beat patterns to be observed. The wing is capable of operating at a number of flapping frequencies, allowing the effect of an anisotropic membrane at different frequencies to be analysed and compared. Various different anisotropic membranes were tested, and their effect on wing-folding and extension were observed.

Originally a 6-axis force sensor was to be used to measure the force output generated by the wing. However these sensors were large and required a mounting plate in order to function properly. Mounting one of these onto the wing

would have a significant impact on the experimental data that it would collect, while mounting a sensor in another location would not prove an effective form of measurement. Therefore long exposure photography was used to show the effects of the anisotropic membrane on passive wing-folding. This technique was used as a means to determine the effectiveness of different anisotropic membranes. Initially, a figure-of-eight shaped passive wing-folding behaviour that settled at a steady state was observed (Figure 7.1b). This required the motors to run at approximately 198rpm. Further experiments at the same speed show that more significant extension-folding behaviour can be achieved using an elastic network that can be stretched easier (Figure 7.1c). At low speeds, this wing-folding behaviour was not observed (Figure 7.1a). This is a limitation of using this passive technique. Therefore, it can be concluded that the wing must maintain a minimum flapping frequency in order to exhibit its wing extending and folding behaviour effectively. However, it may be possible to design a wing that will exhibit passive wing-folding even at lower flapping frequencies. However, this robotic wing does not achieve such functionality. Ultimately, it has been shown that drawing inspiration from nature can have both practical and feasible applications in the field of robotics.

The *BatWing.ino* application was designed to initialise the Arduino Uno environment and run a continuous loop for the purposes of actuating the robotic wing and collecting motor data. It provides an efficient and reliable solution, and manages to meet the strict memory management limits imposed by the Arduino without having to change its functionality. However, the functionality provided by this application is limited due to the lack of user interaction during runtime. While this was not necessary for the scope of the project, it may be useful when considering further work. For example, the development of a UAV which can be controlled by a user would benefit from this by overriding default behaviour with user define behaviour. In order to implement this functionality, additional hardware to enable the user to communicate with the UAV wirelessly would likely be required.

The *SerialConnection.py* application was designed with ease-of-use in mind,

without affecting functionality. This was accomplished by defining and adhering to an interaction state diagram. The application has unambiguous prompts and simple menu options. Technical errors, such as *OSError*s are abstracted from the user and replaced with intuitive error messages. Despite using a high-level language, the footprint and scope of the application ensures that it remains lightweight and quick during use. Scalability is an issue when considering further development of this application. One reason for this is the Global Interpreter Lock (GIL) used by Python for multi-threading. Other mechanisms, such as the implementation used by the Java Virtual Machine are more scalable. However, switching from CPython which is written in C, to JPython which is written in Java would eliminate these scalability issues.

The *PlotData.m* function was designed to integrate into the MATLAB environment without requiring a steep learning curve while maintaining functionality. It provides useful graphs for analysis, and accommodates a range of customisation. When appropriate, the function provides easy to understand error messages that can be understood by a non-technical user. There are a number of limitations to this function. For example, the location of the legend is unspecified, although it is possible to specify a position vector to determine the location. This was considered an inconsequential issue as the legend can be moved after the graph is plotted. Also, there may be some situations where there is no ideal position to display the legend. As this is largely a user preference, and the user would likely not be able to make an informed decision prior to plotting the graph, this functionality is not provided within the function.

## 8.2 Evaluation Against Requirements

### 8.2.1 User Requirements

The system was designed and implemented with a major objective being ease-of-use, without compromising on functionality. It is deployable on multiple operating systems, and has been tested on Mac OS X and Windows 7. The system follows the design of an interaction state diagram that outlines an

intuitive interface. User interaction is also limited to the essential requirements, and when possible provides default values to reduce the number of parameters that the user is required to provide.

### **8.2.2 Functional Requirements**

The system organises, analyses and displays bat wing data in an easy to understand manner, through the use of graphs. It supports the generation of a time-series as well as a return map. A built in filter accounts for noise in the experimental data. It accepts as data any compliant three-column matrix, where the first column corresponds to the time, and the other two columns correspond to numerical data values.

### **8.2.3 Hardware Requirements**

The robotic wing achieves passive wing-folding during flapping flight. This is due to the anisotropic properties of its membrane. Experimental results have shown that this passive wing-folding contributes to efficient flight, these results can be seen in Section 7.1. It is also shown in this section that the elastic membrane has a major effect on the wing-folding behaviour of the robotic wing. The wing was designed as per the given specifications of the *Cynopterus brachyotis* [3].

### **8.2.4 Other Requirements**

Chapter 7 address the analysis of experimental data, including the wing-folding behaviour of different anisotropic stiffness distributions (Figure 7.1). A suitable flapping frequency is presented against a baseline measurement to demonstrate the effect of the anisotropic membrane, and its effectiveness is analysed using sensor readings. Chapter 9 considers further work as a result of the conclusions drawn from designing a robotic wing with an anisotropic membrane.

## Chapter 9

# Conclusion and Future Work

There have been a number of key points that have been learnt during this project. The literature review emphasises the potential for applying nature and natural behaviour to robotics. The application of nature to research fields such as robotics and artificial intelligence provides seemingly endless possibilities for innovation and advancement. The development of novel approaches to well explored topics such as the paradigm of flight can revolutionise the design of next generation technology. This is demonstrated in the research efforts in biologically-inspired robotics, in particular, soft robotics.

Throughout the project, there have been a number of issues that have arisen as a result of hardware limitations. For example, the motors that were used to power the robotic wing would have benefited with a means to precisely determine and dictate the angle of the motor shaft without requiring extra hardware, such as an external encoder. While a stepper motor would have provided a sufficient degree of accuracy, it would not provide a sufficiently high rpm to make the robotic wing flap efficiently. It would also have been advantageous to provide a means to precisely determine and dictate the elastic potential of the wing membrane. This would provide an absolute scale to

measure results with, eliminating the need to measure results in relation to each other.

Ultimately, appendages, such as a bat wing, are specialised to fulfil a particular set of tasks. It is important in robotics to limit the scope of what will be used to derive inspiration when considering biologically inspired technologies. It was necessary to make decisions to prioritise aspects of the bat wing to focus on for the purposes of research and development. Limiting the robotic wing to one degree of actuation was one such decision. Despite falling shy of perfectly imitating a bat wing, the robotic counterpart that was created manages to achieve a number of goals and objectives. Using this counterpart, it has been shown that utilising an anisotropic membrane does contribute to wing-folding in a robotic wing. Furthermore, the exact effects of an anisotropic membrane have been demonstrated via experimental results. It has also been shown that the elasticity of this membrane can be altered in order to produce different wing-folding behaviour.

The project has addressed one area of robotic flight. Further work could focus on dynamic modelling of the wing, computer simulations, novel design and optimisation techniques, the development of flight algorithms that would complement a UAV with an anisotropic membrane, and extensive testing in wind tunnels and various up-draft conditions. Flight planning algorithms would be especially useful, as additional software would not result in disadvantaging a UAV like adding additional actuating mechanisms would, simply because software itself adds no physical weight to the UAV. Intelligent algorithms would also open the way for the development of a dynamic anisotropic membrane, one that can alter its elastic properties based on real time data. Additional work could consider the effect of the elastic network on not only the behaviour of wing-folding, but also other flight properties of a flapping wing UAV. For example, the role that the elastic network plays in performing quick and precise manoeuvres. Additional work could consider a robotic wing capable of precise movement in three dimensions, as per the bat wing motion documented in previous research [20]. This work may look into programming this motion ex-

plicitly, or achieving the motion via passive means. It may be feasible to build a more complex model, substituting active actuation with passive alternatives when possible. Further study into the fabrication of an anisotropic wing membrane can explore a means of altering the elastic properties of the membrane in real-time, based on the UAV's needs or environmental conditions.

This robotic wing serves as a foundation for further research into flapping wing UAVs. It highlights some of the benefits of considering the flight of bats when designing UAVs, and demonstrates a practical application in robotics.

# References

- [1] Douglas L Altshuler and Robert Dudley. The ecological and evolutionary interface of hummingbird flight physiology. *J Exp Biol*, 205(Pt 16):2325–2336, Aug 2002.
- [2] Terry Bahill and Jesse Daniels. Using object-oriented and uml tools for hardware design: A case study, 2002.
- [3] Joseph W Bahlman, Sharon M Swartz, and Kenneth S Breuer. Design and characterization of a multi-articulated robotic bat wing. *Bioinspiration & Biomimetics*, 8(1):016009, 2013.
- [4] Leonid Bersheidsky. Bersheidsky on europe: Drone delivery tested. <http://www.bloomberg.com/news/2013-12-10/bersheidsky-on-europe-drone-delivery-tested.html>, December 2013. Last checked: 24/02/2014.
- [5] K. L. Billiar and M. S. Sacks. Biaxial mechanical properties of the natural and glutaraldehyde treated aortic valve cusp—part i: Experimental results. *Journal of Biomechanical Engineering*, 122(1):23–30, 07 1999.
- [6] R D Bullen and N L McKenzie. Scaling bat wingbeat frequency and amplitude. *J Exp Biol*, 205(Pt 17):2615–2626, Sep 2002.
- [7] J Colorado, A Barrientos, C Rossi, and K S Breuer. Biomechanics of smart wings in a bat robot: morphing wings using sma actuators. *Bioinspiration & Biomimetics*, 7(3):036006, 2012.
- [8] Thomas L Daniel and Stacey A Combes. Flexible wings and fins: bending

- by inertial or fluid-dynamic forces? *Integr Comp Biol*, 42(5):1044–1049, Nov 2002.
- [9] N. Goulbourne, Y. Wang, S. Son, and A. Skulborstad. Microstructure and material characterization of bat wing tissue for active skin composites. In *18th International Conference on Composite Materials*, Jeju, 2011.
- [10] KA Holbrook and GF Odland. A collagen and elastic network in the wing of the bat. *Journal of anatomy*, 126(Pt 1):21–36, 05 1978.
- [11] BBC News. Amazon testing drones for deliveries. <http://www.bbc.co.uk/news/technology-25180906>, December 2013. Last checked: 24/02/2014.
- [12] UllaM. Norberg. Functional osteology and myology of the wing of the dog-faced bat *rousettus aegyptiacus* (é. geoffroy) (mammalia, chiroptera). 73(1):1–44, 1972.
- [13] Ben Parslew and William Crowther. Simulating avian wingbeats and wakes. *Journal of biomechanics*, 45:S7–, 07 2012.
- [14] Daniel K. Riskin, Attila Bergou, Kenneth S. Breuer, and Sharon M. Swartz. Upstroke wing flexion and the inertial cost of bat flight. *Proceedings of the Royal Society B: Biological Sciences*, 279(1740):2945–2950, 2012.
- [15] Daniel K. Riskin, David J. Willis, José Iriarte-Díaz, Tyson L. Hedrick, Mykhaylo Kostandov, Jian Chen, David H. Laidlaw, Kenneth S. Breuer, and Sharon M. Swartz. Quantifying the complexity of bat wing kinematics. *Journal of Theoretical Biology*, 254(3):604 – 615, 2008.
- [16] Susanne Sterbing-D’Angelo, Mohit Chadha, Chen Chiu, Ben Falk, Wei Xian, Janna Barcelo, John M. Zook, and Cynthia F. Moss. Bat wing sensors support flight control. *Proceedings of the National Academy of Sciences*, 2011.

- [17] S. M. Swartz, M. S. Groves, H. D. Kim, and W. R. Walsh. Mechanical properties of bat wing membrane skin. *Journal of Zoology*, 239(2):357–378, 1996.
- [18] Sharon M. Swartz, Michael B. Bennett, and David R. Carrier. Wing bone stresses in free flying bats and the evolution of skeletal design for flight. *Nature*, 359(6397):726–729, October 1992.
- [19] Sharon M Swartz and Kevin M Middleton. Biomechanics of the bat limb skeleton: scaling, material properties and mechanics. *Cells Tissues Organs*, 187(1):59–84, 2008.
- [20] Xiaodong Tian, Jose Iriarte-Diaz, Kevin Middleton, Ricardo Galvao, Emily Israeli, Abigail Roemer, Allyce Sullivan, Arnold Song, Sharon Swartz, and Kenneth Breuer. Direct measurements of the kinematics and dynamics of bat flight. *Bioinspiration & Biomimetics*, 1(4):S10, 2006.
- [21] Masayoshi Tokita, Takaaki Abe, and Kazuo Suzuki. The developmental basis of bat wing muscle. *Nat Commun*, 3:1302, 12 2012.
- [22] Y. Wang, S. Son, S.M. Swartz, and N.C. Goulbourne. A mixed von mises distribution for modeling soft biological tissues with two distributed fiber properties. *International Journal of Solids and Structures*, 49(21):2914 – 2923, 2012.
- [23] Chung-Daw Young, Jing-Ran Wu, and Tai-Li Tsou. High-strength, ultra-thin and fiber-reinforced phema artificial skin. *Biomaterials*, 19(19):1745 – 1752, 1998.