Dynamical Modelling for a Dragonfly Micro Aerial Vehicle

by

Vidya Menon

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
University of Toronto Institute for Aerospace Studies
University of Toronto

© Copyright 2018 by Vidya Menon
Abstract

Dynamical Modelling for a Dragonfly Micro Aerial Vehicle

Vidya Menon
Master of Applied Science
University of Toronto Institute for Aerospace Studies
University of Toronto
2018

Researchers are developing ever smaller aircraft called Micro Aerial Vehicles (MAVs). The Space Robotics Group has joined the field by developing a dragonfly-inspired MAV. This thesis presents two contributions to this project. The first is the development of a dynamical model of the internal MAV components to be used for tuning design parameters and as a future plant model. This model is derived using the Lagrangian method and differs from others because it accounts for the internal dynamics of the system. The second contribution of this thesis is an estimation algorithm that can be used to determine prototype performance and verify the dynamical model from the first part. Based on the Gauss-Newton Batch Estimator, this algorithm uses a single camera and known points of interest on the wing to estimate the wing kinematic angles. Unlike other single-camera methods, this method is probabilistically based rather than being geometric.
Acknowledgements

First, my I would like to thank my supervisor, Professor D’Eleuterio, for his invaluable guidance over the years. Thank you for teaching me rigour, an eye for detail and an appreciation for the world beyond engineering.

I would also like to thank the members of my research assessment committee: Professor Damaren and Professor Grant, for their valuable insight and ideas in my committee meetings.

To my friends and colleagues at the Space Robotics Group: Pete, Adam, Adam, Alex, Anton, Paul, Dave, and Zain, thank you for your friendship, support and commiseration throughout the years.

To Jon: thank you for your unfailing encouragement and endless support.

And finally to my family, Mom, Dad and Priya: without you none of this would have been possible.
# Contents

Acknowledgments iii

Table of Contents iv

List of Figures vii

Acronyms x

Notation xi

Introduction 1

I Dynamical Model 4

1 Literature Review 6

1.1 Current and Past Insect-Inspired MAV projects 6

1.2 Dynamical Models for Flapping-Wing Micro Aerial Vehicles (MAVs) 9

1.3 Insect Flight and Unsteady Aerodynamic Models 10

2 Model Derivation 14

2.1 Actuator-Transmission-Wing Model 14

2.2 Aerodynamic Model 25

3 Simulation Results 29

3.1 Simulation Set-Up 29

3.2 Simulation Results and Discussion 31

4 Future Work 38

5 Summary 39
## II Estimation

### 6 Literature Review

- 6.1 Single-Camera Motion Estimation Techniques .......................... 43
- 6.2 Multicamera Motion Estimation Techniques .......................... 45

### 7 Kinematics Estimation Algorithm

- 7.1 Batch Maximum Likelihood Estimator Derivation .................... 49
- 7.2 Derivation of the Measurement Model and Error Jacobians ......... 52

### 8 Results and Discussion

- 8.1 Simulated Datasets ................................................. 60
- 8.2 Estimator Simulation Set-Up ..................................... 60
- 8.3 Estimator Results and Discussion ................................ 61

### 9 Future Work

### 10 Summary

### Conclusion

### References

### A Complete Equations of Motion

### B Complete Simulation Results

- B.1 \(A = 1 \text{ mN, } f = 10 \text{ Hz} \) ................................... 84
- B.2 \(A = 1 \text{ mN, } f = 20 \text{ Hz} \) ................................... 86
- B.3 \(A = 1 \text{ mN, } f = 30 \text{ Hz} \) ................................... 88
- B.4 \(A = 1 \text{ mN, } f = 40 \text{ Hz} \) ................................... 90
- B.5 \(A = 1 \text{ mN, } f = 50 \text{ Hz} \) ................................... 92
- B.6 \(A = 1 \text{ mN, } f = 60 \text{ Hz} \) ................................... 94
- B.7 \(A = 1 \text{ mN, } f = 70 \text{ Hz} \) ................................... 96
- B.8 \(A = 10 \text{ mN, } f = 10 \text{ Hz} \) ................................ 98
- B.9 \(A = 10 \text{ mN, } f = 20 \text{ Hz} \) ................................ 100
- B.10 \(A = 10 \text{ mN, } f = 30 \text{ Hz} \) ............................... 102
- B.11 \(A = 10 \text{ mN, } f = 40 \text{ Hz} \) ................................ 104
- B.12 \(A = 10 \text{ mN, } f = 50 \text{ Hz} \) ................................ 106
- B.13 \(A = 10 \text{ mN, } f = 60 \text{ Hz} \) ................................ 108
- B.14 \(A = 10 \text{ mN, } f = 70 \text{ Hz} \) ................................ 110
C Derivation of the Jacobian of $C^Tv$ 113
D Derivation of Equation (C.5) 116
E Derivation of Equation (C.7) 117
## List of Figures

1. The prototype of the Space Robotics Group’s dragonfly MAV ........................................... 2
2. A male *Sympetrum sanguineum* ......................................................................................... 2

1.1 A rendering of the MFI prototype ...................................................................................... 7
1.2 Some initial OVMI prototypes .......................................................................................... 7
1.3 A prototype of Harvard’s Microfly ..................................................................................... 8
1.4 A prototype of the Carnegie-Mellon MAV .......................................................................... 9
1.5 A comparison of Sane and Dickinson’s quasisteady aerodynamic model and empirical results ....................................................................................................................... 13

2.1 Components of the ATW system ....................................................................................... 15
2.2 Schematic of the prototype’s DoFs .................................................................................... 16
2.3 Schematic of the prototype’s geometric parameters .......................................................... 17
2.4 Aerodynamic forces and their directions .......................................................................... 26
2.5 Curve-fit for $C_{L,trans}$ .................................................................................................. 27
2.6 Curve-fit for $C_{D,trans}$ .................................................................................................. 28

3.1 Left wing kinematics with $A = 1$ mN and $f = 40$ Hz ...................................................... 32
3.2 Aerodynamic forces with $A = 1$ mN and $f = 40$ Hz ......................................................... 32
3.3 Left wing kinematics with $A = 10$ mN and $f = 40$ Hz ....................................................... 33
3.4 Aerodynamic forces with $A = 10$ mN and $f = 40$ Hz ....................................................... 33
3.5 Maximum wing angles versus actuator drive frequency..................................................... 34
3.6 Generated vertical forces with $A = 1$ mN and varying frequency ..................................... 35
3.7 Generated vertical forces with $A = 10$ mN and varying frequency .................................... 35
3.8 Left wing kinematics with $A = 10$ mN and $f = 30$ Hz ....................................................... 36
3.9 Total energy - work with $A = 1$ mN and $f = 40$ Hz ......................................................... 37

7.1 Schematic of an idealized pinhole camera .......................................................................... 53
7.2 Reference wings for a forewing only prototype .............................................................. 55
7.3 Schematic of the experimental set-up used for estimation ............................................... 56

8.1 Estimated error in the wing kinematics with no measurement noise .................................. 62
8.2 Estimated error in the camera-wing translation with no measurement noise . 62
8.3 Estimated error in the wing kinematics with $\sigma = 1$ pixel measurement noise 63
8.4 Estimated error in the camera-wing translation with $\sigma = 1$ pixel measurement noise . 63
8.5 Estimated error in the wing kinematics with $\sigma = 0.5$ pixel measurement noise 64
8.6 Estimated error in the camera-wing translation with $\sigma = 0.5$ pixel measurement noise . 65
8.7 Estimated error in the wing kinematics with $\sigma = 0.1$ pixel measurement noise 65
8.8 Estimated error in the camera-wing translation with $\sigma = 0.1$ pixel measurement noise . 66
8.9 Estimated vs. true wing kinematics with $\sigma = 1$ pixel measurement noise . 66
8.10 Estimated vs. true wing kinematics with $\sigma = 0.1$ pixel measurement noise . 67

B.1 Left wing states . . . . . . . . . . . . . . . . . . . 84
B.2 Right wing states . . . . . . . . . . . . . . . . . . . 85
B.3 Total Energy . . . . . . . . . . . . . . . . . . . . . . . 85
B.4 Aerodynamic Forces . . . . . . . . . . . . . . . . . . 86
B.5 Left wing states . . . . . . . . . . . . . . . . . . . . 86
B.6 Right wing states . . . . . . . . . . . . . . . . . . . . 87
B.7 Total Energy . . . . . . . . . . . . . . . . . . . . . . . 87
B.8 Aerodynamic Forces . . . . . . . . . . . . . . . . . . 88
B.9 Left wing states . . . . . . . . . . . . . . . . . . . . 88
B.10 Right wing states . . . . . . . . . . . . . . . . . . . . 89
B.11 Total Energy . . . . . . . . . . . . . . . . . . . . . . . 89
B.12 Aerodynamic Forces . . . . . . . . . . . . . . . . . . 90
B.13 Left wing states . . . . . . . . . . . . . . . . . . . . 90
B.14 Right wing states . . . . . . . . . . . . . . . . . . . . 91
B.15 Total Energy . . . . . . . . . . . . . . . . . . . . . . . 91
B.16 Aerodynamic Forces . . . . . . . . . . . . . . . . . . 92
B.17 Left wing states . . . . . . . . . . . . . . . . . . . . 92
B.18 Right wing states . . . . . . . . . . . . . . . . . . . . 93
B.19 Total Energy . . . . . . . . . . . . . . . . . . . . . . . 93
B.20 Aerodynamic Forces . . . . . . . . . . . . . . . . . . 94
B.21 Left wing states . . . . . . . . . . . . . . . . . . . . 94
B.22 Right wing states . . . . . . . . . . . . . . . . . . . . 95
B.23 Total Energy . . . . . . . . . . . . . . . . . . . . . . . 95
B.24 Aerodynamic Forces . . . . . . . . . . . . . . . . . . 96
B.25 Left wing states . . . . . . . . . . . . . . . . . . . . 96
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.26 Right wing states</td>
<td>97</td>
</tr>
<tr>
<td>B.27 Total Energy</td>
<td>97</td>
</tr>
<tr>
<td>B.28 Aerodynamic Forces</td>
<td>98</td>
</tr>
<tr>
<td>B.29 Left wing states</td>
<td>98</td>
</tr>
<tr>
<td>B.30 Right wing states</td>
<td>99</td>
</tr>
<tr>
<td>B.31 Total Energy</td>
<td>99</td>
</tr>
<tr>
<td>B.32 Aerodynamic Forces</td>
<td>100</td>
</tr>
<tr>
<td>B.33 Left wing states</td>
<td>100</td>
</tr>
<tr>
<td>B.34 Right wing states</td>
<td>101</td>
</tr>
<tr>
<td>B.35 Total Energy</td>
<td>101</td>
</tr>
<tr>
<td>B.36 Aerodynamic Forces</td>
<td>102</td>
</tr>
<tr>
<td>B.37 Left wing states</td>
<td>102</td>
</tr>
<tr>
<td>B.38 Right wing states</td>
<td>103</td>
</tr>
<tr>
<td>B.39 Total Energy</td>
<td>103</td>
</tr>
<tr>
<td>B.40 Aerodynamic Forces</td>
<td>104</td>
</tr>
<tr>
<td>B.41 Left wing states</td>
<td>104</td>
</tr>
<tr>
<td>B.42 Right wing states</td>
<td>105</td>
</tr>
<tr>
<td>B.43 Total Energy</td>
<td>105</td>
</tr>
<tr>
<td>B.44 Aerodynamic Forces</td>
<td>106</td>
</tr>
<tr>
<td>B.45 Left wing states</td>
<td>106</td>
</tr>
<tr>
<td>B.46 Right wing states</td>
<td>107</td>
</tr>
<tr>
<td>B.47 Total Energy</td>
<td>107</td>
</tr>
<tr>
<td>B.48 Aerodynamic Forces</td>
<td>108</td>
</tr>
<tr>
<td>B.49 Left wing states</td>
<td>108</td>
</tr>
<tr>
<td>B.50 Right wing states</td>
<td>109</td>
</tr>
<tr>
<td>B.51 Total Energy</td>
<td>109</td>
</tr>
<tr>
<td>B.52 Aerodynamic Forces</td>
<td>110</td>
</tr>
<tr>
<td>B.53 Left wing states</td>
<td>110</td>
</tr>
<tr>
<td>B.54 Right wing states</td>
<td>111</td>
</tr>
<tr>
<td>B.55 Total Energy</td>
<td>111</td>
</tr>
<tr>
<td>B.56 Aerodynamic Forces</td>
<td>112</td>
</tr>
</tbody>
</table>
Acronyms

MAV  Micro Aerial Vehicle
ATW  actuator-transmission-wing
UAV  Unmanned Aerial Vehicle
UTIAS University of Toronto Institute for Aerospace Studies
DoF  degree of freedom
CFD  Computational Fluid Dynamics
MFI  Micromechanical Flying Insect
OVMI Objet Volant Mimant l’Insecte
GUF  Grand Unified Fly
ODE  ordinary differential equation
PoI  point of interest
RPE  Reprojected Pixel Error
HRMT Hull Reconstruction Motion Tracking
SLAM Simultaneous Localization and Mapping
MLE maximum likelihood estimator
SURF Speeded Up Robust Features
SIFT Scale-Invariant Feature Transform
Notation

General

\( \mathbf{a} \) : Bold lowercase variables are column vectors.

\( \mathbf{A} \) : Bold uppercase variables are matrices.

\( \dot{a} \) : Variables with a single dot above them represent the first time derivative of that variable: \( \frac{da}{dt} \).

\( \ddot{a} \) : Variables with two dots above them represent the second time derivative of that variable: \( \frac{d^2a}{dt^2} \).

Dynamical Modelling

\( \mathbf{v}_i \) : The velocity of the \( i \)'th component with respect to the inertial frame, expressed in the inertial frame.

\( \mathbf{\omega}_i \) : The angular velocity of the \( i \)'th component with respect to the inertial frame, expressed in the inertial frame.

\( \mathbf{J}_i \) : The 2nd moment of inertia of the \( i \)'th component.

\( J_{i,jk} \) : The \( j,k \)-element of the 2nd moment of inertia of the \( i \)'th component.

\( m_i \) : The mass of the \( i \)'th component.

\( k_i \) : The coefficient of stiffness of the spring attached to the \( i \)'th component.

\( T_i \) : The kinetic energy of the \( i \)'th component.

\( V_i \) : The potential energy of the \( i \)'th component.

\( \mathcal{L}_i \) : The Lagrangian for the \( i \)'th component.

Estimation

\( ^n\mathbb{R} \) : The space of the \( n \times 1 \) column of real numbers.

\( ^n\mathbb{R}^m \) : The space of the \( n \times m \) matrices of real numbers.

\( \mathcal{A} \) : Uppercase variables in the script font are transformations.

\( \mathbf{a}_b \) : The coordinates of the vector \( \mathbf{a} \), expressed in frame \( \mathbf{b} \).

\( \rho_{cb} \) : The translation from point \( \mathbf{a} \) to \( \mathbf{b} \), expressed in frame \( \mathbf{c} \).

\( \mathbf{C}_{ba} \) : The rotation matrix representing the rotation from frame \( \mathbf{a} \) to \( \mathbf{b} \).
Introduction

Recent developments in materials and microfabrication techniques have allowed the development of smaller and smaller aircraft for use in indoor or tightly constrained spaces. The design and manufacturing of Micro Aerial Vehicles (MAVs), a class of miniature Unmanned Aerial Vehicles (UAVs), currently represents a rapidly developing research area. While researchers initially focused on fixed-wing or rotor configurations, these were found to be inefficient because of the low Reynolds number aerodynamic flows created by the small scale of the vehicles. As a result, research moved towards flapping-wing configurations (inspired by insects) which were more efficient. These insect-inspired MAVs also offer hovering abilities as well as extreme manoeuvrability.

The Space Robotics Group at the University of Toronto Institute for Aerospace Studies (UTIAS) has embarked on the development of a dragonfly-inspired MAV; an engineering prototype of this MAV is seen in Figure 1. Dragonflies were chosen as the reference insect because of their relatively simple flight characteristics in hovering and manoeuvring, which are easier to recreate mechanically. Specifically, unlike most other insects, dragonfly flapping is constrained to a plane (even during manoeuvres), called the stroke plane. Dragonflies also do not perform “clap-and-fling” (a mechanism that will be discussed later in this work), which also simplifies the mechanics of flapping. Based on the dragonfly *Sympetrum sanguineum*, seen in Figure 2, the current MAV design attempts to replicate the physical characteristics of this species in order to achieve similar flight properties. This species of dragonfly has been intensively studied by biologists and its flight mechanics and performance are well characterized [1], [2], [3].

At this point in the project, the current prototypes of the dragonfly MAV are limited to only actuating the flapping motion of the wings. Research in the group has mostly focused on designing the actuator-transmission-wing (ATW) system in order to achieve the desired flapping (or stroke) angles based on the kinematics of the system. However, to predict the actual motion of the prototype wings, the internal dynamics of the actuator and transmission, as well as the generated aerodynamic forces all need to be considered. This requires a dynamical rather than a kinematic model. To this end, this thesis presents two contributions to the Space Robotics Group’s dragonfly MAV project.

The first contribution is a derivation of a dynamical model for the ATW subsystem of
Figure 1: A recent engineering prototype of the Space Robotics Group’s dragonfly MAV. This prototype includes both the forewing and hindwing pairs.

Figure 2: A male specimen of *Sympetrum sanguineum*. Photo by Charles Sharp from https://www.sharpphotography.co.uk/.
the dragonfly MAV. This model is intended to be used for future iterations of prototype
design and is the first step to deriving a full plant model that will eventually be used for
designing a control system. This part of the thesis will begin by covering the existing state
of modelling insect-inspired MAVs, both in terms of the dynamics of the MAVs and insect
aerodynamics. None of these works account for the internal dynamics of their systems and
instead assume either that the actuator force and aerodynamics apply to the wing directly
or assume specified wing kinematics. However, since the actuator composes the majority
of the mass of the system, its motion and dynamics will affect the overall dynamics of the
system. These overall dynamics are necessary to understand for any further prototype design
or control design. As a result, a complete model of the ATW system of the dragonfly MAV
is derived using Lagrangian techniques. Simulation results are then presented and discussed
before continuing with a description of future work.

The second contribution of this thesis is an estimation algorithm that can be used to
verify the dynamical model presented in the previous part and can also be used to determine
the dragonfly MAV prototype performance. Specifically, the algorithm estimates the wing
kinematics of the MAV. This second part begins by covering the techniques to estimate
wing motion used by biologists and other labs building insect-inspired MAVs. Most of these
methods are either purely geometric and deterministic or require expensive laboratory setups
like having multiple high-speed cameras. Because of equipment constraints and a desire for
rigour, this thesis proposes a completely different, probabilistic method of estimation inspired
by the canonical Gaussian-Newton Batch Estimator (equivalent to a fixed-interval smoother).
A derivation of this estimation algorithm based on first principles is presented. Then the
estimation algorithm is tested on simulated data and is shown to converge to the known true
trajectory. A discussion of the algorithm performance is presented before concluding with
some suggestions for future work.
Part I

Dynamical Model
There are a number of reasons why a dynamical model of the dragonfly MAV would benefit the overall project. First, it would expedite the design process. Currently refinements to the design are evaluated by building a prototype and testing it. A dynamical model would allow parameter variations to be tested in simulation before committing to building a new prototype. Thus, less effective variations need not be built for testing, saving both time and resources. Furthermore, a model can provide insight into the system; for example, the optimal driving frequency can be found by finding the natural frequency of the system given by the eigenvalues of the model. Finally, this model can be used for a control system plant that would be implemented once flight is achieved. To these ends, it is desired to have the most complete model possible, which can then be simplified as necessary.

This part of the thesis will begin with an overview of the other existing insect-inspired MAV projects before describing the modelling techniques used for these projects, both for the dynamics of the overall system and the aerodynamics of insect flight. Then the dynamical model for the dragonfly MAV will be derived using the Lagrangian method before presenting simulation results, analysis and future work.
Chapter 1

Literature Review

1.1 Current and Past Insect-Inspired MAV projects

There are a number of groups working on simulations and prototypes for insect-inspired flapping wing MAVs. The following are the more notable projects.

University of California Berkeley Biomimetic Systems Lab: Micromechanical Flying Insect (MFI)

The MFI project (illustrated in Figure 1.1) was one of the first attempts to build an insect-inspired MAV, this one specifically based on a blowfly of the genus *Calliphora*. While the project envisioned a complete system including an integrated power source and sensors, such a prototype was never built. Instead components of the MAV, including an actuator and a transmission, were built and tested. As a result, this system never flew. Most of the work focussed strongly on developing models and control architectures for the MAV. A hierarchical control scheme was developed and tested in simulation. These results Deng *et al.* proposed [4], [5] are still used by other researchers.

French National Research Agency: Objet Volant Mimant l’Insecte (OVMI) Project

The OVMI project is a multi-institute initiative. While work is being done to develop prototypes, which are seen in Figure 1.2, currently all published results have been obtained using simulations [6], [7]. The prototypes are assumed to have two actuated degrees of freedom (DoFs). Much of the work has focussed on developing control algorithms for the MAV. The dynamical model of the system is very simplistic and applies the aerodynamic forces calculated from the wing kinematics directly to the body [6].
Figure 1.1: A rendering of the complete MFI prototype that was planned and never actually built [4].

Figure 1.2: An initial prototype from the OVMI project [6].
Chapter 1. Literature Review

Figure 1.3: An early prototype of the Harvard Microfly [9]. In later years, sensors and other actuators were added as well.

Harvard Microrobotics Laboratory: Microfly

To date, this is the only to-scale prototype that has achieved flight [8]. Based on the housefly *Drosophila melanogaster*, the Microfly (seen in Figure 1.3) has one actuated DoF, the stroke angle, and one passive DoF, the angle of attack. In 2007, this prototype was able to achieve vertical flight guided by two wires while tethered to an off-board power supply.

Further research is focussed on developing actuation methods and sensors for control. There are two main development streams for actuation: one for active control and one for passive control. For active control, Harvard has implemented two mechanisms that allow for control over MAV pitch and roll. To create a pitching moment, they apply a bias voltage that shifts the average centre of lift over a wing-beat either ahead or behind the centre of mass of the MAV. To create a rolling moment, they create asymmetrical stroke amplitudes by shifting components of the transmission [10], [11]. This results in asymmetrical aerodynamic forces on the two wings, creating a rolling torque on the body. Other authors have also used the Microfly model and suggested different modifications to allow for active control [12], [13].

For passive control, Sreetharan and Wood have designed a torque balancing transmission, similar in concept to the automobile differential [14], [15]. They have experimentally verified that aerodynamic torques from each wing become close to equal when using this transmission.

For sensors, research has primarily focussed on optical sensors [16], [17]. Initial results have shown that these sensors are capable of stabilizing free flight [17].

University of Delaware/Purdue University: Dragonfly Robot

Like the Space Robotics Group at UTIAS, Deng at the University of Delaware, and later at Purdue University, is developing a dragonfly-inspired MAV. However, they have approached
the problem by building larger-scale prototypes and miniaturizing them. The third generation of prototype had 4 wings and a mass of 4 g. The current generation (the fifth generation) has been reduced to 2 g, but appears to have only two wings [18]. This is the only other known dragonfly-inspired MAV. Thus far, no empirical results have been published based on this generation of the prototype. However, some initial kinematic and force results were published for the second generation prototype [19].

Carnegie-Mellon University NanoRobotics Laboratory

The NanoRobotics lab at Carnegie-Mellon University is currently working on a two-winged MAV (seen in Figure 1.4) similar to the Harvard Microfly. Like the first Microfly prototype, it has one actuated DoF (the stroke angle), with a passive angle of attack. However, the Carnegie-Mellon MAV has two actuators for independent control of each wing. While the current prototype’s mass is not given, it was empirically measured to have a lift-to-weight ratio of 0.2. Using simulations, researchers have predicted that a half-scale prototype will have a lift-to-weight ratio of 2, allowing liftoff [20].

The main difference between these projects and the dragonfly MAV being developed at UTIAS is that most of these projects are based on flies rather than dragonflies. As a result of the extremely different flight mechanics, the work done to model these systems needs to be repeated for the dragonfly MAV, taking the differences into account.

1.2 Dynamical Models for Flapping-Wing MAVs

For the most part, researchers tend to use simplified dynamical models. The first assumption is that since the mass of the wings are much smaller than that of the body, the inertial
effects of wings and the interbody forces between the wings and internal components of the MAV are ignored. In essence this means that the MAV is modelled by having aerodynamic forces and torques applied directly to a rigid body. However, an important implication of this method is that these researchers must assume defined wing kinematics. In reality, these wing kinematics are a result of the interaction between the control forces and torques applied to the wing by the actuator through the transmission and the aerodynamic forces generated by the wing. This nuance is not fully captured by most dynamical models.

Only Harvard has developed dynamical models for their systems based on the Lagrangian formulation; these were used in developing their control schemes. However, they claim that the moments of inertia of the transmission components is negligible and apply the actuator and aerodynamic forces and torques directly to the wing [15]. They do no account for the interbody forces present in the transmission. More important, they do not account for the motion of the actuator for their models because the prototype body is fixed in their experiments [15]. However, the actuator motion can affect the overall system dynamics in free flight, limiting the usefulness of this model. Harvard verifies their simulated results by measuring the wing kinematics of their prototypes using high-speed stereo cameras.

Perhaps the most comprehensive model in literature, although indirectly related to flapping-wing MAVs, is the Grand Unified Fly (GUF) simulation written by Dickson et al. [21]. The simulation consists of 5 integrated modules: the sensory system model, the rigid body dynamical model, the aerodynamic model, the control model and the environment model. Of note, the rigid body dynamical model assumes that the wings and the body of the fly all act as rigid bodies connected by three DoF ball joints. However, because this is a model for a biological fly and not a flapping-wing MAV, this model also does not account for dynamics of internal components like the actuator. The dynamics of the system are given using the Open Dynamics Engine, an open source physics engine that is often used for video games [21].

1.3 Insect Flight and Unsteady Aerodynamic Models

A common thread in all these dynamical simulations is the model used for the aerodynamics. Insect flight has long interested scientists and engineers, but it is only recently that technology has advanced enough to decipher their secrets. The use of high-speed cameras and unsteady Computational Fluid Dynamics (CFD) simulations, among other technological advancements, have allowed further insight into the highly unsteady nature of insect flight. In order to develop a model of the dragonfly MAV, a good understanding of these effects is needed. Furthermore, since an analytical model is desired for further research, it is necessary to determine which unsteady effects have analytical representations and whether such representations accurately model them. Typically quasisteady approximations are used to model these effects. Some of
the unsteady mechanisms that have been proposed include the clap-and-fling mechanism, the Wagner effect, delayed stall, the Kramer effect, added-mass effect and wing-wake interactions.

However, it should be noted that dragonflies have different flight mechanisms from most other insects since dragonflies are considered to be older and thus more primitive [2]. They have four wings, as opposed to the two wings found on most insects. It has been observed that in hovering the hindwings mimic the trajectories of the forewings, but with a 90° phase lead [22]. In hovering, dragonflies also exhibit significantly different flight mechanics. Most modern, two-winged insects perform “normal” hovering, where the wings stroke (flap) back and forth in a horizontal plane [22]. Rather than a horizontal stroke plane, dragonflies have a stroke plane that is inclined to the horizontal [2], [22]. For the dragonfly *Sympetrum sanguineum*, the stroke plane is angled at 50° [2].

**Clap-and-Fling Mechanism**

Most modern insects fly using the “clap-and-fling” technique, which was first described by Weis-Fogh in 1973. At the peak of the upstroke, the wings clap together and then fling apart. This creates circulatory flow in between the wings, which enhances the lift [2]. This is one of the unsteady effects that could not be accounted for by using a quasisteady aerodynamic model. However, dragonflies do not display this mechanism and so it does not need to be modelled for the dragonfly MAV.

**Wagner Effect**

When a wing is flapping, it is constantly undergoing changes in acceleration — especially as it pauses and changes directions at the top and bottom of each stroke. Each time the wing restarts its motion, vortices are shed at the trailing edge rotating in the direction opposite to the circulation around the wing [23]. This counter rotating vortex decreases the lift until it moves farther downstream. This is called the Wagner effect, named after the aerodynamicist Herbert A. Wagner who first formulated the theory. Recent studies have shown that this effect does not play a large role at the Reynolds numbers at which insects fly. As a result, most recent aerodynamic models of insect flight do not include this effect [24].

**Delayed Stall**

Delayed stall is the one of the earliest unsteady aerodynamic mechanisms of insect flight to be understood. As a wing’s angle of attack increases, the flow at the leading edge separates from the wing, but reattaches to the trailing edge while the angle is below a certain threshold. The vortex that forms on the surface of the wing is called a “bound vortex” as it remains attached to the wing. This circulation on the wing increases the lift generated [24]. Eventually, the
vortex grows to the point where the flow separates at the trailing edge and the vortex is shed, resulting in a loss of lift and an increase in drag. However, in insects the bound vortex remains in place for longer as there is flow axially along the wing from the root to the tip. This additional flow limits the size of the vortex, preventing it from being shed, thereby delaying stall [24]. The delayed stall can be accounted for in the choice of the lift and drag coefficients used in a quasisteady model.

**Kramer Effect**

The Kramer effect is also called rotational lift. As the wing rotates and flaps, the flow does not immediately reestablish the Kutta condition at the trailing edge. Recall, that the Kutta condition states that the circulation around an airfoil at a given angle of attack is such that the flow smoothly leaves the trailing edge [25]. When the Kutta condition is not met, there exists a shear force at the trailing edge. As a result, additional circulation is created in the flow to counteract the shear, which then reestablishes the Kutta condition [24]. This additional circulation increases the lift generated by the wing [24]. This effect was first described by the aerodynamicist Max Kramer in 1932 when he was examining the effects of rapid increases in the angle of attack caused by gusts [26]. Most current aerodynamic models include a quasisteady approximation of the Kramer effect [24].

**Added-Mass Effect**

This effect is also known as the apparent-mass effect. It is a noncirculatory force, in that it is not caused by circulation in the flow around the wing. Rather, the added-mass effect is caused by the reaction forces imparted by the fluid to a wing that is accelerating through the fluid [23], [24]. This term can also be modelled using a quasisteady approximation [24], but to date most models do not include it.

**Wing-Wake Interactions**

Wing-wake interactions are one of the most important unsteady effects in insect flight. Studies of insect flight kinematics combined with CFD modelling of insect wings have shown that the wing passes through the vortex wake shed by a prior stroke. This allows the wing to “recapture” some of energy lost in a previous stroke [24]. Unfortunately, despite the importance of this effect, no quasisteady approximation exists, and as a result, it is not used in models.

Currently, many of these unsteady effects are modelled using computational methods rather than using the analytic formulations desired for the dynamical model of the dragonfly MAV.
Figure 1.5: A comparison between a quasisteady model of the aerodynamics of *Drosophila melanogaster* and empirical results generated by a dynamically scaled wing model (from Sane and Dickinson [27]). The quasisteady results are shown in black and the measured values are shown in red.

As a result, most researchers use versions of a quasisteady model proposed by Sane and Dickinson [27] for the fruit fly *Drosophila melanogaster*. This quasisteady model approximates some of the unsteady aerodynamic effects, namely the additional circulation that results from wing rotation at the ends of half-strokes, by combining a traditional translational aerodynamic model and a quasisteady model for the Kramer effect. It is used over other quasisteady models, such as the Pitt-Peters model [28], because the aerodynamic forces are functions of the same state variables as the dynamical equations of motion. Sane and Dickinson’s model is summarized in Section 2.2. They verified their model by comparing the theoretical results to empirical results from a dynamically scaled wing model. While it has been shown that a quasisteady aerodynamic model is still insufficient to describe the flight mechanics of dragonflies [2], [22], it can be seen from Figure 1.5, the quasisteady approximation provides a fairly good model. The peaks at the beginning of the strokes probably result from unsteady effects that are not modelled, such as wing-wake interactions. Similar results have been obtained by a number of other authors including Yamamoto and Isogai [29] and Deng *et al.* [4].
Chapter 2

Model Derivation

2.1 Actuator-Transmission-Wing Model

Unlike the models presented in literature, the model presented in this work for the dragonfly MAV includes the dynamics of the internal components. The mass of the actuator is a significant percentage of the total mass of the prototype, and its motion can affect the overall system dynamics and stability. Furthermore it is easier to simplify a more complete model by removing elements than it is to expand a simpler model.

The primary purpose of the model developed in this thesis is to be used in conjunction with the prototype design process to help determine the physical parameters of the system to achieve the desired performance. At this time, the forewing pair kinematics and the resulting aerodynamic forces are of primary importance. This reflects the current experiments performed in the lab, where prototypes of the forewing pair are attached to a force sensor for measurements. For future prototypes, the hindwing pair and the motion of the body will also need to be considered.

The dragonfly MAV model is derived by considering the motion of the individual internal components of the prototype. These components consist of the frame $B_0$, which is fixed to the force sensor, the actuator $B_1$, the actuator linkage $B_2$, the transmission base $B_3$, the left and right transmission flanges $B_4$ and $B_6$, and the left and right wings $B_5$ and $B_7$. These structures are illustrated in the SolidWorks model in Figure 2.1. These components are all connected to each other with joints, which are modelled as springs. Note that the dynamical model that will be presented is based on the prototype labelled 2P15, which does not include a stroke plane. Adding a stroke plane later is a simple exercise compared to deriving the dynamics of the system.

With the exception of the prototype frame, each component is restricted to at most two DoFs by the geometry of the system. This geometry can actually be be simplified slightly for easier modelling as shown in Figures 2.2 and 2.3. These figures also identify the DoFs of each component. The DoFs for some of the bodies are also limited by some of the geometric
Figure 2.1: Illustration of the elements of the drive system of the dragonfly MAV prototype, consisting of the actuator and the transmission. SolidWorks model courtesy Peter Szabo [30].
Figure 2.2: Schematic showing the DoFs of the dragonfly MAV’s ATW system.
Figure 2.3: Schematic showing the geometric parameters of the dragonfly MAV's ATW system.
parameters of the transmission system. These physical parameters are an important part of the prototype design and are also identified in Figures 2.2 and 2.3.

From this simplified geometry, the equations of motion can be derived using the Lagrangian technique. The Lagrangian method was chosen over a Newton-Euler method because it was simpler to formulate the equations using just the DoFs of each component. Also the Lagrangian method makes the geometric relations between the physical parameters of the actuator-transmission system and the DoFs of the components explicit, while simultaneously reducing the system of equations to the desired output, namely the wing kinematics.

To set up the Lagrangian of the entire system, the kinetic and potential energies of each of the individual components are first derived, as well as the geometric constraints linking the physical parameters of the system and the DoF. These constraints are then linearized. Finally the constraints are substituted into the energy equations to reduce the system to the desired states. Note that in this derivation, all quantities are expressed in the inertial frame unless otherwise specified.

\[ \mathcal{B}_0: \text{Body} \]

In the current experiments, since the dragonfly MAV prototype is fixed to a force sensor, the motion of the body is negligible and is there assumed to be fixed in inertial space.

\[ \mathcal{B}_1: \text{Actuator} \]

The actuator has one DoF with respect to the body, \( \theta_1 \), which is a rotation around the y-axis. The angular velocity of the actuator, \( \omega_1 \), is

\[
\omega_1 = \omega_{1,\text{int}} = \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ 0 \end{bmatrix}
\]  

(2.1)

and the velocity of the actuator, \( v_1 \) is

\[
v_1 = \omega_1 \times \rho_{1,\text{cm}} = \begin{bmatrix} \frac{1}{2} l_1 \sin \theta_1 \dot{\theta}_1 \\ 0 \\ \frac{1}{2} l_1 \cos \theta_1 \dot{\theta}_1 \end{bmatrix},
\]  

(2.2)

where \( \rho_{1,\text{cm}} \) is the position of the centre of mass of the actuator, given as

\[
\rho_{1,\text{cm}} = \begin{bmatrix} -\frac{1}{2} l_1 \cos \theta_1 \\ 0 \\ \frac{1}{2} l_1 \sin \theta_1 \end{bmatrix}.
\]  

(2.3)
The kinetic energy is then
\[ T_1 = \frac{1}{8} m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} J_{1,22} \dot{\theta}_1^2 \]  
and the potential energy is
\[ V_1 = \frac{1}{2} m_1 g l_1 \sin \theta_1 + \frac{1}{2} k_1 \theta_1^2. \]

The Lagrangian for the actuator is
\[ \mathcal{L}_1 = T_1 - V_1. \]

\[ B_2: \text{Actuator-Transmission Linkage} \]

This linkage has one DoF with respect to the body, \( \theta_2 \), which is a rotation around the \( y \)-axis. The angular velocity of the linkage in the body-fixed frame, \( \omega_{2,\text{int}} \) is then
\[ \omega_{2,\text{int}} = \begin{bmatrix} 0 \\ \dot{\theta}_2 \\ 0 \end{bmatrix} \]  
and the total angular velocity of the linkage, \( \omega_2 \), is
\[ \omega_2 = \omega_{1,\text{int}} + \omega_{2,\text{int}} = \begin{bmatrix} 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \\ 0 \end{bmatrix}. \]

The velocity of the linkage, \( v_2 \) is given by
\[ v_2 = \omega_{1,\text{int}}^x (\rho_{1,2} + \rho_{2,\text{cm}}) + \omega_{2,\text{int}}^x \rho_{2,\text{cm}} \]
\[ = \begin{bmatrix} \dot{\theta}_1 (l_1 \sin \theta_1 + \frac{1}{2} l_2 \cos \theta_2) + \dot{\theta}_2 (\frac{1}{2} l_2 \cos \theta_2) \\ 0 \\ \dot{\theta}_1 (l_1 \cos \theta_1 + \frac{1}{2} l_2 \sin \theta_2) + \dot{\theta}_2 (\frac{1}{2} l_2 \sin \theta_2) \end{bmatrix}, \]

where \( \rho_{1,2} \) is the position of the origin of the linkage frame with respect to the origin of the actuator frame
\[ \rho_{1,2} = \begin{bmatrix} -l_1 \cos \theta_1 \\ 0 \\ l_1 \sin \theta_1 \end{bmatrix}. \]
and where $\rho_{2,cm}$ is the position of the centre of mass of the linkage relative to the origin of the linkage frame

$$
\rho_{2,cm} = \begin{bmatrix}
-\frac{1}{2} l_2 \sin \theta_2 \\
0 \\
\frac{1}{2} l_2 \cos \theta_2
\end{bmatrix}.
$$

The kinetic energy of the linkage is then

$$
T_2 = \frac{1}{2} J_{2,22} \left( \dot{\theta}_1 + \dot{\theta}_2 \right)^2 \\
+ \frac{1}{2} m_2 \left[ \left( l_1^2 + \frac{l_2^2}{4} + l_1 l_2 \sin (\theta_1 + \theta_2) \right) \dot{\theta}_1^2 + l_1 l_2 \sin (\theta_1 + \theta_2) \dot{\theta}_1 \dot{\theta}_2 + \frac{l_2^2}{4} \dot{\theta}_2^2 \right].
$$

The potential energy is

$$
V_2 = m_2 g \left( l_1 \sin \theta_1 + \frac{1}{2} l_2 \cos \theta_2 \right) + \frac{1}{2} k_2 \theta_2^2.
$$

Therefore the Lagrangian for the linkage is

$$
\mathcal{L}_2 = T_2 - V_2
$$

with the following holonomic constraint equation given by the geometry of the system

$$
l_1 = l_1 \cos \theta_1 + l_2 \sin \theta_2.
$$

$B_3$: Transmission base

The transmission base has one DoF with respect to body: translation along z-axis. Thus the velocity is given by the z-component of

$$
\omega_{1, \text{int}}^x \rho_{1,2} + \omega_{2, \text{int}}^x \rho_{2,3} = 
\begin{bmatrix}
\dot{\theta}_1 (l_1 \sin \theta_1 + l_2 \cos \theta_2) + \dot{\theta}_2 l_2 \cos \theta_2 \\
0 \\
\dot{\theta}_1 (l_1 \cos \theta_1 + l_2 \sin \theta_2) + \dot{\theta}_2 l_2 \sin \theta_2
\end{bmatrix},
$$

where $\rho_{2,3}$ is the position of the origin of the transmission base frame with respect to the origin of the actuator frame

$$
\rho_{2,3} = \begin{bmatrix}
-l_2 \sin \theta_2 \\
0 \\
l_2 \cos \theta_2
\end{bmatrix}.
$$
Therefore the velocity of the base, \( v_3 \), is

\[
v_3 = \begin{bmatrix} 0 \\ 0 \\ (l_1 \cos \theta_1 + l_2 \sin \theta_2) \dot{\theta}_1 + l_2 \sin \theta_2 \dot{\theta}_2 \end{bmatrix}.
\] (2.19)

The kinetic energy is then

\[
T_3 = \frac{1}{2} m_3 \left[ (l_1 \cos \theta_1 + l_2 \sin \theta_2) \dot{\theta}_1 + l_2 \sin \theta_2 \dot{\theta}_2 \right]^2
\] (2.20)

and the potential energy is

\[
V_3 = m_3 g \left( l_1 \sin \theta_1 + l_2 \cos \theta_2 \right) + \frac{1}{2} k_3 \left( \frac{\pi}{2} - \theta_2 \right)^2.
\] (2.21)

The Lagrangian of the transmission base is then

\[
\mathcal{L}_3 = T_3 - V_3
\] (2.22)

**\( \mathcal{B}_4, \mathcal{B}_6: Transmission Flanges **

The transmission flanges have two DoFs with respect to body: translation along z-axis and rotation around x-axis. For the left flange, this rotation is denoted by \( \theta_4 \). The translation of the base of the flange is the same as the translation of the transmission base. The angular velocity of the left flange in the body fixed frame is given by

\[
\omega_{4,int} = \begin{bmatrix} \dot{\theta}_4 \\ 0 \\ 0 \end{bmatrix}
\] (2.23)

and the velocity \( v_4 \) is given by

\[
v_4 = v_3 + \omega_{4,int}^x \rho_{4,cm}
\] (2.24)

\[
= \begin{bmatrix} 0 \\ -\frac{1}{2} \dot{\theta}_4 \cos \theta_4 \\ (l_1 \cos \theta_1 + l_2 \sin \theta_2) \dot{\theta}_1 + l_2 \sin \theta_2 \dot{\theta}_2 - \frac{1}{2} l_4 \sin \theta_4 \dot{\theta}_4 \end{bmatrix},
\] (2.25)

where \( \rho_{4,cm} \) is the position of the centre of mass of the flange relative to the origin of the flange frame

\[
\rho_{4,cm} = \begin{bmatrix} 0 \\ -\frac{1}{2} l_4 \sin \theta_4 \\ \frac{1}{2} l_4 \cos \theta_4 \end{bmatrix}.
\] (2.26)
The kinetic energy is then

\[ T_4 = \frac{1}{2} m_4 \left[ \frac{1}{4} l_4^2 \cos^2 \theta_4 \dot{\theta}_4 + \left( (l_1 \cos \theta_1 + l_2 \sin \theta_2) \dot{\theta}_1 \right. \right. \]
\[ + l_2 \sin \theta_2 \dot{\theta}_2 - \frac{1}{2} l_4 \sin \theta_4 \dot{\theta}_4 \left. \right] + \frac{1}{2} J_{4,11} \dot{\theta}_4^2 \]

(2.27)

and the potential energy is

\[ V_4 = m_4 g \left( l_1 \cos \theta_1 + l_2 \sin \theta_2 + \frac{1}{2} l_4 \cos \theta_4 \right) + \frac{1}{2} k_4 \theta_4^2. \]

(2.28)

The Lagrangian of the left flange is then

\[ \mathcal{L}_4 = T_4 - V_4 \]

(2.29)

with the geometric, holonomic constraint

\[ l_2 + l_4 = l_1 \sin \theta_1 + l_2 \cos \theta_2 + l_4 \cos \theta_4 - l_5 \sin \phi_5. \]

(2.30)

The right transmission flange has the same DoFs as the left flange. The rotation of the right flange is denoted \( \theta_6 \). The equations for the kinetic energy, potential energy and Lagrangian are similar to those of \( B_4 \).

\( B_5, B_7: \) Wings

The wings have two DoFs with respect to body: rotation around x-axis, or stroke angle, and rotation around y-axis, or geometric angle of attack. For the left wing, the stroke angle is denoted \( \phi_5 \), and the geometric angle of attack by \( \alpha_5 \). The angular velocity of the wing is given as

\[ \omega_5 = \begin{bmatrix} \dot{\phi}_5 \cos \alpha_5 \\ \dot{\alpha}_5 \\ \dot{\phi}_5 \sin \alpha_5 \end{bmatrix}. \]

(2.31)

The kinetic energy is then

\[ T_5 = \frac{1}{2} J_{5,11} \cos \alpha_5^2 \dot{\phi}_5^2 + J_{5,33} \sin \alpha_5^2 \dot{\alpha}_5^2 + J_{5,12} \cos \alpha_5 \phi_5 \dot{\phi}_5 \dot{\alpha}_5 + \frac{1}{2} J_{5,22} \dot{\alpha}_5^2 \]

(2.32)

and the potential energy is

\[ V_5 = m_5 g \left( l_4 - \cos \phi_5 \sin \alpha_5 x_{5,cm} - \sin \phi_5 y_{5,cm} \right) + \frac{1}{2} k_\phi \phi_5^2 + \frac{1}{2} k_\alpha \alpha_5^2. \]

(2.33)
The Lagrangian of the left wing is then
\[ \mathcal{L}_5 = T_5 - V_5 \]  
(2.34)
with the geometric, holonomic constraint
\[ l_5 = l_4 \sin \theta_4 + l_5 \cos \phi_5. \]  
(2.35)

The right wing has the same DoFs of freedom as the left wing. The stroke angle of the left wing is denoted by \( \phi_7 \), and the geometric angle of attack is denoted by \( \alpha_7 \). The equations for the kinetic energy, potential energy and Lagrangian are similar to those of \( B_5 \). The geometric constraint for the right side of the transmission is
\[ l_7 = l_6 \sin \theta_6 + l_7 \cos \phi_7. \]  
(2.36)
Also from the symmetry of the transmission, there is another holonomic constraint that
\[ \phi_7 = -\phi_5. \]  
(2.37)

**Linearization**

Before calculating the final equations of motion, it should be noted that the constraint equations can be simplified since some of the internal angles are small. Specifically, \( \theta_1, \theta_2, \theta_4 \) and \( \theta_6 \) are all small. Therefore Equations (2.16), (2.35), (2.36) and (2.30) can be rewritten as follows:
\[ l_1 = l_1 \cos \theta_1 + l_2 \sin \theta_2 \quad \Rightarrow \quad \theta_2 = \frac{1}{2} \frac{l_1}{l_2} \theta_1^2 \]  
(2.38)
\[ l_5 = l_4 \sin \theta_4 + l_5 \cos \phi_5 \quad \Rightarrow \quad \theta_4 = \frac{l_5}{l_4} \left(1 - \cos \phi_5 \right) \]  
(2.39)
\[ l_7 = l_6 \sin \theta_6 + l_7 \cos \phi_7 \quad \Rightarrow \quad \theta_6 = \frac{l_7}{l_6} \left(1 - \cos \phi_7 \right) \]  
(2.40)
\[ l_2 + l_4 = l_1 \sin \theta_1 + l_2 \cos \theta_2 + l_4 \cos \theta_4 - l_5 \sin \phi_5 \quad \Rightarrow \quad \theta_1 = \frac{1}{2} \frac{l_2^2}{l_1 l_4} \left(1 - \cos \phi_5 \right)^2 - \frac{l_5}{l_1} \sin \phi_5 \]  
(2.41)

The energy equations can be linearized similarly:
\[ T_{1,lin} = \frac{1}{2} J_{1,22} \dot{\theta}_1^2 + \frac{1}{8} m_1 l_1^2 \dot{\theta}_1^2 \]  
(2.42)
\[ V_{1,lin} = \frac{1}{2} m_1 g l_1 \theta_1 + \frac{1}{2} k_1 \theta_1^2 \]  
(2.43)
\[ T_{2,\text{lin}} = \frac{1}{2} J_{2,22} (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} m_2 \left[ \dot{\theta}_1^2 \left( \frac{1}{4} l_1^2 + \frac{1}{4} l_2^2 \right) \right. \\
+ l_1 l_2 \left( \dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2 \right) (\theta_1 + \theta_2) + \frac{1}{2} l_2^2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{4} l_2^2 \dot{\theta}_2^2 \]  
\[ (2.44) \]

\[ V_{2,\text{lin}} = m_2 g \left( l_1 \theta_1 + \frac{1}{2} l_2 \right) + \frac{1}{2} k_2 \theta_2^2 \]  
\[ (2.45) \]

\[ T_{3,\text{lin}} = \frac{1}{2} m_3 \left[ \dot{\theta}_1 l_1 \left( 1 - \frac{1}{2} \theta_1^2 \right) + \left( \dot{\theta}_1 + \dot{\theta}_2 \right) l_2 \theta_2 \right]^2 \]  
\[ (2.46) \]

\[ V_{3,\text{lin}} = m_3 g \left( l_1 \theta_1 + l_2 \right) + \frac{1}{2} k_3 \theta_2^2 \]  
\[ (2.47) \]

\[ T_{4,\text{lin}} = \frac{1}{2} m_4 \left[ \frac{1}{4} l_3^2 \left( 1 - \frac{1}{2} \theta_4^2 \right)^2 \dot{\theta}_4^2 + \left[ \left( l_1 \left( 1 - \frac{1}{2} \theta_4^2 \right) + l_2 \theta_2 \right) \dot{\theta}_4 \right. \right. \right. \]
\[ + l_2 \theta_2 \dot{\theta}_4 - \frac{1}{2} l_4 \dot{\theta}_4 \]  
\[ \left. \left. \left. \left. \left. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \r

The final Lagrangian of the entire system is

\[ \mathcal{L} = \sum_{i=1}^{7} \mathcal{L}_i. \]  
\[ (2.50) \]

To reduce the equations of motion to desired states, the linearized constraint equations, Equations 2.38 - 2.41, as well Equation 2.37 are substituted into the Lagrangian. The resulting equation is in terms of \( \phi_5 \), \( \alpha_5 \), and \( \alpha_7 \). The equations of motion are then obtained by

\[ \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\phi}_5} \right) - \frac{\partial \mathcal{L}}{\partial \phi_5} = \frac{\partial \mathcal{L}}{\partial \phi_5} \tau_{\text{app}} + \tau_{5,\phi,\text{aero}} + \tau_{7,\phi,\text{aero}} \]  
\[ (2.51) \]

\[ \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\alpha}_5} \right) - \frac{\partial \mathcal{L}}{\partial \alpha_5} = \tau_{5,\alpha,\text{aero}} \]  
\[ (2.52) \]

\[ \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\alpha}_7} \right) - \frac{\partial \mathcal{L}}{\partial \alpha_7} = \tau_{7,\alpha,\text{aero}} \]  
\[ (2.53) \]

The actual arithmetic for these equations was performed in Maple, resulting in equations
of motion of the form:

$$M_{eom} \ddot{x}_{eom} = f_{\text{non}} (\dot{x}_{eom}, x_{eom}) + f_{\text{other}} (\dot{x}_{eom}, x_{eom}, \tau_{\text{app}}, \tau_{5,aero}, \tau_{7,aero})$$ (2.54)

where $x_{eom} = [\phi_5 \, \alpha_5 \, \alpha_7]^\top$. The complete equations can be see in Appendix A.

### 2.2 Aerodynamic Model

As the literature review in the previous section shows, the quasisteady aerodynamic model proposed by Sane and Dickinson [27] provides a reasonable approximation of the unsteady aerodynamic effects present in insect flight. Furthermore, the equations for the aerodynamic forces are written in terms of parameters that can easily be linked to the states of the equations of motion of the dragonfly MAV. Other models, like the Pitt-Peters model [28], often depend on states which are not as easily linked to the motion of the dragonfly MAV. Furthermore, often these other models do not present the equations for aerodynamic model in closed, analytical form, which is desirable if this model is to be extended as a plant model for a control system.

Sane and Dickinson’s [27] quasisteady aerodynamic model is given as follows. The lift ($f_L$) is a sum of the conventional translational lift and the rotational lift generated by the delayed stall effect. The magnitude of the translational lift is given by [27]

$$f_{L,\text{trans}} = \frac{1}{2} \rho U_{cp}^2 S C_l(\alpha_e)$$ (2.55)

where, $\rho$ is the density of the fluid, $S$ is the area of the wing, $U_{cp}$ is the speed of the flow at the centre of pressure, $C_l(\alpha_e)$ is the lift coefficient as a function of $\alpha_e(t)$, the effective angle of attack. The the translational lift is perpendicular to the direction of $U_{cp}$.

The magnitude of the rotational lift is given by [27]

$$f_{L,\text{rot}} = \frac{1}{2} \rho U_{cp} \omega_w \hat{c}_{max} C_{rot}$$ (2.56)

and it acts perpendicular to the surface of the wing. $C_{rot}$ is the rotational lift coefficient, $\hat{c}$ is calculated as follows, $c_{max}$ is the maximum chord length of the wing, $\phi(t)$ is the stroke angle of the wing and $\omega_w$ is the angular velocity of the wing. The rotational lift coefficient is given by the following approximation [4]

$$C_{rot} = 2\pi \left(\frac{3}{4} - \hat{x}_o\right)$$ (2.57)
where \( \hat{x}_o \) is the dimensionless distance from the elastic axis to the leading edge. For insect wings, it is conventionally assumed to be at the quarter chord \([4]\). The parameter \( \hat{c} \) is given by

\[
\hat{c} = \frac{\int_0^L c(r)^2 r dr}{\hat{r}_2 L S c_{\text{max}}} \tag{2.58}
\]

where \( L \) is the length of the wing and \( \hat{r}_2 \) is the second moment of area,

\[
\hat{r}_2 = \frac{\int_0^L c(r)r^2 dr}{L^2 S} \tag{2.59}
\]

The drag component of the force \( (f_D) \) is generated only by conventional translational aerodynamics and acts parallel to \( U_{cp} \) \([27]\)

\[
f_{D,\text{trans}} = \frac{1}{2} \rho U_{cp}^2 S C_d(\alpha_e) \tag{2.60}
\]

where \( C_d(\alpha_e) \) is the coefficient of drag as a function of \( \alpha_e(t) \). See Figure 2.4 to see an illustration of the directions of the aerodynamic forces. The speed of the flow at the centre of pressure of the wing, \( U_{cp} \), is given by

\[
U_{cp} = \| \vec{v}_0 - \rho_{cp} \times (\vec{\omega}_0 + \vec{\omega}_{\text{int}}) \| \tag{2.61}
\]

The effective angle of attack is given as

\[
\alpha_e = \arctan \left( \frac{U_{cp,\perp}}{U_{cp,\parallel}} \right) \tag{2.62}
\]

The equations for coefficients of translational lift and drag, \( C_{L,\text{trans}} \) and \( C_{D,\text{trans}} \) are obtained from the empirical data collected by Wakeling and Ellington \([1]\). Using MATLAB, the raw data from Wakeling and Ellington’s experiments were curve-fit to a sinusoidal function.
Figure 2.5: The results from curve-fitting the empirical data from Wakeling and Ellington [1] to find the equation for the coefficient of lift.

using a least-squares fitting technique (see Figures 2.5, 2.6). The resulting functions for $C_{L,\text{trans}}$ and $C_{D,\text{trans}}$ were then used to calculate the aerodynamic forces in the simulation.

This work also uses Deng’s [4] quarter chord assumption for $\hat{x}_o$ to determine the coefficient for the rotational lift. The parameter $\hat{c}$ is calculated numerically in MATLAB using parameters provided by Peter Szabo [31].
Figure 2.6: The results from curve-fitting the empirical data from Wakeling and Ellington [1] to find the equation for the coefficient of drag.
Chapter 3

Simulation Results

As previously stated, at this stage of the dragonfly MAV project the main purpose of the dynamical model is to be used for tuning the design of the prototype to meet the desired performance. Primarily the actuator forcing amplitude and frequency that generates the desired wing kinematics (and therefore the desired vertical force) need to be determined by the prototype developers.

3.1 Simulation Set-Up

The equations of motion from Appendix A were formulated as a system of ordinary differential equations (ODEs), were coded into MATLAB and solved using the built-in ode113 solver, which is an implementation of a variable-step, variable-order Adams-Bashforth-Moulton solver. From the output wing kinematics, the aerodynamic forces were calculated again in the inertial frame for plotting.

The simulation models the behaviour of the prototype labelled internally as 2P15. The parameters for each of the components of this prototype are listed in Tables 3.1 and 3.2, which were provided by Peter Szabo from his SolidWorks models of the prototype, [31], [30].

Currently, the peak actuator driving-force amplitude is only measured to be 1 mN, but is desired to be 10 mN. Both of these values were used in the simulation. The forcing frequency was varied from 10 Hz to 70 Hz, in 10 Hz increments to see how the prototype responded to the forcing frequency. These frequencies were chosen to be centred on the desired flapping frequency of *Sympetrum sanguineum* (40 Hz). The complete simulation results can be found in Appendix B.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1$</td>
<td>0.16e-3</td>
<td>kg</td>
</tr>
<tr>
<td>$l_1$</td>
<td>14e-3</td>
<td>m</td>
</tr>
<tr>
<td>$k_1$</td>
<td>0.03629</td>
<td>$\frac{kg \cdot m}{s^2}$</td>
</tr>
<tr>
<td>$J_{1,22}$</td>
<td>0.481e-9</td>
<td>$kg \cdot m^2$</td>
</tr>
<tr>
<td>$m_2$</td>
<td>0.00486e-3</td>
<td>kg</td>
</tr>
<tr>
<td>$l_2$</td>
<td>3e-3</td>
<td>m</td>
</tr>
<tr>
<td>$k_2$</td>
<td>0.0000014</td>
<td>$\frac{kg \cdot m}{s^2}$</td>
</tr>
<tr>
<td>$J_{2,22}$</td>
<td>0.00368e-9</td>
<td>$kg \cdot m^2$</td>
</tr>
<tr>
<td>$m_3$</td>
<td>0.0119e-3</td>
<td>kg</td>
</tr>
<tr>
<td>$k_3$</td>
<td>0.0000014</td>
<td>$\frac{kg \cdot m}{s^2}$</td>
</tr>
<tr>
<td>$m_4$</td>
<td>0.00324e-3</td>
<td>kg</td>
</tr>
<tr>
<td>$l_4$</td>
<td>3e-3</td>
<td>m</td>
</tr>
<tr>
<td>$k_4$</td>
<td>0.0000014</td>
<td>$\frac{kg \cdot m}{s^2}$</td>
</tr>
<tr>
<td>$J_{4,11}$</td>
<td>0.00245</td>
<td>$kg \cdot m^2$</td>
</tr>
<tr>
<td>$m_6$</td>
<td>0.00324e-3</td>
<td>kg</td>
</tr>
<tr>
<td>$l_6$</td>
<td>3e-3</td>
<td>m</td>
</tr>
<tr>
<td>$k_6$</td>
<td>0.0000014</td>
<td>$\frac{kg \cdot m}{s^2}$</td>
</tr>
<tr>
<td>$J_{6,11}$</td>
<td>0.00245</td>
<td>$kg \cdot m^2$</td>
</tr>
</tbody>
</table>

Table 3.1: Physical parameters of the actuator and transmission from prototype 2P15, courtesy Peter Szabo.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_5$</td>
<td>2.661e-6</td>
<td>kg</td>
</tr>
<tr>
<td>$J_5$</td>
<td>$\begin{bmatrix} 0.815647e-9 &amp; -0.075660e-9 &amp; 0 \ -0.075660e-9 &amp; 0.019660e-9 &amp; 0 \ 0 &amp; 0 &amp; 0.835289e-9 \end{bmatrix}$</td>
<td>$kg \cdot m^2$</td>
</tr>
<tr>
<td>$l_5$</td>
<td>0.65e-3</td>
<td>m</td>
</tr>
<tr>
<td>$m_7$</td>
<td>2.661e-6</td>
<td>kg</td>
</tr>
<tr>
<td>$J_7$</td>
<td>$\begin{bmatrix} 0.815647e-9 &amp; 0.075660e-9 &amp; 0 \ 0.075660e-9 &amp; 0.019660e-9 &amp; 0 \ 0 &amp; 0 &amp; 0.835289e-9 \end{bmatrix}$</td>
<td>$kg \cdot m^2$</td>
</tr>
<tr>
<td>$l_7$</td>
<td>0.65e-3</td>
<td>m</td>
</tr>
<tr>
<td>$x_{cm}$</td>
<td>1.74e-3</td>
<td>m</td>
</tr>
<tr>
<td>$y_{cm}$</td>
<td>15.53e-3</td>
<td>m</td>
</tr>
<tr>
<td>$k_{\phi}$</td>
<td>0.0000014</td>
<td>$\frac{kg \cdot m}{s^2}$</td>
</tr>
<tr>
<td>$k_{\alpha}$</td>
<td>0.00000171</td>
<td>$\frac{kg \cdot m}{s^2}$</td>
</tr>
</tbody>
</table>

Table 3.2: Physical parameters of the wings from prototype 2P15, courtesy Peter Szabo.
3.2 Simulation Results and Discussion

Using the current actuator force, 1 mN, and the desired forcing frequency, 40 Hz, the simulation should show the behaviour of prototype 2P15. The resulting wing kinematics can be seen in Figure 3.1. These wing kinematics are very small and periodic “beats” can be seen. The system also shows damped behaviour, which can be seen more readily in the plot of the aerodynamic forces created by the wing flapping (Figure 3.2). This damping appears to indicate that the actuator is not generating enough force to overcome the aerodynamic forces that damp the system. Furthermore, the vertical force generated by the simulated prototype is also very small and is clearly not enough for lift off (i.e., overcome the normal force on the prototype due to gravity). The prototype would need to generate 1.8 mN (calculated from the total mass of the prototype) for lift off.

If the actuator forcing amplitude is increased in the simulation to the desired value of 10 mN, it can be seen that the amplitudes of the wing kinematics increase and the damping is not as strong, Figures 3.3 - 3.4. However, there is still not enough force generated for lift off.

Without varying the physical parameters of this prototype, the actuator forcing frequency can also be varied to see if larger wing angles and more force can be generated. Figure 3.5 shows the variation of the maximum stroke angle and the maximum geometric angle of attack of the left wing as the actuator drive frequency varies. For the stroke angle, there is a clear jump at 30 Hz for both actuator force amplitudes. Otherwise, there is not much variation in the maximum stroke angle. The maximum geometric angle of attack also shows a large jump at 30 Hz for both actuator force amplitudes. However, it responds differently to increasing frequencies at the different actuator force amplitudes. At 1 mN, the geometric angle of attack increases slightly with frequency, not including the sudden jump at 30 Hz. At 10 mN, it increases until the actuator frequency reaches 60 Hz, before decreasing slightly at 70 Hz. In all cases however, the peak vertical force generated is still less than what is needed for liftoff. According to the results of these simulations, both actuator performance and the total mass of the system must be drastically improved.

This jump in the stroke angle and the geometric angle of attack is also reflected in the amplitude of the vertical forces generated by the simulated prototype, seen in Figures 3.6 and 3.7. In addition at 1 mN driving force amplitude, the peak vertical force generated by the simulated prototype also follows a similar trend as the geometric angle of attack with varying frequency – namely the amplitude of the total vertical force increases as the frequency is increased. However, at 10 mN actuator force amplitude, the peak generated vertical force increases until the actuator frequency reaches 60 Hz, before decreasing slightly at 70 Hz. In all cases however, the peak vertical force generated is still less than what is needed for liftoff. According to the results of these simulations, both actuator performance and the total mass of the system must be drastically improved.

Returning to the jump in kinematics and forces when the actuator forcing frequency was 30 Hz, it should also be noted wing kinematics were not nicely periodic, as seen in Figure 3.8.
Figure 3.1: Simulated kinematics of the left wing with an actuator driving-force amplitude of 1 mN and a forcing frequency of 40 Hz.

Figure 3.2: The aerodynamic forces generated by the simulated wing kinematics with an actuator driving-force amplitude of 1 mN and a forcing frequency of 40 Hz. These forces are presented in the inertial frame.
Figure 3.3: Simulated kinematics of the left wing with an actuator driving-force amplitude of 10 mN and a forcing frequency of 40 Hz.

Figure 3.4: The aerodynamic forces generated by the simulated wing kinematics with an actuator driving-force amplitude of 10 mN and a forcing frequency of 40 Hz. These forces are presented in the inertial frame.
Figure 3.5: The variation of the maximum stroke angle and the geometric angle of attack for the left wing with changing actuator drive frequencies.
Figure 3.6: Forces in the vertical direction calculated from the simulated wing dynamics where the actuator driving-force amplitude was set to 1 mN.

Figure 3.7: Forces in the vertical direction calculated from the simulated wing dynamics where the actuator driving-force amplitude was set to 10 mN.
Can such behaviour be seen in the prototype? With the current experimental setup for this project, while the vertical forces generated by the prototype can be measured, there is no way to determine the wing kinematics simultaneously. Without such a method, it is difficult to say whether such behaviour is also seen in the prototype. There could be a variety of reasons for this issue. It could simply be a numerical instability in the ODE-solver or it could be a fundamental frequency of the system. A more far-fetched theory is that the derived system of equations is chaotic and this frequency is the parameter that displays chaotic behaviour. If this is a fundamental aspect of the system, it should be also seen in the prototype. It is clear that this issue needs more investigation.

Furthermore, if the simulated model cannot be verified against a physical prototype, is there any way to tell if the model is correct? A method to determine the wing kinematics of the prototype will be addressed in the second part of this thesis, but in the meantime, it can be determined that the simulation is internally consistent by examining the total energy of the system and the work done by the applied forces. The total energy of the system (the sum of the kinetic and potential energy) minus the work will be constant. In this simulation, the kinetic and potential energies are given in the derivation of the equations of motion. The work is added as the last state in the equations of motion such that

$$\dot{W} = \tau_{\text{app}}\dot{\theta}_1 + (\tau_{5,\text{aero},\phi} + \tau_{7,\text{aero},\phi})\dot{\phi}_5 + \tau_{5,\text{aero},\alpha_5}\dot{\alpha}_5 + \tau_{7,\text{aero},\alpha}\dot{\alpha},$$  \hspace{1cm} (3.1)$$

where \(\tau_{i,\text{aero},\phi}\) is the component of torque acting around the x-axis of either the left or right
Figure 3.9: The total energy of the system minus the work for the case when the actuator driving-force amplitude was 1 mN and the forcing frequency was 40 Hz.

wing. Similarly, $\tau_{i,aero,\alpha}$ is the component of torque acting around the y axis of the left or right wing. In all cases, the total energy of the system minus the work was constant to within numerical error (e.g., see Figure 3.9). In this case, “constant to within numerical error” meant that the differences between $max(T + V - W)$ and $min(T + V - W)$ were on the order of $10^{-20}$ to $10^{-18}$, while the integration tolerance was set to $10^{-17}$. From this, it can be concluded that there were no errors in the derivation of the equations of motion and that the model is internally consistent.
Chapter 4

Future Work

The next logical step would be to determine the natural frequency of the system. Doing so would provide insight into whether the previously discussed unstable results are caused by forcing the system at a fundamental frequency. It would also help the prototype design process by providing actuator forcing frequencies that could generate large wing kinematics and forces. This analysis could provide insight into the stability of the overall system as well.

There are also improvements that need to made to the aerodynamic model. The current quasisteady model does not include the additional damping from the added-mass effect. A quasisteady model for this effect was derived in the literature [27], but this model would need to be extended to the dragonfly MAV. Furthermore, the aerodynamic model relies on the assumption that because the dragonfly MAV wings match the mass distribution and geometry of the wings of *Sympetrum sanguineum*, the aerodynamic characteristics would also match. This is not necessarily true. Ideally, the coefficients of translational lift and drag, and rotational lift would be determined empirically for the actual prototype wings. For the most accurate aerodynamic model, it would be necessary to move away from quasisteady models completely, as there are aerodynamics effects that remain unaccounted for [27]. However, the desire for the most accurate model would need to be tempered with the need for relatively simple equations to be used for control design.

Finally, the current model simulates the behaviour of the dragonfly MAV as strapped to a force sensor. To develop a plant model for a control system, the degrees of freedom for the motion of the body should be added. This can be done using the same method as was presented in this thesis. Furthermore, it is easy to add control inputs in the form of torques around the $\alpha_i$ DoF of each wing. This would provide active control over the angle of attack, rather than allowing the wing to rotate passively and would provide a method of attitude control for the MAV.
Chapter 5

Summary

This part of the thesis presented the derivation of a multibody dynamical model of the dragonfly MAV prototype. The model is primarily intended to be used as a tool used for testing and improving the prototype design by simulating wing kinematics generated based on the internal dynamics of the ATW system. Currently the model does not account for the motion of the body of the MAV.

The equations of motion of the ATW system were derived using the Lagrangian method. The overall Lagrangian of the system was formulated in terms of the DoFs for each component and the constraints were identified from the geometric relations. Both the Lagrangian and the constraints were linearized. Then, the constraints were substituted into the Lagrangian to reduce the equations to only the free DoFs, specifically the stroke angle of the left wing, \( \phi_5 \), and the geometric angles of attack of the left and right wings, \( \alpha_5 \) and \( \alpha_7 \).

The equations for the aerodynamic forces used in the equations of motion are based on the quasisteady model presented by Sane and Dickinson [27]. The equations coefficients of the translational lift and drag were found by curve-fitting empirical data from Wakeling and Ellington [1]. The coefficient of rotational lift was calculated using Deng’s [4] quarter chord assumption and the physical parameters of the wing.

The prototype 2P15 was simulated in MATLAB by solving the equations of motion using one of the built-in ODE solvers. The resulting wing kinematics and aerodynamic forces generated by varying the actuator driving force amplitude and frequency were presented. From these results, prototype 2P15 does not appear to be able to achieve liftoff. However, there was some unexpected behaviour in the system when the actuator driving frequency was set to 30 Hz. This behaviour requires further investigation to see if it could be exploited to generate kinematics and forces large enough for liftoff. Furthermore, the DoFs for the MAV body and the hind-wing pairs needed to be added in order to derive a complete plant model, which can then be used to design a control system. There are also improvements to be made to the aerodynamic model as not all the unsteady effects associated with insect flight are accounted for.
The model presented in this thesis differs from the others found in literature in that it accounts for the internal dynamics of the MAV. While these internal dynamics might have a minor effect when the MAV is clamped to a lab bench, they become important once the MAV is able to fly freely. At this point, the motion of the internal components, most important the actuator, will impact the overall dynamics of the system as they compose a significant percentage of the total mass. It will be necessary to consider these dynamics when designing a controller for the MAV.
Part II

Estimation
Before the dynamical model of the dragonfly MAV can be used for controller design or prototype parameter optimization, it must be verified. Since the motion of the wings and the aerodynamic forces are interdependent, both the aerodynamic forces and the wing kinematics need to be measured simultaneously to verify all aspects of the model. The aerodynamic forces are measured indirectly by attaching the MAV prototype to a force sensor and measuring the body forces as the wings flap. The wing kinematics, namely the stroke angle and geometric angle of attack, will be measured using an algorithm that will be developed in the following chapters. This algorithm can also be used for validation in the future when developing new aerodynamic models and when testing prototype performance.

The problem of determining an object’s motion from measurements has been well-researched. When specifically dealing with the motion of insects and insect-inspired MAVs, the small sizes mean that most sensors cannot be used. Some researchers specifically build oversized prototypes ([11], [32]) in order to use systems like the VICON system which requires adding markers to the wings. However, to avoid determining whether the dynamics of the system scale with prototype size, something that would be quite difficult to determine, this thesis constrains itself to working directly with the MAV prototype. Another constraint is that the frequency of wing flapping results in very fast motion; any technique used to measure the wing kinematics requires significant bandwidth to achieve a desirable resolution. Insect flight researchers have dealt with these two constraints by using high-speed cameras and visual techniques.

This part of the thesis will begin by outlining some of the methods that have been used to determine insect wing kinematics in the past. The estimation algorithm used for the dragonfly MAV project will then be derived. The algorithm was tested on simulated datasets, and these results will be discussed. This part of the thesis will conclude with some suggestions for future work.
Chapter 6

Literature Review

As previously seen, insect flight has long fascinated researchers. Early quantitative studies of insect flight typically measured lift forces; however, some studies also attempted to quantify the wing kinematics [33], [34]. These studies used simple geometric tools such as protractors (Weis-Fogh in 1956 [33]) and haversine nomographs\(^1\) (Vogel in 1966 [34]) on still images of insects in flight. Both Weis-Fogh and Vogel obtained the images using strobe-synchronized cameras to record images over multiple wing-beat cycles. Because the images came from different points in multiple stroke cycles, these early works only reported the kinematic extremes of the wing motion and could only hypothesize the time-varying continuous motion.

As technology advanced, researchers began to use high-speed video cameras to record continuous sequences of insect flight. Thus it also became possible to use more advanced techniques to measure the wing kinematics. These methods can be classified as either single- or multicamera methods.

\subsection*{6.1 Single-Camera Motion Estimation Techniques}

Initially, most researchers used single high-speed video cameras to determine wing kinematics, possibly because of budgetary constraints. Even today, good quality high-speed video cameras are expensive. The primary issue with using a single, or monocular, camera to recover the 3D motion of an object is that the object’s position and orientation can only be determined up to a scale factor. This is why stereo cameras are commonly used for these kinds of problems. To overcome this deficiency, researchers had to reintroduce scale into the images using different techniques including making assumptions about the wing motion and geometry, using specialized camera calibration routines or projecting known patterns over the insect.

Ellington in 1984 studied the kinematics of a variety of insect in free flight using a single high-speed camera [36]. Since the insects’ motion was not constrained in any way, it was

\(^1\)Haversine nomographs are devices used in spherical trigonometry that can determine angles visually, somewhat akin to a 3D protractor [35].
necessary to make some assumptions to determine what wing motions were possible. Ellington
chose to assume that the wing motion was bilaterally symmetric, or symmetric around the
long axis of the insect body, to determine the stroke angle of the wing [36]. This is a poor
assumption since it relies on the researchers to observe in which situation the wings are
moving symmetrically. Since insect wing kinematics can be small and subtle, it can be hard
to distinguish asymmetries. Ellington’s technique also required manual selection of points
of interest (PoIs) on the wing as well as knowledge of the wing length. Using the apparent
wing length in each frame, Ellington calculated the position of the wing tip in a 3D inertial
reference frame. He then used the symmetry assumption to define the motion of the wing tip
with respect to the body [36]. Dudley and Ellington [37] expanded this method to include
calculating the pitching angle of the wing; however, this updated method still required the
bilateral symmetry assumption.

In 1994, Berg developed a similar geometric method to measure rotations that did not
require the symmetry assumption [38]. Berg’s method required that the magnification of
the lens was known, that two PoIs were visible for each rotation to be measured and that
the distance between the PoIs were known. This last point ensured that scale in the images
could be determined. Once the PoIs were manually found and digitized, the angle of rotation
could be calculated using basic geometry [38].

A later study by Willmott and Ellington was also able to dispense with the symmetry
assumption to determine the pitching angle of the wing [39]. Willmott and Ellington did
so by modelling the wing as a set of rigid strips that were free to rotate around the elastic
axis of the wing. This meant that the planform profile of the wing, most importantly the
chord length as a function of the span, had to be determined \emph{a priori}. They then overlaid
the strips on the images, rotating them until the model wing aligned with the image [39].
This was done manually.

Researchers have also used custom camera calibration techniques with single cameras
to introduce a known scale into images. For example, Russell [40] used a custom-made 3D
calibration cube to calibrate the camera in a such a way as to determine scale. This allowed
the 3D position of PoIs on the wing to be determined. His calibration method defined three
reference frames: the global frame whose origin was located at the back left corner of the
 calibration cube, local frames which were aligned with each face of the cube, and the image
frame (or pixel frame) that was located on the image plane. Using the calibration cube,
the transformations from the image frame to the local frame and from the local frame to
the global frame were found. These successive transformations were used to express the 3D
position of a PoI on the wing in a local frame on one of the cube faces. The positions of the
point in local frames on opposite cube faces were used to form a line passing through the
cube. In theory, the intersection of two lines formed from two pairs of opposite faces would
give the position of the PoI in 3D space. In practice though, the lines do not intersect; as
an approximation, the point on one line that is closest to the second line is found (and vice versa). The average of these two points is used as the 3D position of the PoI [40]. Assuming the wing was represented as a triangular plate whose vertices were three PoIs on the wing the wing kinematics could also be calculated. The error in the stroke angle was estimated to be $\pm 3^\circ$ in the worst case and $\pm 1^\circ$ typically, while the error in the angle of attack was estimated as $\pm 12^\circ$ in the worst case and $\pm 4^\circ$ typically [40]. It is important to note that procedure for acquiring these error estimates was never presented.

Another way to determine scale in images from a single camera is to project a known pattern onto the scene. Projected comb fringe patterns were most commonly used in insect flight studies. The earliest form of this technique used two fringe patterns that were projected orthogonally to each other to determine the stroke angle and angle of attack of the wing [41]. When the wing was assumed to be a rigid plate, the stroke angle of the wing was measured directly using the shadow created by the wing on one of the projected comb fringes. The angle of attack of the wing was calculated using the chord length of the wing shadow in the two projections; however, this calculation did not account for the stroke angle of the wing. The authors of this study claimed that the standard deviation of the resolution was $0.9^\circ$ for the stroke angle and $5^\circ$ for the angle of attack [41].

A more advanced version of this technique dispensed with the rigid plate assumption and instead measured the wing camber as well the wing kinematics [42]. In this study, the distortion of the projected comb fringe patterns were measured directly rather than using the wing shadow. The accuracy of the measurement of points on the wing had a standard deviation of 0.05 mm [42]. This study explicitly measured wing camber, but the same technique was used to measure wing kinematics of dragonflies in forward flight and turning manoeuvres by Wang et al. [43].

It is important to note that all these single-camera techniques are based on purely geometric, deterministic methods. Furthermore, the errors and uncertainties calculated for these methods were derived similarly and are constant for the flapping sequence. This is an issue because certain orientations of the wing should be more difficult to determine using the same PoIs and this should be reflected in the errors and uncertainties.

### 6.2 Multicamera Motion Estimation Techniques

Many of the techniques described above used single cameras because of budgetary constraints. By moving to multicamera techniques, researchers were able to relax some of the assumptions made in the previous studies and make more general measurements. For example multicamera techniques allow for unknown positions of PoIs on the wing; in some of the previous techniques the positions of the PoIs had to be known. These multicamera techniques range from simple image matching to more advanced techniques based on photogrammetry to novel hull
reconstruction methods.

At the most basic level, researchers constructed models of the insect wings and bodies and manually overlaid these models on the images obtained from three orthogonal cameras [44], [45]. When the model matched all three images, the kinematic values were simply read off. Fry et al. [44] did not provide error estimates for their method. Mou et al. [45] had mean errors of $\pm 3^\circ$ for the stroke angle and $\pm 5^\circ$ for the pitching angle of the wing, because the image matching was done manually. This error was calculated using the Reprojected Pixel Error (RPE). The RPE is given by the difference between the measured pixel coordinates of a point and the pixel coordinates of the same point projected back into the image plane using the estimated wing motion [46].

More sophisticated techniques used photogrammetry algorithms originally developed for map-making. One of the earliest photogrammetric-based techniques was presented by Zanker [47]. In this study, a single camera and mirrors were used to simulate four different camera views of a tethered insect. Using a 3D cross for calibration, Zanker was able to determine the transformation from the coordinates of a point in a 3D reference frame to the coordinates in the image space for each “camera”. Then using any two “camera” images, specifically the two with the best view of the wings, he was able to calculate the 3D position of any PoI on the wing. To calculate the kinematics, he manually identified six PoIs (vein intersections that were easily identifiable in the images) and calculated their 3D positions at different times in the wing beat. These six PoIs were fit with a regression plane which represented the plane of the wing (the wing was assumed to be a flat plate). By projecting each point onto the regression plane and using a predefined wing-fixed reference frame, the kinematics of the wing were calculated. The resulting errors from this technique ranged from $\pm 1.1^\circ$ to $\pm 2.8^\circ$ for the stroke angle and from $\pm 1.6^\circ$ to $\pm 4.0^\circ$ for the angle of attack of the wing [47].

Walker et al. used a more modern photogrammetric technique (based on the bundle adjustment algorithm formulation by Hartley and Zisserman [48]) to measure the twist and camber of the wing [46]. Unlike some of the previous methods that required a separate camera calibration routine, the bundle adjustment algorithm can estimate the camera parameters as well as the positions of the PoIs simultaneously in a nonlinear optimization. However, to improve this method, Walker et al. first used a calibration grid with known PoIs to find the camera parameters (i.e., calibrate the camera) and then used the camera parameters to solve for the 3D positions of the PoIs on the wing. This allowed the error in the image coordinates of one PoI to remain independent of the accuracy of the estimated position of the another PoI. This study was also notable because it was one of the few to have semiautomated tracking of the PoIs; typically that was done manually. The errors were calculated using the RPE like Mou et al. and the standard deviation of the error was less than 0.88° for all angles and all trials [46].

The previous two studies both required that the insect’s body was fixed relative to the
camera. Moving away from photogrammetry-inspired algorithms, researchers found other techniques to determine the wing kinematics of free-flying insects. One such technique is called Hull Reconstruction Motion Tracking (HRMT). While HRMT was originally used for other applications, it was first used to study insect flight by Ristroph et al. [49]. This process reconstructed a 3D model of the insect, which was then broken down into the body and the wings. HRMT for insect flight was completely automated and required no manual input. Ristroph et al. used three high-speed cameras that were precisely aligned so that the centre of the camera view was the same point in space to within a few pixels. Each camera was strongly backlit using projectors so that the insect was always viewed in silhouette. The background of each image could then be subtracted and the remaining image was thresholded to black and white. Using these black and white images, a bounding box, or region of interest, was created around each silhouette. The region of interest was then scaled and translated so that the pixel coordinates of the bounding box match appropriately. To reconstruct the hull, or the volume that that represents the 3D model of the insect, the volume pixels whose 2D projections were in each of the three silhouettes were found. A volume pixel was defined as a cube of dimension 2 pixel $\times$ 2 pixel $\times$ 2 pixel. This large grouping of volume pixels was then broken into smaller clusters representing the body and wings by using an algorithm based on Euclidean distance. Using principal component analysis and some geometric information about the insect to calculate the principal axes of inertia and the centroids of each cluster, Ristroph et al. estimated the kinematics of the body and the wings. The error in the angular measurements that resulted from this technique could be more than 5°, which is larger than the errors from the other multicamera techniques discussed previously. This is partially caused by allowing the insects to fly freely, since certain body orientations obscure the wings [49].
Chapter 7

Kinematics Estimation Algorithm

In general, there is a recent trend to find more rigorous methods of determining wing kinematics. The paper by Walker et al. [46] exemplifies this since they treat the problem as a Bundle Adjustment, or Simultaneous Localization and Mapping (SLAM), problem and apply a classic state estimation technique to solve it. However because of budget constraints, this thesis is restricted to working with a single high-speed camera and cannot treat estimating the wing kinematics as a conventional SLAM problem. Nonetheless, the philosophy of treating this like a classical state estimation problem and the corresponding probabilistic solution methods can be applied. This is a different approach, unlike the geometric, deterministic techniques used by previous researchers who also depended on single-cameras. The advantage to using probabilistic methods is they provide better uncertainty estimates that reflect the configuration-dependant difficulties of the problem (i.e., certain poses are harder to estimate).

In this context, estimation is the problem of determining the state of the system from noisy measurements [50]. This is a common problem in robotics since the robot’s pose and velocity need to be known for problems like path planning. Most robotic applications use filters, such as the classic Kalman filter, for state estimation since they require real-time knowledge of the robot’s motion. As we are primarily estimating the wing kinematics for model verification, we do not require online state estimation. This means we can use a batch estimator (equivalent to the canonical fixed-interval smoother). Batch estimators are generally accepted to give the best solution as they use every measurement for the estimate of every state [51]. In a filter, only the past and current measurements are used to estimate the state at a particular point in time. In fact, batch estimators are so desirable that there is research into incorporating batch filters into online estimation frameworks for problems like SLAM [51], [52]. Batch estimators are also easy to modify so that they do not require a motion model. Since the kinematics estimation algorithm is being used to verify the motion model of the wing, we do not want to use it as part of the estimation algorithm. If the motion model is used in the estimation algorithm is incorrect, it can result in very unpredictable estimator results.
7.1 Batch Maximum Likelihood Estimator Derivation

To begin deriving the batch estimator, the measurement model for the system needs to be defined:

\[ y_k = g(x_k) + n_k \]  

(7.1)

where \( y_k \) is the measurement at time-step \( k \), \( g(x_k) \) is a nonlinear function of the state \( x \) at time-step \( k \) and \( n_k \) is the Gaussian noise on the measurement such that \( n_k \sim \mathcal{N}(0, Q) \). The batch estimator is framed as a maximum likelihood estimator (MLE), where the estimated state maximizes the likelihood function of the state given the measurements \([53]\). The estimated state is defined as \( X^* = \{x_1^*, \ldots, x_k^*, \ldots, x_N^*\} \) such that

\[ X^* = \arg \max_X \mathcal{L}(X|Y), \]  

(7.2)

where \( X = \{x_1, \ldots, x_k, \ldots, x_N\} \) is the set of states to be estimated, \( Y = \{y_1, \ldots, y_k, \ldots, y_N\} \) is the set of measurements and \( \mathcal{L}(X|Y) \) is the likelihood of the all the states given the measurements. The likelihood function \( \mathcal{L}(X|Y) \) is identically equal to the conditional probability of \( Y \) given \( X \) \([53]\):

\[ \mathcal{L}(X|Y) \equiv p(Y|X) = \prod_{k=1}^{N} p(y_k|x_k). \]  

(7.3)

If \( x^* \) is defined as the column of the states, \( x^* \triangleq \text{col}\{x_k^*\} \), (7.2) can be rewritten as

\[ x^* = \arg \max_x p(Y|X) \]

\[ = \arg \min_x -\ln \prod_{k=1}^{N} p(y_k|x_k), \]  

(7.4)

where

\[ p(y_k|x_k) = \eta e^{[y_k - g(x_k)]^\top Q^{-1} [y_k - g(x_k)]} \]  

(7.5)

and \( \eta \) is the normalization coefficient of the normal distribution for the measurement noise. Then,

\[ x^* = \arg \min_x -\sum_{k=1}^{N} [y_k - g(x_k)]^\top Q^{-1} [y_k - g(x_k)] - \ln \eta. \]  

(7.6)

Noting that the term \( \ln \eta \) is not dependent on the state, from (7.6) an optimization cost
function is defined [53]:

\[ J(x_k) \triangleq - \sum_{k=1}^{N} [y_k - g(x_k)]^\top Q^{-1} [y_k - g(x_k)] \]  

(7.7)

Defining the error \( e_k \) as the difference between the actual measurement and the estimated measurement calculated from the estimated state,

\[ e_k(x_k) = y_k - g(x_k) \]  

(7.8)

(7.7) can be rewritten in terms of \( e_k \). Then if the error terms are stacked for each time step, the cost function becomes

\[ J(x) = -e(x)^\top T^{-1} e(x) \]  

(7.9)

where,

\[ x \triangleq \text{col} \{x_k\} \]  

(7.10)

\[ e \triangleq \text{col} \{e_k\} \]  

(7.11)

\[ T \triangleq \text{diag} \{Q\} \]  

(7.12)

The Gauss-Newton optimization algorithm to find \( x^* \) requires a cost function of the form \( a^\top a \) [53]. Thus, define a substitute variable \( v(x) \) as

\[ v(x) = Le(x), \quad L^\top L = T^{-1}. \]  

(7.13)

The matrix \( L \) can be found from the Cholesky decomposition of \( T^{-1} \) (or from the eigenvalues of \( T^{-1} \) since it is positive-definite and symmetric). The cost function, (7.9), is then written as

\[ J(x) = v(x)^\top v(x). \]  

(7.14)

The Gauss-Newton optimization algorithm can now be applied. The algorithm starts from some initial condition \( \bar{x} \) and calculates an update step based on the approximate Jacobian and Hessian matrices [53].

Beginning by taking a Taylor expansion of \( v(x) \),

\[ v(\bar{x} + \delta x) \approx v(\bar{x}) + \left( \frac{\partial v}{\partial x} \right)_{\bar{x}} \delta x \]  

(7.15)

and substituting it into the cost function \( J(x) \):

\[ J(\bar{x} + \delta x) \approx \left( v(\bar{x}) + \left( \frac{\partial v}{\partial x} \right)_{\bar{x}} \delta x \right)^\top \left( v(\bar{x}) + \left( \frac{\partial v}{\partial x} \right)_{\bar{x}} \delta x \right). \]  

(7.16)
Algorithm 1: General Gauss-Newton optimization for MLE batch estimator

1. Initialize \( \bar{x} \), \( \delta x^* \), \( \alpha \);
2. while \( | J(\bar{x} + \alpha \delta x^*) - J(\bar{x}) | > \epsilon \) do
3. Evaluate \( H, T^{-1} \);
4. Solve for update \( \delta x^* \) using \( H^T T^{-1} H \delta x^* = -H^T T^{-1} e(\bar{x}) \);
5. if \( J(\bar{x} + \delta x^*) > J(\bar{x}) \) then
6. Find \( \alpha \) such that \( J(\bar{x} + \alpha \delta x^*) < J(\bar{x}) \)
7. else
8. \( \alpha = 1 \)
9. \( \bar{x} \leftarrow \bar{x} + \alpha \delta x^* \)
10. return \( \bar{x} \);

We then minimize the cost function with respect to \( \delta x \) to find the optimal step:

\[
\frac{\partial J(\bar{x} + \delta x)}{\partial \delta x} = \left( v(\bar{x}) + \left( \frac{\partial v}{\partial x} \bigg|_{\bar{x}} \right) \delta x^* \right)^T \left( \frac{\partial v}{\partial x} \bigg|_{\bar{x}} \right) = 0. \tag{7.17}
\]

Rearranging, the equation for the optimal update step \( \delta x^* \) is:

\[
\left( \frac{\partial v}{\partial x} \bigg|_{\bar{x}} \right)^T \left( \frac{\partial v}{\partial x} \bigg|_{\bar{x}} \right) \delta x^* = -\left( \frac{\partial v}{\partial x} \bigg|_{\bar{x}} \right)^T v(\bar{x}). \tag{7.18}
\]

Substituting \( v(x) = Lx \), the update equation becomes

\[
H^T T^{-1} H \delta x^* = -H^T T^{-1} e(\bar{x}), \tag{7.19}
\]

where

\[
H = \frac{\partial e(x)}{\partial x} \bigg|_{\bar{x}}. \tag{7.20}
\]

After solving for \( \delta x^* \), the estimate of the state is updated according to

\[
\bar{x} \leftarrow \bar{x} + \delta x^*. \tag{7.21}
\]

In practice, the optimal step may cause the value of the cost function to increase because the Jacobian is approximated [53]. To avoid this, a line-search is used to update the estimate according to

\[
\bar{x} \leftarrow \bar{x} + \alpha \delta x^*, \tag{7.22}
\]

where \( \alpha \in [0,1] \) is chosen such that \( J(\bar{x}) > J(\bar{x} + \alpha \delta x^*) \). This process continues iteratively until the difference between subsequent evaluations of the cost is lower than a user-defined threshold. Once this occurs, the state \( \bar{x} \) is the solution to the optimization. This algorithm is summarized in Algorithm 1.
7.2 Derivation of the Measurement Model and Error Jacobians

Now that the general batch MLE has been derived, a measurement model still needs to be specified, more specifically the function $g(x_k)$. Since only a single high-speed camera is used, the 3D motion of the wing can only be estimated up to a scale factor. In other words, a point in the image cannot be projected into 3D coordinates. The common solution to this limitation is to introduce known reference lengths into the object being estimated. As previously stated in the single-camera section of the literature review, insect flight researchers accomplished this in different ways. Most commonly, this was done by using PoIs on the wing. However unlike many of the previous studies, in this work the PoIs on the wing should be able to be found by a computer automatically in the video sequence. While it is possible to create PoIs on the wing using permanent markers without changing the mass properties of the wing, the accuracy of the estimate would be limited by how well the positions can be measured. Furthermore, since the estimation procedure will eventually be automated, adding PoI using permanent markers may not work since the marks are not lighting-invariant. To make features detection more robust, any method to create PoIs must not be dependent on specific lighting conditions.

The solution to these two problems is to use the existing venation structure on the wings to incorporate a known scale into the images; specifically the tips of the veins can be used since their positions are known from the SolidWorks models used to manufacture the wings. Furthermore, the position of the vein tips are constant in the wing reference frame, assuming the wing is rigid. Rigid wings are a common assumption that is used when studying insect flight. For the dragonfly prototype in particular, it is valid since the venation results in the wing structure being much stiffer than the joints that allow wing rotation. This should result in minimal twist and camber. Thus the measurements used in the estimation algorithm are the pixel coordinates of the vein tips. The function $g(x_k)$ in the measurement model is the transformation of the coordinates of the PoIs in the wing reference frame to the pixel coordinates in the image. This can be broken down into two subtransformations: the transformation from coordinates in the camera frame to pixel coordinates and the transformation from coordinates in the wing frame to coordinates in the camera frame.

The transformation from coordinates in the camera frame to pixel coordinates is called the camera model. For this estimation algorithm, Bouguet’s camera model [54] is used, which is similar to the model by Heikkila and Silven [55]. To begin, the high-speed camera is assumed to be represented by an idealized pinhole camera and that there is a reference frame whose origin is located at the optical centre of this camera (see Figure 7.1). A point expressed in
Figure 7.1: A schematic of an idealized pinhole camera. Note that the image plane is in front of the camera rather than behind the camera as expected. This is because when the image plane is behind the camera, the image projected onto the plane is flipped. Placing the image plane in front of the camera is equivalent to having the image plane behind the camera and flipping the projected image.
the camera frame is then mapped to the image plane using the perspective transform $\mathcal{P}$:

$$\mathcal{P} : \mathbb{R}^3 \rightarrow \mathbb{R}^2,$$

$$x_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = P(x_c) \triangleq \begin{bmatrix} x/z \\ y/z \end{bmatrix}, \text{ where } x_c = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (7.23)$$

The resulting point on the image plane, $x_n$, is said to be in normalized image plane coordinates.

Since modern cameras have lenses, lens distortion must also be accounted for to improve the accuracy of the model. The most common forms of distortion present in modern cameras are radial and tangential distortions. To model these two effects Brown’s plumb bob distortion model $\mathcal{D}$ [56] is used:

$$\mathcal{D} : \mathbb{R}^2 \rightarrow \mathbb{R}^2,$$

$$x_d = \left(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6\right) x_n + \begin{bmatrix} 2\tau_1 x_n y_n + \tau_2 \left(r^2 + 2x_n^2\right) \\ 2\tau_2 x_n y_n + \tau_1 \left(r^2 + 2y_n^2\right) \end{bmatrix}, \quad (7.24)$$

where $r = \|x_n\|$ and $\kappa_i = 1,...,3$ and $\tau_i = 1,2$ are the radial and tangential distortion coefficients. These coefficients are found from a camera calibration process using Bouguet’s MATLAB camera calibration toolbox [54]. This point is then mapped to pixel coordinates, $u$, where the origin is the upper-left corner of the image, using the camera transformation $\mathcal{K}$:

$$\mathcal{K} : \mathbb{R}^2 \rightarrow \mathbb{R}^2,$$

$$u = \begin{bmatrix} u \\ v \end{bmatrix} = \mathcal{K}(x_d) \triangleq \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} x_d + \begin{bmatrix} c_x \\ c_y \end{bmatrix}. \quad (7.25)$$

The focal lengths, $f_x$ and $f_y$, and the principal point, $c_x$ and $c_y$, are parameters of the camera. They are obtained from the same camera calibration process as before.

The complete camera model is given by

$$u = (\mathcal{K} \circ \mathcal{D} \circ \mathcal{P})(x_c). \quad (7.26)$$

In practice, before images are used for estimation, they are undistorted. Again this can be done using Bouguet’s toolbox. For this reason, the images are assumed to be undistorted and the final camera model becomes

$$u = (\mathcal{K} \circ \mathcal{P})(x_c). \quad (7.27)$$

The next step is to derive the transformation of coordinates in the wing frame to the camera frame. The same prototype design as in the first part of this work will be used to do
Figure 7.2: Reference frames for a forewing prototype. Note that for this prototype the stroke plane and the body frame are coincident; in other words, this prototype is mounted parallel to the ground.

so. Note that for this experimental set-up the stroke plane frame \( \mathcal{F}_s \) and the body frame \( \mathcal{F}_b \) are assumed to be aligned, see Figure 7.2.

Since the wings are considered to be rigid, the coordinates of the vein tips in the wing-fixed reference frame, \( \mathbf{\rho}^i_w \), are constant. The superscript \( j \) is the index used to distinguish each vein tip, while the subscript \( w \) refers to the reference frame in which the position is expressed. It should also be noted that the only relative motion between each wing and the body is the rotation around the wing hinge, where the origin of the wing-fixed reference frame is located. The coordinates of each feature in the body frame are given by:

\[
\mathbf{\rho}^i_{b,k} = \mathbf{C}^\top_{wb,k} \mathbf{\rho}^i_w,
\]

(7.28)

where \( \mathbf{C}^\top_{wb,k} \) is the rotation from the wing frame to the body frame at time step \( k \).

Since this kinematic estimation method will be performed in conjunction with force measurements, the prototype will be fixed to the force sensor. Thus, the body of the prototype will be fixed in translation and rotation with respect to the camera frame, see Figure 7.3. We then express the vein tip coordinates in the the camera frame:

\[
\mathbf{\rho}^i_{c,k} = \mathbf{C}^\top_{bc} (\mathbf{\rho}^{O,\mathcal{O}_w}_b + \mathbf{C}^\top_{wb,k} \mathbf{\rho}^i_w)
\]

(7.29)

where \( \mathbf{C}^\top_{bc} \) is the rotation matrix from the body frame to the camera frame and \( \mathbf{\rho}^{O,\mathcal{O}_w}_b \) is translation from the origin of the wing frame to the origin of the camera frame, expressed in the body frame. The measurement model for each PoI becomes

\[
\mathbf{y}^i_k = (\mathcal{K} \circ \mathcal{P}) (\mathbf{\rho}^i_{c,k}) + \mathbf{n}_k.
\]

(7.30)

The wing kinematics are represented by the rotation from the wing frame to the body frame \( \mathbf{C}^\top_{wb,k} \). If this rotation matrix is represented using the 1-2-3 Euler angle set, the state
Figure 7.3: A schematic of the experimental set-up used to estimate the wing kinematics. In this case forewing prototype is being tested. The prototype is mounted beneath the camera on the force sensor.
to be estimated is
\[
\theta_k = \begin{bmatrix}
\phi_k \\
\alpha_k \\
\psi_k
\end{bmatrix}
\] (7.31)
where \(\phi\) is the stroke angle, \(\alpha\) is the geometric angle of attack and \(\psi\) is the sweep angle of the wing. While the sweep angle is expected to be zero due to the mechanical design of the prototype, this angle is still estimated to verify the prototype is behaving as expected. If the prototype had a nonzero stroke plane, the rotation matrix \(C^T_{wb,k}\) would simply become \(C^T_{wb,k} = C^T_{eb}C^T_{ws,k}\). The state to estimate would then be \(C^T_{ws,k}\), which is represented by the same Euler angle set.

However, the rotation \(C^T_{wb,k}\) is not the only state that requires estimation; the translation from the origin of the wing frame to the camera frame \(\rho^O_{o,w}\) is difficult to measure directly and must be estimated. This is because the origin of the camera frame is located at the optical centre of the camera, somewhere inside the lens. Thus, the final state to be estimated is
\[
x_k = \begin{bmatrix}
\rho^O_{o,w} \\
\theta_k
\end{bmatrix}.
\] (7.32)

The measurement model can then be rewritten as
\[
y^j_k = g(x_k, \rho^j_w) + n_k,
\]
\[
g(x_k, \rho^j_w) \triangleq (\mathcal{H} \circ \mathcal{P}) (\rho^j_{c,k})
\] (7.33)
and the error for each measurement at time-step \(k\) is
\[
e^j_k = y^j_k - g(x_k, \rho^j_w).
\] (7.34)
To rewrite the measurement errors in the form used in the derivation of the Gauss-Newton optimization, all the measurements at each time step are stacked to form the column \(e_k\):
\[
e_k \triangleq \text{col} \{e^j_k\}.
\] (7.35)

Now that the error function is defined, the next step is to calculate the Jacobian \(H\), defined as
\[
H \triangleq \left. \frac{\partial e}{\partial x} \right|_x = \begin{bmatrix}
\frac{\partial e_1}{\partial x_1} & \frac{\partial e_1}{\partial x_2} & \cdots & \frac{\partial e_1}{\partial x_N} \\
\frac{\partial e_2}{\partial x_1} & \frac{\partial e_2}{\partial x_2} & \cdots & \frac{\partial e_2}{\partial x_N} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial e_N}{\partial x_1} & \frac{\partial e_N}{\partial x_2} & \cdots & \frac{\partial e_N}{\partial x_N}
\end{bmatrix}.
\] (7.36)
Note that
\[
\frac{\partial e_i}{\partial x_j} \equiv 0 \text{ when } i \neq j, \tag{7.37}
\]
so that \(H\) can be written as
\[
H = \text{diag} \left\{ \frac{\partial e_k}{\partial x_k} \bigg|_{\bar{x}_k} \right\} = \text{diag} \left\{ G_{x,k} \right\}. \tag{7.38}
\]
Then \(G_{x,k}\) can be rewritten as
\[
G_{x,k} \triangleq \left. \frac{\partial e_k}{\partial x_k} \right|_{\bar{x}_k, \rho_{c,k}} = \text{col} \left\{ G_{x,k}^j \right\}, \tag{7.39}
\]
where
\[
G_{x,k}^j \triangleq \left. \frac{\partial e_k}{\partial x_k} \right|_{\bar{x}_k, \rho_{w}} = -\left. \frac{\partial g(x_k, \rho_{w}^j)}{\partial x_k} \right|_{\bar{x}_k, \rho_{w}}. \tag{7.40}
\]

To derive \(G_{x,k}^j\), apply the chain rule:
\[
\left. \frac{\partial g(x_k, \rho_{w}^j)}{\partial x_k} \right|_{\bar{x}_k, \rho_{w}} = \left. \frac{\partial \mathcal{K}}{\partial x_n,k} \right|_{\bar{x}_n,k} \left. \frac{\partial \mathcal{P}}{\partial \rho_{c,k}^j} \right|_{\bar{x}_c,k} \left. \frac{\partial \rho_{c,k}^j}{\partial \rho_{b}^j} \right|_{\bar{x}_b,k} \left. \frac{\partial \rho_{b}^j}{\partial \rho_{w}^j} \right|_{\bar{x}_w,k}. \tag{7.41}
\]
The first term, which is the Jacobian of the camera transform, is
\[
K_{x_n} \triangleq \left. \frac{\partial \mathcal{K}}{\partial x_n,k} \right|_{\bar{x}_n,k} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix}. \tag{7.42}
\]
The second term is the Jacobian of the perspective transform,
\[
P_{x_c} \triangleq \left. \frac{\partial \mathcal{P}}{\partial x_c} \right|_{\bar{x}_c} = \begin{bmatrix} 1/\bar{z} & 0 & -\bar{y}/\bar{z}^2 \\ 0 & 1/\bar{z} & -\bar{x}/\bar{z}^2 \end{bmatrix}, \tag{7.43}
\]
where \(\bar{x}_c = \rho_{c,k}^j\). The last term is more complicated and requires an additional application of chain rule. This Jacobian can be broken down as follows:
\[
\left. \frac{\partial \rho_{c,k}^j}{\partial x_k} \right|_{\bar{x}_k} = \begin{bmatrix} \frac{\partial \rho_{c,k}^j}{\partial \rho_{b}^j_{c,w}} & \frac{\partial \rho_{c,k}^j}{\partial \theta_k} \end{bmatrix}. \tag{7.44}
\]
From (7.29), it can be seen that
\[
\left. \frac{\partial \rho_{c,k}^j}{\partial \rho_{b}^j_{c,w}} \right|_{\bar{x}_b,k} = C_{bc}^T, \tag{7.45}
\]
Algorithm 2: MLE batch estimator used for estimating the wing kinematics of the robotic dragonfly

1. Initialize $\bar{x}, \delta x^*, \alpha$;
2. while $|J(\bar{x} + \alpha\delta x^*) - J(\bar{x})| > \epsilon$ do
   3. for $k=1:N$ do
      4. for Each measurement do
         5. Calculate $G^j_{x,k} = -K_{x,a}P_{xc}C_{bc}^T\begin{bmatrix} 1 & C_{wb,k}^T\rho_w^j \times \bar{S} \end{bmatrix}_{\bar{x}_k, \rho_w^j}$;
         6. Create $G^j_{x,k} = \text{col}\{G^j_{x,k}\}$;
         7. Create $Q_k = \text{diag}\{Q\}$;
         8. Create $H = \text{diag}\{G^j_{x,k}\}$;
         9. Create $T = \text{diag}\{Q_k\}$;
      10. Solve for update $\delta x^*$ using $H^T H^{-1} \delta x^* = -H^T e(\bar{x})$;
      11. if $J(\bar{x} + \delta x^*) > J(\bar{x})$ then
          12. Find $\alpha$ such that $J(\bar{x} + \alpha\delta x^*) < J(\bar{x})$
      13. else
          14. $\alpha = 1$
      15. $\bar{x} \leftarrow \bar{x} + \alpha\delta x^*$
   16. return $\bar{x}$;

and the second term is

$$\frac{\partial \rho^j_{c,k}}{\partial \theta_k} = C_{wb,k}^T \rho_w^j \times \bar{S}.$$  \hfill (7.46)

For a full derivation of (7.46), see Appendix C. Finally, the expression for $G^j_{x,k}$ can be rewritten as

$$G^j_{x,k} = -K_{x,a}P_{xc}C_{bc}^T\begin{bmatrix} 1 & C_{wb,k}^T\rho_w^j \times \bar{S}_k \end{bmatrix}_{\bar{x}_k, \rho_w^j},$$  \hfill (7.47)

where

$$\bar{S}_k = \begin{bmatrix} C_3(\bar{\psi}_k)C_2(\bar{\alpha}_k)1_1 & C_3(\bar{\psi}_k)1_2 & 1_3 \end{bmatrix},$$  \hfill (7.48)

and $C_1, C_2$ and $C_3$ are the principal rotation matrices. A summary of the algorithm can be found in Algorithm 2.
Chapter 8

Results and Discussion

8.1 Simulated Datasets

This estimation algorithm was tested in MATLAB by creating simulated measurement datasets which consist of the pixel coordinates of all the PoIs over a number of wing beats. To create the measurements, the idealized wing kinematics are used to calculate the rotation from the wing to the body frame. These rotations were applied to each of the PoIs to obtain their coordinates in the body frame. The measurement model was then applied to these positions in 3D space to obtain the position of the PoIs in pixel coordinates. For the simulated data, the translation from the origin of the wing frame to the origin of the camera frame as expressed in the body frame, $\rho_{Ow}^{Oc}$, and the rotation of the camera frame with respect to the body frame, $C_{bc}^T$ are assumed to be some values that reflect a real-world experimental setup. Similarly, the parameters of the camera model used for this transformation were obtained by performing a lens calibration routine on a Canon 50 mm f/1.8 lens using Bouguet’s calibration toolbox [54]. Gaussian noise was then added to the pixel coordinates of the PoIs to create the noisy measurements that are used in the estimation algorithm.

8.2 Estimator Simulation Set-Up

To create the simulated datasets that were used to test the estimation algorithm, an idealized sinusoidal wing trajectory with the frequency and amplitudes inspired by the dragonfly *Sympetrum sanguineum* was used:

\[
\begin{align*}
\phi &= \frac{\pi}{4} \cos(\omega t) \\
\alpha &= \frac{\pi}{4} \sin(\omega t) \\
\psi &= 0
\end{align*}
\]
where $\omega = 80\pi \text{ s}^{-1}$ (or 40 Hz). The rotation from the camera frame to the body frame, is represented by the rotation matrix created using the 1-2-3 Euler angle set:

$$C_{bc} = C_{1,2,3} \left( \frac{14\pi}{15}, 0, \frac{\pi}{2} \right),$$

and the translation from the origin of the frame to the origin of the body frame (expressed in the body frame) in metres is

$$\rho_{b O_c O_w} = \begin{bmatrix} -0.2 \\ 0.1 \\ -1.5 \end{bmatrix}.$$

The positions of the PoIs on the wing are provided courtesy of Peter Szabo from his SolidWorks model of the prototype [31]. They are given below in the wing frame, in metres:

$$\rho_w^1 = \begin{bmatrix} 0 & -0.03 & 0 \end{bmatrix}^\top$$
$$\rho_w^2 = \begin{bmatrix} 0.0045 & -0.0283 & 0 \end{bmatrix}^\top$$
$$\rho_w^3 = \begin{bmatrix} 0.0068 & -0.0194 & 0 \end{bmatrix}^\top$$
$$\rho_w^4 = \begin{bmatrix} 0.0062 & -0.0088 & 0 \end{bmatrix}^\top.$$

The threshold used for the line search was $10^{-15}$; this was chosen to be close to the numerical error in MATLAB. Finally, the process noise used in the measurement model was varied in the different simulation runs.

### 8.3 Estimator Results and Discussion

Using simulated datasets, it can be seen that the estimator converges reasonably well depending on the choice of measurement noise, $Q$. The first check to see if the algorithm works is to test it using measurements generated with no noise. When the estimator was run with a measurement noise of 1 pixel, the estimated error converged to zero, showing that the algorithm works. The results can be seen in Figure 8.1 and Figure 8.2.

The measurement noise can then be varied to see the effect on performance. Starting with the measurement noise having a standard deviation of 1 pixel, it can be seen that the performance has shows some large errors, especially in the z-coordinate of $\rho_{b O_c O_w}$, Figure 8.4. The maximum standard deviation of the errors of the kinematics are $12.0^\circ$ for $\phi$, $11.9^\circ$ for $\alpha$ and $3.4^\circ$ for $\psi$. This performance is clearly worse than the other single-camera techniques mentioned in the literature review. However, 1 pixel standard deviation for the measurement
Figure 8.1: Estimated error in the wing kinematics with no noise applied to the measurements.

Figure 8.2: The estimated error in the translation between the camera frame and the body frame with no noise applied to the measurements.
Figure 8.3: Estimated error in the wing kinematics with 1 pixel standard deviation in the measurement noise, plotted with the $\pm 1\sigma$ of the estimated error.

Figure 8.4: Estimated error in the translation from the camera frame to the body frame with 1 pixel standard deviation in the measurement noise, plotted with the $\pm 1\sigma$ of the estimated error.
Figure 8.5: Estimated error in the wing kinematics with 0.5 pixel standard deviation in the measurement noise, plotted with the ±1σ of the estimated error.

noise is likely a very conservative estimate of the quality of the measurements. Most feature detection algorithms return subpixel coordinates for PoIs and so it would reasonable to assume the uncertainty in the pixel coordinates is also subpixel.

If the standard deviation of the measurement noise is reduced to 0.5 pixels, it can be seen the performance improves; see Figure 8.5 and Figure 8.6. Now the maximum standard deviation of the errors of the kinematics are 5.5° for φ, 2.6° for α and 1.6° for ψ.

If the standard deviation of the measurement noise is reduced even further to 0.1 pixels, it can be seen the performance also improves; see Figure 8.7 and Figure 8.8. Now the maximum standard deviation of the errors of the kinematics are 1.0° for φ, 0.9° for α and 0.3° for ψ. From these simulations, it clear that performance can be tuned with the measurement noise. If the measurement noise of the feature detector is small enough, then this algorithm can provide better error estimates than the other single-camera methods presented previously. More important, the error estimates are provided in a rigorous manner, unlike the previous work.

These results also show that certain wing configurations are more susceptible to measurement noise. This can be seen in the plot of the estimated kinematics versus the true kinematics, Figure 8.9. It appears that the largest estimated errors are when the stroke angle and geometric angle of attack are at their largest amplitudes. As the measurement noise decreases, these peak errors also decrease; see Figure 8.10. This issue could be solved by finding more PoIs on the wings that are better distributed.
Figure 8.6: Estimated error in the translation from the camera frame to the body frame with 0.5 pixel standard deviation in the measurement noise, plotted with the $\pm 1\sigma$ of the estimated error.

Figure 8.7: Estimated error in the wing kinematics with 0.1 pixel standard deviation in the measurement noise, plotted with the $\pm 1\sigma$ of the estimated error.
Figure 8.8: Estimated error in the translation from the camera frame to the body frame with 0.1 pixel standard deviation in the measurement noise, plotted with the $\pm 1\sigma$ of the estimated error.

Figure 8.9: The estimated kinematics vs. the true kinematics with 1 pixel standard deviation measurement noise.
Figure 8.10: The estimated kinematics vs. the true kinematics with 0.1 pixel standard deviation measurement noise.
Chapter 9

Future Work

There are a couple further steps necessary to apply the algorithm to real-world data. First, it is necessary to estimate the rotation of the camera frame with respect to the body frame, $C_{bc}^T$ prior to estimating the wing kinematics. It is not possible to add this additional transformation to the estimated states because the system would then be underconstrained. However, it is possible to use images of the still prototype in the experimental setup to estimate $C_{bc}^T$ using similar estimation method as described in Section 7.1. Second, it is also necessary to develop a method to detect the PoIs in the images. This can be as simple as a program that allows the user to manually select PoIs, or as complicated as a program to automatically detect and match the PoIs.

The largest weakness in this estimator setup is that it relies on using PoIs with known relative positions. Because of this requirement, the measurements are limited to features that can be measured and identified by humans, which in turn limits the number of features that can be used. This is especially apparent in some of the spikes in error that occur in certain configurations which are more susceptible to noise in the measurements. To resolve this limitation, more PoIs on the wing could be used. However, this requires a multicamera measurement system that would allow for arbitrary features, whose positions are unknown, to be used. In turn, this allows the use of algorithms to detect features such as Speeded Up Robust Features (SURF) or Scale-Invariant Feature Transform (SIFT). These feature detection algorithms are also more robust to environmental changes.
Chapter 10

Summary

This part of the thesis covered the development of an estimation algorithm, based on a Gauss-Newton Batch Estimator, used to estimate the wing kinematics of the dragonfly MAV prototype. The algorithm is a MLE where the estimated state maximizes the likelihood of the state given the measurements. In this thesis, the state is primarily the wing kinematic angles ($\phi$, $\alpha$ and $\psi$). In addition to the wing kinematics, the translation from the origin of the camera frame to the origin of the wing frame is also estimated since this is a parameter of the system that cannot be precisely measured. For the measurements, the algorithm uses a single camera to determine the pixel coordinates of known PoIs on the wing resulting from the venation structure. The measurement model is the transformation of the positions of the PoIs on the wing to pixel coordinates. Unlike the normal Gauss-Newton Batch Estimator, a motion model was not used. This is because the estimator is primarily intended to be used to verify the dynamical model derived in the first part of this thesis by measuring the wing kinematics, while the existing force sensor measures the forces generated by the prototype. The model that needs to be verified should not be part of the method used for verification.

This algorithm was tested in MATLAB using simulated datasets and showed convergence. However, the performance depends heavily on the choice of measurement noise. In order to achieve better performance than the other single-camera methods presented in literature, the measurement uncertainty needs to be sufficiently small. This can reasonably be assumed as true because most automated feature detection algorithms have subpixel accuracy. However, it can be seen that certain wing configurations are more susceptible to measurement noise, primarily when the stroke and geometric angles of attack are at their maximum amplitudes. In this regard, the algorithm can be improved by using more PoIs on the wing, i.e., more measurements. Acquiring more measurements would necessitate moving to a multicamera setup to allow PoIs that have unknown positions on the wings. Finding these PoIs could be done using feature detection algorithms like SURF or SIFT. This would also allow for the entire process to be automated.

Unlike previous work, this estimation algorithm is based on rigorous probabilistic methods
rather than geometric, deterministic methods. This allows the estimation algorithm to provide better estimates of the uncertainty of the solution. It also is more aligned with the recent multicamera methods used by Walker et al. [46].
Conclusion

The aim of this thesis was twofold. The first was to provide a dynamical model of the dragonfly Micro Aerial Vehicle (MAV)’s actuator-transmission-wing (ATW) system that could be used in conjunction with the design process to allow performance testing for different prototype configurations without actually building them. It would also be used to inform the design process by providing insight into the physical characteristics of the system. In addition, this model should be able to be used as a plant when designing as system controller. The second aim of this thesis was to develop an estimation algorithm that could be used with the prototypes to determine their performance and simultaneously verify the dynamical model of the system. Currently only the forces generated by the prototype can be measured. But to determine if the ATW system is working as designed, the wing kinematics need to be measured as well.

The first part of the thesis presented a dynamical model of the dragonfly MAV’s ATW system, where the equations of motion were derived using the Lagrangian method and the aerodynamic model followed the quasisteady model proposed by Sane and Dickinson [27]. Unlike many of the models presented in literature, the model derived in this work accounts for the internal dynamics of the MAV. These internal dynamics are important to consider for the free-flight of the MAV prototype since the internal components, especially the actuator, compose a large percentage of the overall mass of the prototype.

This dynamical model was simulated in MATLAB using the parameters from prototype 2P15 and the resulting wing kinematics and aerodynamic forces were presented. The simulations varied the actuator driving force amplitude and frequency to determine if there were any combinations that would allow this current prototype to achieve liftoff. From these results, it did not appear that the current prototype would be able to do so.

However, there was some unexpected behaviour in the system at certain driving frequencies. This behaviour requires further investigation to determine the cause and to see if it could be used to generate kinematics and forces for liftoff. Furthermore, to allow the current model to be used as a plant for control design, the degrees of freedom (DoFs) for the MAV body and the hindwing pairs needed to be added. Improvements also need to be made to the aerodynamic model as not all the unsteady effects present in insect flight are accounted for.

The second part of the thesis developed an estimation algorithm used to estimate the wing
kinematics of the dragonfly MAV prototype. Based on a Gauss-Newton Batch Estimator, this algorithm uses a single camera and known points of interest (PoIs) on the wing to estimate these angles ($\phi$, $\alpha$, and $\psi$). Unlike previous work using a single camera, this estimation algorithm is based on rigorous probabilistic methods rather than geometric methods. This allows the estimation algorithm to provide better estimates of the uncertainty of the solution. It is also more similar to the recent multicamera methods being used by researchers.

The algorithm is a maximum likelihood estimator where the estimated state maximizes the likelihood of the state given the measurements. Though it is based on the Gauss-Newton Batch Estimator, it differs because a motion model was not used. This is because the estimator is primarily intended to be used to verify the dynamical model derived in the first part of this thesis by measuring the wing kinematics, and the same model should not be used as part of the verification method.

This estimator was tested in MATLAB using simulated datasets. It can be seen from these test results that in order to achieve better performance than methods in literature, the measurement uncertainty needs to be sufficiently small. Since feature detection algorithms have subpixel accuracy, this is a reasonable expectation. It can also be seen that certain wing configurations are more susceptible to measurement noise. This issue can be resolved by using more measurements, namely more PoIs on the wings. This would necessitate moving to a multicamera setup to allow PoIs that have unknown positions on the wings. Finding these PoIs could be done using feature detection algorithms like Speeded Up Robust Features (SURF) or Scale-Invariant Feature Transform (SIFT). This would also allow for the entire process to be automated.

These two contributions also differ from the work found in literature as they attempt to be more rigorous in their derivations. Even though they are still preliminary work, both are essential steps to the eventual flight of the dragonfly MAV.
References


[33] T. Weis-Fogh, “Biology and physics of locust flight. ii. flight performance of the desert locust (schistocerca gregaria),” *Philosophical Transactions of the Royal Society of*


Appendix A

Complete Equations of Motion

This appendix presents the complete equations of motion for the dynamical model of the dragonfly MAV that was derived in Part I I, Chapter 2. It is an expansion of the terms of Equation 2.54. Recall this equation is

$$M_{eom} \ddot{x}_{eom} = f_{non} (\dot{x}_{eom}, x_{eom}) + f_{other} (\dot{x}_{eom}, x_{eom}, \tau_{app}, \tau_{5,aero}, \tau_{7,aero}),$$

where, $x_{eom} = \begin{bmatrix} \phi_5 & \alpha_5 & \alpha_7 \end{bmatrix}^T$. 
Appendix A. Complete Equations of Motion

\[
M_{\text{com}}(1, 1) = J_1, l_2^2 \left[ \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right]^2 + \frac{1}{4} m_1 l_2^2 \left[ \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right]^2
+ J_2, l_2^2 \left[ \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right] - \frac{l_2}{l_1} \left( \frac{1}{2} \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5)^2 + \frac{l_5}{l_1} \sin \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left[ \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right]^2
\]

\[
+ \frac{1}{2} m_2 \left[ l_1 l_2 \left( -2 \frac{l_2}{l_1} \left( \frac{1}{2} \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2
\]

\[
+ \frac{1}{2} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2
\]

\[
\frac{1}{2} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2
\]

\[
+ J_4, l_2^2 \sin^2 \phi_5 + \frac{1}{2} m_4 \left[ - \frac{1}{2} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \right)^2 \right] \frac{l_2^2}{l_1 l_4} \sin^2 \phi_5
\]

\[
\frac{1}{2} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5)^2 + \frac{l_5}{l_1} \sin \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)
\]

\[
+ \frac{1}{2} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2
\]

\[
+ \frac{1}{2} \left[ \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5)^2 + \frac{l_5}{l_1} \sin \phi_5 \right) \right] \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5)^2 + \frac{l_5}{l_1} \sin \phi_5 \right)^2
\]

\[
+ J_5, l_2^2 \sin^2 \phi_5 + \frac{1}{2} m_5 \left[ \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5^2) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right).
\]

\[
\frac{1}{2} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5)^2 + \frac{l_5}{l_1} \sin \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5)^2 + \frac{l_5}{l_1} \sin \phi_5 \right)^2
\]

\[
+ J_5, l_2^2 \cos^2 \alpha_5 + J_5, l_3^2 \sin^2 \alpha_5 + J_5, l_2^2 \sin^2 \phi_5 + J_5, l_3^2 \cos^2 \alpha_5 + \frac{1}{2} m_6 \left[ \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2
\]

\[
+ \frac{1}{2} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5)^2 + \frac{l_5}{l_1} \sin \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right)^2
\]

\[
J_1 \cos \alpha_5; \quad \text{(A.1)}
\]

\[
M_{\text{com}}(1, 2) = J_5, l_2 \cos \alpha_5; \quad \text{(A.2)}
\]

\[
M_{\text{com}}(1, 3) = -J_5, l_2 \cos \alpha_5; \quad \text{(A.3)}
\]

\[
M_{\text{com}}(2, 1) = J_5, l_2 \cos \alpha_5; \quad \text{(A.4)}
\]

\[
M_{\text{com}}(2, 2) = J_5, l_2 \cos \alpha_5; \quad \text{(A.5)}
\]

\[
M_{\text{com}}(2, 3) = 0 \quad \text{(A.6)}
\]

\[
M_{\text{com}}(3, 1) = -J_5, l_2 \cos \alpha_5; \quad \text{(A.7)}
\]

\[
M_{\text{com}}(3, 2) = 0 \quad \text{(A.8)}
\]

\[
M_{\text{com}}(3, 3) = J_5, l_2 \cos \alpha_5; \quad \text{(A.9)}
\]
\[ f_{\text{non},1} = - (c_{\phi_5, \text{non},1} + c_{\phi_5, \text{non},2} c_{\phi_5, \text{non},3}) \dot{\phi}_5^2 - c_{\phi_5, \text{non}} \ddot{\phi}_5 - c_{\phi_5, \text{non}} \phi_5^2 - c_{\phi_7, \text{non}} \dot{\phi}_7^2 \]  
\( \text{A.10} \)
Appendix A. Complete Equations of Motion

\[ c_{\phi_5, \text{non.}_2} = J_{4,11} \frac{l_5^2}{l_6^2} \sin \phi_5 \cos \phi_5 \]

\[ + \frac{1}{2} l_{14} \left\{ - \frac{1}{2} l_{1}^2 \sin^3 \phi_5 (1 - \cos \phi_5) \left( 1 - \frac{1}{2} \left( \frac{l_5}{l_6} (1 - \cos \phi_5) \right)^2 \right) + \frac{1}{2} l_{1}^2 \left( 1 - \frac{1}{2} \left( \frac{l_5}{l_6} (1 - \cos \phi_5) \right)^2 \right) \right\} \cos \phi_5 \sin \phi_5 \]

\[ + 2 l_{1} \left\{ \frac{l_5^2}{l_{14}^2} \sin^2 \phi_5 + \frac{l_5^2}{l_{14}^2} \left( \frac{l_5}{l_1} \cos \phi_5 - \frac{l_5}{l_1} \sin \phi_5 \right) \right\} \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{14}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ - \left( \frac{l_5^2}{l_{14}^2} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 - \frac{l_5}{l_1} \sin \phi_5 \right) \left( \frac{l_5^2}{l_{14}^2} \left( 1 - \cos \phi_5 \right) \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right\} \right( \frac{l_5^2}{l_{14}^2} (1 - \cos \phi_5) \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \cos \phi_5 \sin \phi_5 \]

\[ c_{\phi_5, \text{non.}_3} = J_{6,11} \frac{l_5^2}{l_6^2} \sin \phi_5 \cos \phi_5 \]

\[ + \frac{1}{2} l_{16} \left\{ - \frac{1}{2} l_{1}^2 \sin^3 \left( 1 - \cos \phi_5 \right) \left( 1 - \frac{1}{2} \left( \frac{l_5}{l_6} (1 - \cos \phi_5) \right)^2 \right) + \frac{1}{2} l_{1}^2 \left( 1 - \frac{1}{2} \left( \frac{l_5}{l_6} (1 - \cos \phi_5) \right)^2 \right) \right\} \cos \phi_5 \sin \phi_5 \]

\[ + 2 l_{1} \left\{ \frac{l_5^2}{l_{16}^2} \sin^2 \phi_5 + \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 - \frac{l_5}{l_1} \sin \phi_5 \right) \right\} \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ - \left( \frac{l_5^2}{l_{16}^2} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 - \frac{l_5}{l_1} \sin \phi_5 \right) \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right\} \right( \frac{l_5^2}{l_{16}^2} (1 - \cos \phi_5) \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \cos \phi_5 \sin \phi_5 \]

\[ \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \sin \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \sin \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \sin \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \sin \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \sin \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \sin \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ \left( \frac{l_5^2}{l_{16}^2} \left( 1 - \cos \phi_5 \right) \sin \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right) \right) \left( 1 - \frac{1}{2} \frac{l_5^2}{l_{16}^2} \left( \frac{l_5}{l_1} \cos \phi_5 + \frac{l_5}{l_1} \sin \phi_5 \right)\right) \cos \phi_5 \sin \phi_5 \]

\[ c_{\alpha, \text{non.}} = - \left( -2 J_{5,11} \cos \alpha_5 \sin \alpha_5 + 2 J_{5,33} \sin \alpha_5 \cos \alpha_5 \right) \dot{\alpha}_5 + \left( -2 J_{7,11} \cos \alpha_7 \sin \alpha_7 + 2 J_{7,33} \sin \alpha_7 \cos \alpha_7 \right) \dot{\alpha}_7 \]

\[ c_{\alpha_5, \text{non.}} = - J_{5,12} \sin \alpha_5 \]
Appendix A. Complete Equations of Motion

\[ c_{\alpha r, \text{non}} = J_{r,12} \sin \alpha \tau \]  
(A.16)

\[ f_{\text{non}, 2} = -(J_{r11} \cos \alpha_5 \sin \alpha_5 - J_{r,33} \sin \alpha_5 \cos \alpha_5) \dot{\phi}_5^2 \]  
(A.17)

\[ f_{\text{non}, 3} = -(J_{r11} \cos \alpha \tau \sin \alpha_5 - J_{r,33} \sin \alpha \tau \cos \alpha \tau) \dot{\phi}_5^2 \]  
(A.18)

\[ f_{\text{other}, 1} = \left[ +k_1 \left( \frac{1}{2} \frac{l_2^2}{l_4} (1 - \cos \phi_5) + \frac{l_5}{l_1} \sin \phi_5 \right) \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \\
+ \frac{k_2}{l_2^2} \left( \frac{1}{2} \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) + \frac{l_5}{l_1} \sin \phi_5 \right)^3 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \\
+ \frac{k_3}{l_2^2} \left( \frac{1}{2} \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) + \frac{l_5}{l_1} \sin \phi_5 \right)^3 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \\
+ \frac{k_4}{l_2^2} (1 - \cos \phi_5) \sin \phi_5 + \frac{k_5}{l_2^2} (1 - \cos \phi_5) \sin \phi_5 + 2k_{p_h} \phi_5 \\
+ \left( \frac{1}{2} m_1 + m_2 + m_3 \right) g l_1 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \\
+ m_4 g \left( l_1 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \right) + \frac{l_2^2}{2 l_4} \frac{l_5}{l_1} \sin \phi_5 (1 - \cos \phi_5) \\
+ m_6 g \left( l_1 \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \right) + \frac{l_2^2}{2 l_6} \frac{l_5}{l_1} \sin \phi_5 (1 - \cos \phi_5) \\
+ m_7 g (\sin \phi_5 \sin \alpha_5 x_{cm} - y_{cm} \cos \phi_5) + m_{17 g} (\sin \phi_5 \sin \alpha \tau x_{cm} - y_{cm} \cos \phi_5) \\
+ \tau_{\text{app}} \left( \frac{l_2^2}{l_1 l_4} (1 - \cos \phi_5) \sin \phi_5 + \frac{l_5}{l_1} \cos \phi_5 \right) \right] \]  
(A.19)

\[ f_{\text{other}, 2} = -(k_\alpha \alpha_5 - m_{15 g} \cos \phi_5 \cos \alpha_5 x_{cm}) \]  
(A.20)

\[ f_{\text{other}, 3} = -(k_\alpha \alpha_\tau - m_{17 g} \cos \phi_7 \cos \alpha \tau x_{cm}) \]  
(A.21)
Appendix B

Complete Simulation Results

This appendix presents the simulation results of the dynamical model of the dragonfly MAV from Part I, Chapter 3. In the simulation runs, the actuator driving-force amplitude, $A$, and the actuator driving-frequency, $f$, were varied to determine the effect they had on the wing kinematics and the generated vertical force. The total energy of the system minus the work was also used to verify the validity of the simulations.

B.1 $A = 1 \text{ mN}, f = 10 \text{ Hz}$

![Graphs showing wing states](image)

Figure B.1: Left wing states
Figure B.2: Right wing states

Figure B.3: Total Energy
Appendix B. Complete Simulation Results

B.2 $A = 1 \text{ mN}, f = 20 \text{ Hz}$

Figure B.4: Aerodynamic Forces

Figure B.5: Left wing states
Appendix B. Complete Simulation Results

Figure B.6: Right wing states

Figure B.7: Total Energy
Figure B.8: Aerodynamic Forces

B.3 $A = 1 \text{ mN}, f = 30 \text{ Hz}$

Figure B.9: Left wing states
Figure B.10: Right wing states

Figure B.11: Total Energy
Appendix B. Complete Simulation Results

B.4 $A = 1 \text{ mN}, f = 40 \text{ Hz}$

Figure B.12: Aerodynamic Forces

Figure B.13: Left wing states
Figure B.14: Right wing states

Figure B.15: Total Energy
Appendix B. Complete Simulation Results

B.5 \( A = 1 \text{ mN}, f = 50 \text{ Hz} \)

Figure B.16: Aerodynamic Forces

Figure B.17: Left wing states
Figure B.18: Right wing states

Figure B.19: Total Energy
Appendix B. Complete Simulation Results

B.6 \( A = 1 \text{ mN}, f = 60 \text{ Hz} \)

Figure B.20: Aerodynamic Forces

Figure B.21: Left wing states
Figure B.22: Right wing states

Figure B.23: Total Energy
Appendix B. Complete Simulation Results

B.7 $A = 1 \text{ mN}, f = 70 \text{ Hz}$

![Aerodynamic Forces in the Inertial Frame](image)

Figure B.24: Aerodynamic Forces

![Left Wing States](image)

Figure B.25: Left wing states
Figure B.26: Right wing states

Figure B.27: Total Energy
B.8  $A = 10$ mN, $f = 10$ Hz

Figure B.28: Aerodynamic Forces

Figure B.29: Left wing states
Figure B.30: Right wing states

Figure B.31: Total Energy
Appendix B. Complete Simulation Results

Figure B.32: Aerodynamic Forces

B.9 $A = 10$ mN, $f = 20$ Hz

Figure B.33: Left wing states
Figure B.34: Right wing states

Figure B.35: Total Energy
B.10 $A = 10 \text{ mN}, f = 30 \text{ Hz}$

Figure B.36: Aerodynamic Forces

Figure B.37: Left wing states
Figure B.38: Right wing states

Figure B.39: Total Energy
Appendix B. Complete Simulation Results

Aerodynamic Forces in the Inertial Frame

Figure B.40: Aerodynamic Forces

B.11 $A = 10 \text{ mN}, f = 40 \text{ Hz}$

Figure B.41: Left wing states
Appendix B. Complete Simulation Results

Figure B.42: Right wing states

Figure B.43: Total Energy
Appendix B. Complete Simulation Results

B.12 \( A = 10 \, \text{mN}, \ f = 50 \, \text{Hz} \)

Figure B.44: Aerodynamic Forces

Figure B.45: Left wing states
Figure B.46: Right wing states

Figure B.47: Total Energy
Appendix B. Complete Simulation Results

B.13 $A = 10$ mN, $f = 60$ Hz

Figure B.48: Aerodynamic Forces

Figure B.49: Left wing states
Appendix B. Complete Simulation Results

Figure B.50: Right wing states

Figure B.51: Total Energy
Appendix B. Complete Simulation Results

Figure B.52: Aerodynamic Forces

B.14 \( A = 10 \text{ mN}, f = 70 \text{ Hz} \)

Figure B.53: Left wing states
Appendix B. Complete Simulation Results

Figure B.54: Right wing states

Figure B.55: Total Energy
Aerodynamic Forces in the Inertial Frame

Figure B.56: Aerodynamic Forces
Appendix C

Derivation of the Jacobian of $C^T v$

This appendix shows the derivation for the Jacobian $\frac{\partial C^T v}{\partial \theta}$ which is used in the derivation of the Jacobian of the error terms in Part II, Chapter 7. Using an Euler angle formulation, every rotation matrix can be written as the product of principal-axis rotation matrices as follows:

\[ C = C(\theta_1, \theta_2, \theta_3) = C_\gamma(\theta_3) C_\beta(\theta_2) C_\alpha(\theta_1), \]
\[ C^T = C^T(\theta_1, \theta_2, \theta_3) = C^T_\alpha(\theta_1) C^T_\beta(\theta_2) C^T_\gamma(\theta_3). \]

The Jacobian can be expressed as

\[ \frac{\partial C^T v}{\partial \theta} = \begin{bmatrix} \frac{\partial C^T v}{\partial \theta_1} & \frac{\partial C^T v}{\partial \theta_2} & \frac{\partial C^T v}{\partial \theta_3} \end{bmatrix}. \]

To simplify the resulting expressions, we will use the following identities:

\[ (C x)^x = C x^x C^T, \]
\[ \frac{\partial C_\alpha(\theta)}{\partial \theta} = -1_\alpha^x C_\alpha(\theta), \]
\[ \frac{\partial C^T_\alpha(\theta)}{\partial \theta} = \left( \frac{\partial C_\alpha(\theta)}{\partial \theta} \right)^T. \]

Derivations for these identities C.5 and C.7 can be found in the following appendices. Identity C.6 is from the course notes for the graduate course on state estimation offered at the University of Toronto Institute for Aerospace Studies [53].
Now, we can derive each of the terms in the Jacobian:

\[
\frac{\partial \mathbf{C}^\top \mathbf{v}}{\partial \theta_1} = \frac{\partial}{\partial \theta_1} \left( \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \mathbf{v} \right)
\]

Using equations C.6 and C.7:

\[
= \mathbf{C}_\alpha^\top(\theta_1) \mathbf{1}_\alpha^\top \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \mathbf{v}
\]

\[
= \mathbf{C}_\alpha^\top(\theta_1) \left[ \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \mathbf{C}_\gamma^\top(\theta_3) \mathbf{C}_\beta^\top(\theta_2) \right] \mathbf{1}_\alpha^\top \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \mathbf{v}
\]

\[
= \left[ \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \right] \left[ \mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{1}_\alpha^\top \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \right] \mathbf{v}
\]

Using equation C.5:

\[
= \left[ \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \right] \left[ \mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{1}_\alpha \right]^\top \mathbf{v}
\]

\[
= -\mathbf{C}^\top \mathbf{v}^\times \left[ \mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{1}_\alpha \right]. \quad (C.8)
\]

Similarly, the second term can be derived as

\[
\frac{\partial \mathbf{C}^\top \mathbf{v}}{\partial \theta_2} = \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \mathbf{1}_\beta^\top \mathbf{C}_\gamma^\top(\theta_3) \mathbf{v}
\]

\[
= \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \left[ \mathbf{C}_\gamma^\top(\theta_3) \mathbf{C}_\gamma^\top(\theta_3) \mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \right] \mathbf{1}_\beta^\top \mathbf{C}_\gamma^\top(\theta_3) \mathbf{v}
\]

\[
= \left[ \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \right] \left[ \mathbf{C}_\gamma(\theta_3) \mathbf{1}_\beta^\top \mathbf{C}_\gamma^\top(\theta_3) \mathbf{v} \right]
\]

\[
= \left[ \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \right] \left[ \mathbf{C}_\gamma(\theta_3) \mathbf{1}_\beta \right]^\top \mathbf{v}
\]

\[
= -\mathbf{C}^\top \mathbf{v}^\times \left[ \mathbf{C}_\gamma(\theta_3) \mathbf{1}_\beta \right]. \quad (C.9)
\]

The last term is then

\[
\therefore \frac{\partial \mathbf{C}^\top \mathbf{v}}{\partial \theta_3} = \mathbf{C}_\alpha^\top(\theta_1) \mathbf{C}_\beta^\top(\theta_2) \mathbf{C}_\gamma^\top(\theta_3) \mathbf{1}_\gamma^\top \mathbf{v}
\]

\[
= -\mathbf{C}^\top \mathbf{v}^\times \mathbf{1}_\gamma. \quad (C.10)
\]

Collecting all the terms and rearranging

\[
\frac{\partial \mathbf{C}^\top \mathbf{v}}{\partial \theta} = \begin{bmatrix}
\frac{\partial \mathbf{C}^\top \mathbf{v}}{\partial \theta_1} & \frac{\partial \mathbf{C}^\top \mathbf{v}}{\partial \theta_2} & \frac{\partial \mathbf{C}^\top \mathbf{v}}{\partial \theta_3}
\end{bmatrix}
\]

\[
= -\mathbf{C}^\top \mathbf{v}^\times \begin{bmatrix}
\mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{1}_\alpha & \mathbf{C}_\gamma(\theta_3) \mathbf{1}_\beta & \mathbf{1}_\gamma
\end{bmatrix}
\]

\[
= -\mathbf{C}^\top \mathbf{v}^\times \mathbf{S}(\theta_2 \theta_3), \quad (C.11)
\]

where

\[
\mathbf{S}(\theta_2 \theta_3) = \begin{bmatrix}
\mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{1}_\alpha & \mathbf{C}_\gamma(\theta_3) \mathbf{1}_\beta & \mathbf{1}_\gamma
\end{bmatrix}. \quad (C.12)
\]

Note this is the same matrix used in the kinematic relation

\[
\mathbf{S}(\theta_2 \theta_3) \dot{\mathbf{\theta}} = \mathbf{\omega}.
\]
The matrix $\bar{S}$ is simply

$$
\bar{S} = S(\theta_2, \theta_3)|_{\bar{\theta}}, \text{ where } \bar{\theta} \triangleq \begin{bmatrix} \theta_1 \\ \bar{\theta}_2 \\ \theta_3 \end{bmatrix}.
$$

(C.13)
Appendix D

Derivation of Equation \((C.5)\)

We start with the generally known identity

\[
(Ax) \times (Ay) \equiv (\det A) A^{-T} (x \times y).
\]  \hfill (D.1)

Take the left hand side of equation \(C.5\) and multiply by \(Cu\)

\[
(Cv^x C^T) (Cu) = Cv^x u
\]

Since \(\det C = 1\) and \(C^T = C^{-1}\)

\[
= \det CC^{-T} v^x u
\]

\[
= (Cv)^x (Cu)
\]

\[
\Rightarrow Cv^x C^T = (Cv)^x
\]
Appendix E

Derivation of Equation (C.7)

From identity C.6, we know
\[
\frac{\partial C_\alpha(\theta)}{\partial \theta} \equiv -1_\alpha^x C_\alpha(\theta).
\] (E.1)

Therefore,
\[
\frac{\partial C_\alpha(\theta)}{\partial \theta}^\top = C_\alpha(\theta)^\top 1_\alpha^x.
\] (E.2)

Using Rodrigues’ formula we can rewrite the rotation matrix as
\[
C_{\alpha}(\theta) = \cos \theta 1 + (1 - \cos \theta) 1_\alpha 1_\alpha^\top 1_\alpha^x - \sin \theta 1_\alpha^x.
\] (E.3)

Taking the transpose
\[
C_\alpha^\top(\theta) = \cos \theta 1 + (1 - \cos \theta) 1_\alpha^\top 1_\alpha + \sin \theta 1_\alpha^x.
\] (E.4)

Now, taking the derivative
\[
\frac{\partial C_\alpha^\top}{\partial \theta} = -\sin \theta 1 + \sin \theta 1_\alpha^\top 1_\alpha + \cos \theta 1_\alpha^x
\]
\[
= \sin \theta (-1 + 1_\alpha^\top 1_\alpha) + \cos \theta 1_\alpha^x
\]
\[
= \sin \theta 1_\alpha^x 1_\alpha^\top 1_\alpha + \cos \theta 1_\alpha^x
\]
\[
= \sin \theta 1_\alpha^x 1_\alpha^\top 1_\alpha + (1 - \cos \theta) 1_\alpha^\top (1_\alpha^x 1_\alpha) + \cos \theta 1_\alpha^x
\]
\[
= (\cos \theta 1 + (1 - \cos \theta) 1_\alpha^\top 1_\alpha + \sin \theta 1_\alpha^x) 1_\alpha^x
\]
\[
= C_\alpha^\top(\theta) 1_\alpha^x
\]
\[
= \left(\frac{\partial C_\alpha}{\partial \theta}\right)^\top
\] (E.5)