UNIVERSITY OF CALIFORNIA,
IRVINE

Battery Lifetime-Aware Flight Control for Flapping Wing Micro Air Vehicles

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Computer Engineering

by

Xiaohong Li

Thesis Committee:
Associate Professor Mohammad Abdullah Al Faruque, Chair
Assistant Professor Haithem Taha
Assistant Professor Marco Levorato

2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank Professor Al Faruque, my advisor, for granting me the precious opportunity to study at the Advanced Integrated Cyber-Physical Systems (AICPS) Lab of University of California, Irvine. Also, I am grateful to my advisor for introducing me to this interesting FWMAV project.

I would also like to thank Professor Taha for teaching me knowledge of the FWMAV dynamics, periodic shooting methods, and so on. Thank you for giving me so many great advice on the experiments.

And thank Professor Al Faruque, Professor Taha and Professor Levorato for kindly being my thesis committee members and reviewing my thesis report. With their support and supervision, I managed to complete this thesis.

A special thanks goes out to Korosh, my colleague during the research on FWMAV project. Together we designed the control algorithm and he provided me many great suggestions on my thesis. Also thank you for teaching me knowledge of control methodology and providing me suggestions on revising my thesis report.

I would like to thank Sujit, Anthony, Sina, Jiang, Salam, Nafiul, Hsin-Wei for your support and making an enjoyable time at the lab. My thanks is also sent to Mohammadali for your support and for sharing your knowledge of FWMAV model.

At last, I am grateful to my family and friends for their support. Thank you for always encouraging and supporting me!

ABSTRACT OF THE THESIS

Battery Lifetime-Aware Flight Control for Flapping Wing Micro Air Vehicles

By

Xiaohong Li

Master of Science in Computer Engineering

University of California, Irvine, 2018

Associate Professor Mohammad Abdullah Al Faruque, Chair

The balance and stability analysis for FWMAV systems is always quite challenging, while the flight duration is also becoming a big challenge. In this thesis, we analyzed the balance and stability of FWMAV system while considering the battery model for best energy usage. We designed two stages for controlling the FWMAV vertical flight in this thesis: static flight and dynamic flight. The static flight stage focuses on the tradeoff between the stability and the energy usage, as well as the effection of the battery model. With the battery model considered, static flight orbits with lower power consumption are found. For better optimization of the power consumption, the dynamic flight control methods are proposed by relaxing the strict constraints of hovering and periodic flight. Compared with the power consumption results in the static flight stage, the power consumption improvement in the dynamic flight stage is even larger than 40% while keeping orbits periodic and hovering.

Chapter 1

Introduction

Flight dynamics of biological insects and birds and their mimetic vehicles, Micro Air Vehicles (MAVs), has been an active area of research in different communities for more than a decade since DARPA started a program named MAV-project in 1997 [24]. MAVs are micro sized aircraft which spans a very wide variety of applications in a multitude of commercial, industrial and military purposes, like observation and search and rescue missions. The main applications are intelligence, surveillance, and reconnaissance missions. They can provide a rapid overview in the area around personnel, without exposing themselves to danger [23]. To perform these missions MAVs should be small sized, have good maneuverability, be well controllable.

MAVs come in various types, like conventional fixed wing aircraft and rotary aircraft. A third type of MAV, flapping-wing micro-air-vehicles (FWMAVs), has attractive characteristics for flight in confined spaces. FWMAVs are different from conventional fixed wing aircraft as they use their flapping wings both for a means of propulsion (thrust) and for a means to sustain flight (lift). FWMAVs have the potential to combine the positive aspects of both fixed wing and rotary flights, while eliminating many of their disadvantages. They are able

to travel at higher velocities than the rotary platforms while being more maneuverable than the fixed wing MAVs [25]. The low RPM of the motors driving the wings make FWMAVs be much quieter than other platforms. FWMAVs may have the ability to hover like rotary aircraft, while they lack the high speed rotating blades that may be dangerous and are easily damaged.

Most research groups working on MAVs today based their design on FWMAVs. The agility of hummingbirds, dragonflies, bees, and fruit flies has inspired scientist to study how they use flapping wings as a mean to generate aerodynamic forces capable of producing often complex maneuvers in air [30].

Many different platforms have been constructed and modeled. Harvard's RoboBees are robotic insects with sub-millimeter scale anatomy and two wafer-thin wings which flaps at 120 times per second, achieving vertical takeoff, hovering and steering [16, 3]. The DelFly FWMAVs from the Delft University of Technology make a series of FWMAVs [18]. The second generation DelFly II was already capable of hovering, flying forward as well as flying backward. It can keep flying for 15 minutes [15]. It was then improved and became the DelFly Explorer, which is capable of sustained flight [4]. The smallest one of the series, named DelFly Micro, weighs only 3.07 grams. It has a wingspan of 10 cm, and can fly for up to 3 minutes with a camera onboard [7]. The Robo Raven from the University of Maryland was the first demonstration of a bird-inspired platform doing outdoor aerobatics using independently actuated and controlled wings. It successfully performed dives, flips, and button hook turns, demonstrating the capability of bio-inspired aerobatic maneuvers afforded by the design [10].

1.1 Motivation

The balance and stability analysis for FWMAV systems is always quite challenging. Many research papers have been published for this topic. However, as FWMAVs are becoming an important tool in a wide variety of defense and civilian applications nowadays [23], the flight duration also becomes a big challenge. Most platforms use the lithium-ion batteries as the power supply. Hence, in this thesis, we analyze the balance and stability of FWMAV system while leverage the knowledge of the battery dynamics for best energy usage.

Several efforts have been attempted to determine the optimum flapping strategy and many different controllers also have been designed for FWMAVs [20, 39, 8, 40, 27, 19]. However, all of these efforts have ignored the interplay between flapping and battery dynamics. To the best of our knowledge, this is the first CPS optimization framework for FWMAVs.

The approach for analysis of the balance and stability is adopted from work [13]. In this approach, it first applies a combined first- and higher-order averaging technique on the FWMAV system. This averaging technique is applied to determine the required amplitude of the periodic forcing for balance. Then, it uses an optimized shooting technique to numerically capture the resulting periodic orbit for hovering. The purpose of the periodic orbit shooting technique is to find a starting point on the periodic orbit that minimize residual, which is a term used to represent the stability of hovering.

Based on the work in [13], we take the power consumption into account and preform the optimization for a trade-off between the stability and the energy usage, as well as the solution for best energy usage.

1.2 Outline

In chapter 2, the FWMAV model and the particular wing configuration of DelFlyII used in this thesis are introduced. The periodic shooting method adopted in this thesis for system stability analysis is also described in chapter 2. The battery model is described in chapter 3. The battery model involves using an effective pseudo-current. The increase of discharging rate will introduce a larger increase in the effective pseudo-current, which results in a lower usable capacity [31, 34, 36]. In chapter 4, the flight control problems and algorithms for balancing the stability of FWMAV system and the energy usage are presented. The experimental results, as well as the analysis of the results are represented in chapter 5. These lead to the conclusion and further investigation in chapter 6.

Chapter 2

Flapping Wing Dynamics

2.1 FWMAV Dynamic Model

Flapping flight dynamics is appropriately represented by a multibody, nonlinear, time-varying system. In [29], the authors deduced the full equations of motion governing the longitudinal flapping flight dynamics.

In general, three types of reference frames are required to formulate the flight dynamic model for a rigid-winged FWMAV: an inertial reference frame x_I, y_I, z_I, a body-fixed reference frame x_b, y_b, z_b, and a wing-fixed reference frame x_w, y_w, z_w for each of the flight vehicle's wings. Because only longitudinal flight is considered in this work, only a single wing-fixed reference frame is required because the opposing wing moves symmetrically.

There are five degrees of freedom (DOF) dynamics in total: the body variables including body pitch angle θ, body velocity components u and w along the x_b and z_b directions; the wing variables including the back and forth flapping angle as well as the wing pitching angle η (maintained constant throughout each half-stroke). Figure 2.1 gives out the schematic

diagram for a FWMAV with a horizontal stroke plane.

As for now we only consider to achieve hovering, the body velocity components w along the z_b directions is the dynamic we care about most. It is convenient to neglect the body pitch angle θ, body velocity components u in x_b direction and the wing pitching angle η as in work [12] and [28]. We constraint the body to move along vertical rails, hence there are only two degrees of freedom: the body vertical motion with a velocity w and the wing back and forth flapping angle φ.

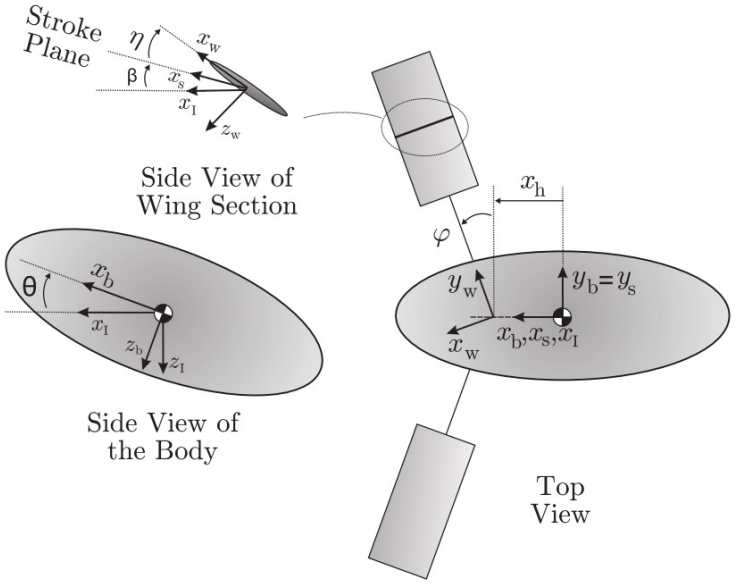

Figure 2.1: Schematic diagram for a FWMAV with a horizontal stroke plane [29]

In this model, there are two degrees of freedom: the body vertical motion with a velocity w and the wing back and forth flapping angle φ. The model is represented by equation (2.1).

$$\begin{aligned} \dot{w}(t) &= g - k_{d1}|\dot{\varphi}(t)|w(t) - k_L\dot{\varphi}^2(t) \\ \ddot{\varphi}(t) &= -k_{d2}|\dot{\varphi}(t)|\dot{\varphi}(t) - k_{d3}w(t)\dot{\varphi}(t) + \frac{\tau_\varphi(t)}{I_F} \end{aligned} \tag{2.1}$$

Where I_F is the flapping moment of inertia, and τ_φ is the flapping control input torque which is written as equation (2.2).

$$\tau_\varphi(t) = U cos\omega t \tag{2.2}$$

k_{d1}, k_L, k_{d2}, k_{d3} are defined as equations (2.3).

$$\begin{aligned} k_{d1} &= \frac{\rho C_{L\alpha} I_{11} cos^2\alpha_m}{2m_v} \\ k_L &= \frac{\rho C_{L\alpha} I_{21} sin\alpha_m cos\alpha_m}{2m_v} \\ k_{d2} &= \frac{\rho C_{L\alpha} I_{31} sin^2\alpha_m}{I_F} \\ k_{d3} &= \frac{\rho C_{L\alpha} I_{21} sin\alpha_m cos\alpha_m}{I_F} \end{aligned} \tag{2.3}$$

where ρ is the air density, which is normally 1.225. $C_{L\alpha}$ is the wing lift curve slope, α_m is the mean angle of attack maintained throughout the entire stroke, m_v is the total mass of the vehicle, and I_{mn} are constraints that depend on the chord distribution of the wing: I_{mn} $= 2\int_0^R r^m C^n(r)dr$.

The system (2.1) can be written in a state-space form as (2.4)

$$\frac{d}{dt}\begin{bmatrix} z(t) \\ \varphi(t) \\ w(t) \\ \dot{\varphi}(t) \end{bmatrix} = \begin{bmatrix} w(t) \\ \dot{\varphi}(t) \\ g \text{ - } k_{d1}|\dot{\varphi}(t)|w(t) \text{ - } k_L\dot{\varphi}^2(t) \\ \text{- } k_{d2}|\dot{\varphi}(t)|\dot{\varphi}(t) \text{ - } k_{d3}\, w(t)\, \dot{\varphi}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{I_F} \end{bmatrix} \tau_\varphi(t) \tag{2.4}$$

which can be written as a typical nonlinear control-affine system as (2.5)

$$\dot{x}(t) = Z(x(t)) + Y(x(t))\tau_\varphi(t) \tag{2.5}$$

$$x(t) = [z(t) \quad \varphi(t) \quad w(t) \quad \dot{\varphi}(t)]^T \tag{2.6}$$

where *x(t)* in equation (2.6) is the state vector. The *z(t)* is the vehicle displacement during the flight, which can be used to calculate the vehicle altitude using equation (2.7)

$$A(t) = A_0 - z(t) \tag{2.7}$$

where *A(t)* is the altitude and A_0 is the initial altitude of the FWMAV.

In this work, the morphological and the wing planform are given in Table 2.1. The values are adopted from the Delfly II as its capability of hovering satisfying our requirements. Its wing planform is shown in figure 2.2 [5].

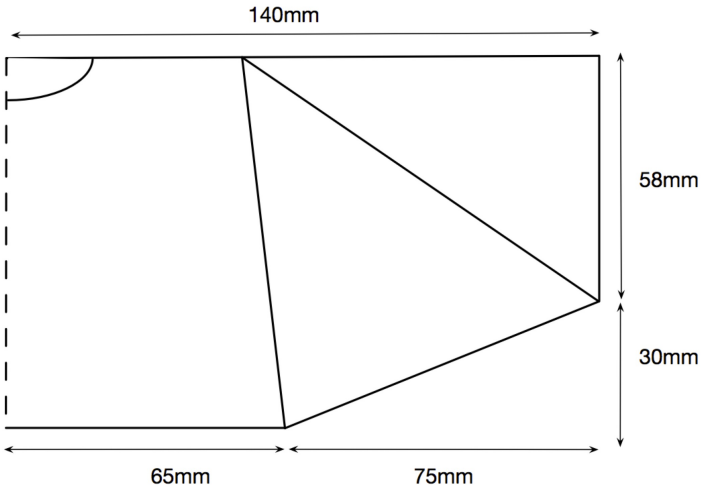

Figure 2.2: Schematic diagram for the wing planform of DelFly II

Parameters	Value
R (mm)	*140*
S (mm^2)	*10177.5*
m_w *(gm)*	*1.1*
m_v *(gm)*	*14*

Table 2.1: Values of parameters for DelFly II

In Table 2.1, R is the semi-span of the wing, S is the area of one wing, m_w is the single wing mass and m_v is the total vehicle mass, including the battery mass.

The moments of the wing chord distribution are defined as equation (2.8)

$$I_{k1} = 2\int_0^R r^k c(r)dr \tag{2.8}$$

where r represents the distance between a point on the wing axis and the wing base. The chord distribution for the insect is defined as equation (2.9):

$$c(r) = \begin{cases} 0.088, & 0 \leqslant r \leqslant 0.065 \\ 0.088 - 0.03 * (r\text{-}0.065)/0.075, & 0.065 < r \leqslant 0.14 \end{cases} \tag{2.9}$$

The mass of the wing is assumed uniform with an areal mass distribution $m^{'}$:

$$m^{'} = \frac{m_w}{S} \tag{2.10}$$

The inertial properties of the wing are then estimated as

$$\begin{aligned} I_x &= 2\int_0^R m^{'} r^2 c(r)dr \\ I_y &= 2\int_0^R m^{'} \hat{d}^2 c^3(r)dr \\ I_z &= I_x + I_y \end{aligned} \tag{2.11}$$

where $\hat{d}$ is the chord-normalized distance from the wing hinge line to the center of gravity line. Here, the hinge is at the top of the wing and we assume the center of gravity lies at the 30% of chord station. Then $\hat{d}$ is given by:

$$\hat{d} = 0.3 * c(0) \tag{2.12}$$

The flapping moment of inertia is calculated by:

$$I_F = I_x sin^2\alpha_m + I_z cos^2\alpha_m \tag{2.13}$$

$C_{L\alpha}$ is the wing lift curve slope of the three dimensional wing, which is given by:

$$C_{L\alpha} = \frac{\pi AR}{1 + \sqrt{(\frac{\pi AR}{a_0})^2 + 1}} \tag{2.14}$$

where AR is the wing aspect ratio. In this work, we consider the wing as two dimensional wing, a_0 is then assumed as 2π. AR is calculated by

$$AR = \frac{R^2}{S} \tag{2.15}$$

where R is the semi-span of the wing, S is the area of one wing as mentioned above.

The power consumption for the FWMAV to hover is given by:

$$P_{mav}(t) = \tau_\varphi(t)\dot{\varphi}(t) \tag{2.16}$$

where $\tau_\varphi(t)$ is the flapping control input torque, $\dot{\varphi}(t)$ is the derivative of the flapping angle which indicates the flapping direction - forward or backward.

2.2 Periodic Shooting Methodology

In [13], the authors summarized the procedure for analyzing the balance and stability of FWMAVs as in Figure. 2.3. The VOC and higher-order averaging are used to determine the required amplitude U for balance. In this thesis, as U is used as one of the control inputs

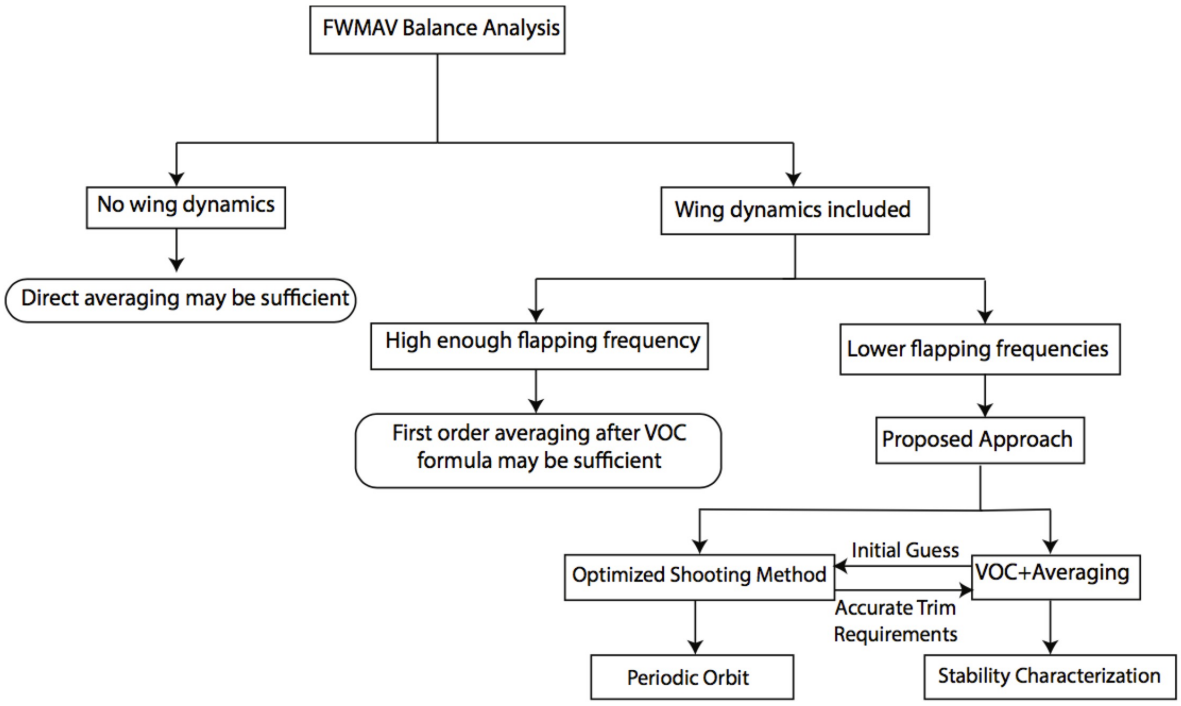

Figure 2.3: Schematic for the analysis procedure of balance problem [13]

and is optimized as a independant variable. Hence, we only adopted the periodic shooting method from the work [13].

The optimized shooting method adopted in [13] came from the work in [6]. Consider the following system of equations (2.17)

$$\dot{x}(t) = f(x(t), \alpha, t) \tag{2.17}$$

where $x \in \mathbb{R}^n$ and f: $\mathbb{R}^n \times \mathbb{R}^k \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$, and α are the system parameters. A solution $x(t)$ to the system (2.17) is periodic if there exists a constant $T > 0$ such that

$$x(t) = x(t + T) \tag{2.18}$$

The optimized shooting method can be applied to any system that can be expressed in the

form of (2.17). A dimensionless time τ is introduced such that $t = \tau\ T$. Equation (2.17) is then written as

$$\frac{dx}{d\tau} = Tf(x(\tau T), \alpha, \tau T) \tag{2.19}$$

And the boundary conditions in Eq. (2.18) then can be represented as $x(\tau = 0) = x(\tau = 1)$ and Eq.(2.19) can be integrated over one period. Now, the residual can be written as

$$R = T \int_0^1 f(x(\tau T), \alpha, \tau T) d\tau \tag{2.20}$$

According to work in [6], the residual depends on the number of quantities to be optimized and can be expressed as

$$\begin{aligned} R = &(x(1) - x(0), x(1 + \Delta\tau) - x(\Delta\tau), ..., \\ &x(1 + (p-1)\Delta\tau) - x((p-1)\Delta\tau)) \end{aligned} \tag{2.21}$$

where $\Delta\tau$ is the integration step size and p is a natural number required by the Levenberg-Marquardt algorithm [9], which is adopt in work [13]. It has to be chosen so that the number of components of the residual is greater than or equal to the number of quantities to be optimized. The number of components in the residual function is given by pN, where N is the dimension of the system.

Chapter 3

Battery Model

Energy and power are very crucial parts of FWMAV design. The flight endurance time of the FWMAV heavily depends on the available energy, while power determines whether it is possible the flight could happen in the first place. There are three characteristics, which are flight time, maneuverability and payload capacity, are essential to the utility of the design of FWMAV [5]. They often have to be traded off against each other. In this section, first the energy sources for FWMAV is discussed. Then, the battery modelling used in this thesis is described.

3.1 Energy Storage for FWMAV

The efficiency of FWMAV flight is not as efficient as other larger air vehicles, as the air viscosity would have greater influence on the air flow for vehicles with low velocity or small size. It is also related to the flight mode of the FWMAV. For example, the hovering flight is a quite power intensive flight mode. Hovering flight needs more power and hence shows shorter flight times than forward flight. Hummingbirds need to feed every 4 to 5min to stay

airborne when they hover [11]. The way how the FWMAV flies is also an very important factor affecting the flight efficiency. The flapping wing flight could partly overcome the loss of lift [5].

Hence, when choosing the energy sources for FWMAV, the specific energy and specific power of the energy storage material are the factors to consider. They are measured by the power-to-weight (kW/kg) and power density (kW/liter). The FWMAVs needs electric power for the flight control and payload systems and mechanical power for propulsion. A conversion system is needed to convert the energy from the source to propulsive and electric power. This conversion system can be efficient and simple for some energy sources. It also can be completely impractical for others. For instance, hydrocarbon fuel engines, with internal combustion or jet turbines, are widely used in larger unmanned aircraft. While existing versions of these systems are too large to be accommodated for micro or nano aircrafts. There are UAVs flying on prototype Lithium-Sulphur cells which are made by the SionPower company. The cells are tested in military. The Li-S cell is a little worse in power-to-weight ratio but 50% better than the Lithium Polymer in energy-to-weight ratio at this stage of development. But the cells are not yet available commercially. They could be useful in the future when the pace of development proceeds.

For small to very small aircrafts, the electric power source which is in the form of a battery powering an electric motor yields a very interesting compromise between complexity and duration. The modern Lithium Polymer batteries have good power density, energy-to-weight ratio and power-to-weight ratio.

3.2 Battery Model

Lithium-ion batteries demonstrate less usable capacity in higher discharge rates (rate-capacity effect). This characteristic is described by the Peukert's Law, which is presented by the German scientist Wilhelm Peukert in 1897. Peukert's Law is represented by equation (3.1) and was developed to model the behavior of the battery for different discharge currents [14].

$$C = I^{pc}t \tag{3.1}$$

where C is the battery capacity expressed in Ah, I is the discharging current, t is the discharging time in hours and pc is the calibration coefficient. When a battery is subjected to discharge using increasingly higher constant currents, the discharged battery capacity decreases up to the cutoff voltage [14]. To compensate for these losses, Peukert introduced the Peukert's constant k.

The value of the Peukert's constant is normally determined experimentally using two different battery discharge curves. Consider that the battery is discharged in rate I_1 for time t_1, and in rate I_2 for time t_2. By applying equation (3.1), we can get below equation (3.2) for calculating the Peukert's constant.

$$pc = \frac{ln(\frac{t_2}{t_1})}{ln(\frac{I_1}{I_2})} \tag{3.2}$$

In this thesis, we adopted the DelFly II characteristics for FWMAV modelling. The DelFly team uses Lithium Polymer batteries from several sources, like the 130 and 220 mAh from Atomic workshop and the 180 mAh NanoTech from HobbyKing [5]. The 130mAh battery from Atomic workshop has discharge curves shown as Figure 3.1.

The relationship between the usable capacity of the battery and the discharge rage is ex-

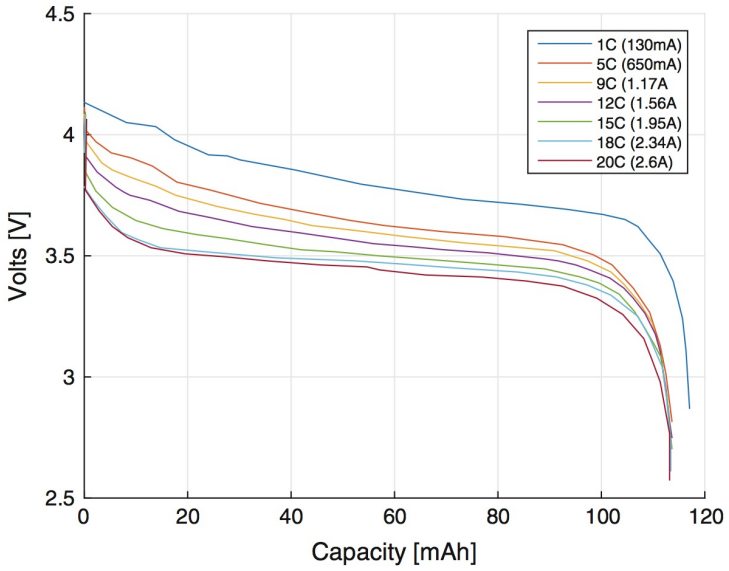

Figure 3.1: Discharge curves of CyclonE-130mAh cell [2]

pressed in Equation 3.3. From this equation, it can be seen that the efficiency of converting chemical energy to electrical energy decreases while increasing the discharge rate. Hence, more chemical reactions are needed in order to provide the same amount of electrical energy [38, 35]. When the discharging current is variable, the model involves using an effective pseudo-current. The increase of discharging rate will introduce a larger increase in the effective pseudo-current, which results in a lower usable capacity [31]. This relationship is expressed by Equation 3.4.

$$C = C_n(\frac{I_n}{I})^{pc-1} \tag{3.3}$$

$$I_{eff} = I(\frac{I}{I_n})^{pc-1} \tag{3.4}$$

In this work, the simplified battery model is adopted by assuming that the voltage and the resistance are not changed along with the discharging.

Power, which is the amount of energy used per second, can be calculated by Equation 3.5. As

the voltage V is assumed not changing during discharging in this work, based on Equation 3.4 and 3.5, the relationship between the actual power P_{act} and effective pseudo-power P_{eff} can be deduced by Equation 3.6.

$$P = VI \tag{3.5}$$

$$\begin{aligned} P_{eff} &= V_n I_{eff} \\ &= V_n I_{act} (\frac{I_{act}}{I_n})^{pc-1} \\ &= V_n \frac{P_{act}}{V_n} (\frac{P_{act}}{V_n})^{pc-1} \\ &= \frac{P_{act}^{pc}}{(V_n I_n)^{pc-1}} \end{aligned} \tag{3.6}$$

The battery specification used in this work is shown as Table 3.1. In order to see the effect of considering battery model in the optimization, we set the value of *pc* as *1.2* which is not as small as the *pc* value of the 130mAh battery from Atomic workshop.

Parameters	Value
V_n *(V)*	*3.7*
C_n *(mAh)*	*130*
pc	*1.2*

Table 3.1: Battery specification

Chapter 4

FWMAV Flight Control

The balance and stability analysis for FWMAV systems is always quite challenging, while the flight duration is also becoming a big challenge. In this thesis, we analyze the balance and stability of FWMAV system while considering the battery model for best energy usage. The periodic orbit, hovering status of the flight and power consumption are the three main aspects we consider when we design the control algorithm. We make sure the orbit is periodic by controlling the residual, which is described in chapter 2. The smaller the value of residual is, the better periodic states the orbit has. The hovering status is also measured by the displacement *z(t)* part inside the residual. Small displacement means the FWMAV is hovering.

This thesis covers two stages of the FWMAV project: static flight and dynamic flight. In our algorithm, the entire flight is composed of flight patterns. The idea is inspired by the control algorithm in work [37, 33, 1, 32], which is based on Model Predictive Control(MPC) method. During the static vertical flight, the values of control inputs stay the same during the flight pattern, hence also stay the same during the entire flight. The optimizer will optimize the control inputs for one pattern. During the dynamic flight, different with the static flight,

the pattern is divided into several parts and each part has a separate set of control inputs. The optimizer will optimize the control inputs for all parts in a pattern together. In this stage, instead of pursuing the hovering flight, the residual equation is adjusted to allow the FWMAV have more flexible flying strategy.

In all of our experiments for both of the static flight and dynamic flight stages, the flight pattern is set as 4 flapping cycles. Each flapping cycle is divided into 8 time segments. For each time segment, the values of state vector are calculated based on the model in equation (2.4) by using *ode45*.

4.1 Flight Optimization Problem

The flight control is responsible for adjusting the control inputs to the FWMAV system for maintaining the system output and state variables in a specific range and target. With the purpose of achieving the balance of stability while minimize the power consumption, we includes the residual and the power consumption into the cost function. The stability is represented by the residual, which is introduced by the periodic shooting method in chapter 2. It is used to ensure the hovering flight. For easier evaluation, we use the sum of the squares of each element in the residual, instead of residual itself, in our cost function. And the energy usage is measured by the average power consumption of the entire flight. The cost function of the optimization in general is represented by equation (4.1). It will be adjusted accordingly in experiments for different stages, which will be described in the following sections.

In equation (4.1), χ is the control input vector, P represents power consumption and R represents the residual. $\mathfrak{w}_1$ and $\mathfrak{w}_2$ are weights for the term of power consumption and residual in the cost function, respectively. P_{max} is the maximum power limitation depending on battery characteristics, w is the body vertical velocity and w_{max} is the maximum body

vertical velocity. The value of w_{max} can be adjusted as needed. This constraint is used to keep the FWMAV away from a large vertical movement during the flight. φ is the flapping angle which is limited in the range of $[-\pi/2, \pi/2]$.

$$\begin{aligned} &\underset{\chi}{min} \quad \mathfrak{w}_1\overline{P} + \mathfrak{w}_2 \sum R^2 \\ &s.t. \\ &\quad \begin{cases} P(t) \leqslant P_{max} \\ |w| \leqslant w_{max} \\ |\varphi| \leqslant \pi/2 \end{cases} \end{aligned} \tag{4.1}$$

4.2 Flight Control Algorithm

The flight control algorithm is designed for estimating all the variables contributing to the system. The optimization solver optimizes all of the variables in order to minimize a cost function while considering the constraints put by the control limits and physical behaviors. Then the optimized control inputs are applied to the FWMAV system and form the reproducible flight pattern.

The FWMAV system is a system with a high sensitivity with respect to the control inputs. Hence, we use the accurate model in continuous time to calculate the sampled system state for each time segment. The number of time segments, which is 32 in our experiments, is decided according to the flapping period while considering the trade-off between the accuracy and the running time of the optimization.

Algorithm 1: FWMAV flight control

1 N_{seg} = number of time segments in a pattern

2 $N_{pattern}$ = number of patterns in the entire flight

3 S_{sv} = size of the state vector

4 N_{ctl} = number of the control inputs

5 $x = 1 \times S_{sv}$ matrix `// state vector`

6 $x_{seg} = 1 \times S_{sv}\ N_{seg}$ matrix `// state vector for all time segments`

7 $\chi = 1 \times N_{ctl}$ matrix `// control inputs`

8 $v \leftarrow [\chi,\ x_{seg}]$ `// optimization variables`

9 χ^0 = random initial guess of control inputs

`// calculate the initial values of sampled state vector of each segment`

10 **for** $k = 1$ *to* N_{seg} **do**

11 $\quad {x_{seg}}^0 \leftarrow FWMAV(\chi^0,\ k)$

12 **end**

13 $v^0 \leftarrow [\chi^0,\ {x_{seg}}^0]$ `// initial values of optimization variables`

`// call optimization solver, the solver will call FWMAV and battery model to calculate power and residual`

14 v_{opt} = Optimize($v,\ v^0$, FWMAV, Battery)

15 **for** $j = 1$ *to* $N_{pattern}$ **do**

16 $\quad Orbit$ = FWMAV(χ_{opt}) `// Reproduce the pattern`

17 **end**

18 **return** *Orbit*

Algorithm 1 illustrates a pseudo-code to simulate the controlling process. First, the number of the time segments and the number of the patterns in the entire flight are defined in lines 1 - 2. Then, the state variables ($x,\ x_{seg}$), control inputs (χ) are defined (lines 3 - 5) and combined in a vector as optimization variables (v) (lines 6). The initial value of the control inputs (χ^0)

are randomly generated in a given range (lines 7) and the initial value of the state variables are calculated using the accurate FWMAV model (lines 8 - 10). They formed the initial values of the optimization variables (lines 11). Input the initial values to the optimizer, the optimization problem is solved and the optimizer returns the optimum solution (line 12). When the optimizer solves the problem, it calls the FWMAV model and the battery model to calculate the new system state and the cost function. As the entire flight is composed of reproducible flight patterns, the optimizer will optimize the control inputs for only one pattern. The optimum solution is then applied to the FWMAV model to get the pattern of the flying orbit. By reproducing the pattern, we get the entire flying orbit (line 14 - 15). More details of the static and dynamic flight control implementation and experiments are described in next sections.

4.3 Static Vertical Flight Control

The FWMAV system uses multiple state variables, which are represented by the system state vector (2.6) to define the current status of the system. In this vector, *z(t)* is the displacement of the flight, $\varphi(t)$ is the flapping angle, *w(t)* is the vertical velocity and $\dot{\varphi}(t)$ is the derivative of flapping angle. $\dot{\varphi}(t)$ represents the flapping speed.

In our algorithm, the entire flight is composed of flight patterns. During the static vertical flight, the values of control inputs stay the same during the flight pattern, hence also stay the same during the entire flight. In this stage, the control inputs of the FWMAV system are represented by the vector (4.2). In the control input vector, the part *[φ(0) w(0) $\dot{\varphi}$(0)]* represents the starting state (at time = 0) of the pattern as well as the entire flight, which are inherited from the work [13]. They are chosen as part of the control inputs for the purpose of applying periodic shooting methodology. Different starting state of the flight

may lead the optimizer to different solution.

$$\chi_{static} = [U \quad \alpha_m \quad \omega \quad \varphi(0) \quad w(0) \quad \dot{\varphi}(0)]^T \tag{4.2}$$

The part $[U \quad \alpha_m \quad \omega]$ is used to control the flight process, where U is the amplitude of the periodic force $\tau_\varphi(t)$ shown in equation (2.2), α_m is the angle of attack of the wings, and ω is the flapping frequency of the FWMAV. These three variables are added into the control input vector to control the flying process while achieving more flexibility for the optimizer.

In our algorithm, the optimization variables are composed of state vector and control inputs. The boundaries of the optimization variables used by the optimizer are shown in table 4.2. The initial values of the control inputs are random values generated within the boundaries listed in table 4.1.

Parameters	Lower Bound	Upper Bound
$\varphi(0)$ (rad)	-0.5π	0.5π
$w(0)$ (m/s)	-0.2	0.2
$\dot{\varphi}(0)$ (rad/s)	-50π	50π
U	0.0001	2
α_m (deg)	10	60
ω (rad/s)	16π	100π

Table 4.1: Bounds for generating random initial values of control inputs in static flight stage

Parameters	Lower Bound	Upper Bound
z (m)	-Inf	Inf
φ (rad)	-0.5π	0.5π
w (m/s)	-0.2	0.2
$\dot{\varphi}$ (rad/s)	-Inf	Inf
U	0	Inf
α_m (deg)	10	80
ω (rad/s)	16π	100π

Table 4.2: Bounds for optimization variables of static flight

The flight pattern is set as 4 flapping cycles and each flapping cycle is divided into 8 time segments. The optimization variables are composed of control inputs (4.2) and state vector

(2.6) of all time segments as in equation (4.3). In equation (4.3), we have *j = 1,2,3 ... , 31, 32*. Hence, there are *134* optimization variables in total. To get the initial values of the optimization variables for the optimizer, we first generate random values of the control inputs, then calculate the values of state vector for all time segments by using the values of randomly generated initial values of the control inputs.

$$v_{static} = [U \quad \alpha_m \quad \omega \quad \varphi(0) \quad w(0) \quad \dot{\varphi}(0) \quad z(j) \quad \varphi(j) \quad w(j) \quad \dot{\varphi}(j)]^T \tag{4.3}$$

The experiments during this static flight stage mainly focus on analyzing the effective of considering battery model during the optimization. Hence, we use different cost functions when we conduct the contrast experiments.

4.3.1 Static Flight without Considering Battery Model

Experiment 1: Static Flight without Considering Battery Model When the battery model is not considered, the power consumption we used to calculate the cost function is the actual power consumption of the FWMAV. In this case, the cost function is represented by equation (4.4), where R is the residual and $\overline{P_{act}}$ is the average of actual power consumption of the FWMAV in this experiment of static flight stage. $\sum R^2$ is calculated as in equation (4.5). $\mathfrak{w}_R$ and $\mathfrak{w}_{Pact}$ are weights for $\sum R^2$ and $\overline{P_{act}}$, respectively.

$$Cost_1 = \mathfrak{w}_{Pact}\overline{P_{act}} + \mathfrak{w}_R \sum R^2 \tag{4.4}$$

The R^2 is a $1 \times S_{sv}$ matrix, where S_{sv} is the size of state vector *x(t)* in equation (2.6). For easier evaluation, we use the sum of all elements inside R^2 ($\sum R^2$) in the cost function.

$$\sum R^2 = \sum_{j=1}^{S_{sv}} R^2(j) \tag{4.5}$$

The R^2 is calculated using the value of state vector at each time segment as shown in equation (4.6). $\mathfrak{w}_x$ is the weight vector for each element inside the state vector *x(t)* in equation (2.6). The value of $\mathfrak{w}_x$ does not affect the trade off between the residual and power, as long as it keeps same during the entire experiment. It is set as *[10 1 1 10]* in our experiments. N_{seg} is the total number of time segments in a pattern, N_{cyc} is the total number of flapping cycles in a pattern, and N_{spp} is the number time segments in a flapping cycle. They have the relationship in equation (4.7). $x(i{\times}T_{seg})$ is the value of state vector at time $i{\times}T_{seg}$ (starting time of the flight is 0), where T_{seg} is the time endurance of one single time segment. As in equation (4.8), we get the value of T_{seg} by dividing flapping period by the number of time segments in a flapping cycle.

$$R^2 = \sum_{i=1}^{N_{seg}-N_{spp}} \mathfrak{w}_x[x((i+N_{spp}) \times T_{\text{seg}}) - x(i \times T_{\text{seg}})]^2 \tag{4.6}$$

$$N_{spp} = \frac{N_{seg}}{N_{cyc}} \tag{4.7}$$

$$T_{seg} = \frac{2\pi}{\omega N_{spp}} \tag{4.8}$$

In the static flight stage, the constants used in the equations are listed in table 4.3.

Parameters	Value
S_{sv}	*4*
N_{seg}	*32*
N_{cyc}	*4*
$\mathfrak{w}_x$	*[10 1 1 10]*
z(0) (m)	*0*

Table 4.3: Constants in static flight stage

4.3.2 Static Flight with Considering Battery Model

Experiment 2: Static Flight with Considering Battery Model When the battery model is considered, instead of using actual power consumption, we use effective power consumption of the FWMAV to calculate the cost function. In this case, the cost function is represented by equation (4.9), where R is the residual and $\overline{P_{eff}}$ is the average of effective power consumption of the FWMAV in this experiment of static flight stage.

$$Cost_2 = \mathfrak{w}_{Peff}\overline{P_{eff}} + \mathfrak{w}_R \sum R^2 \tag{4.9}$$

The effective power consumption $\overline{P_{eff}}$ is calculated according to equation (3.6). The $\sum R^2$ is calculated in the same method with the one in experiment 1, as in (4.5). $\mathfrak{w}_R$ and $\mathfrak{w}_{Peff}$ are weights for $\sum R^2$ and $\overline{P_{eff}}$, respectively. The constants used in this experiment are also same with experiment 1, as in table 4.3.

4.4 Dynamic Vertical Flight Control

The static vertical flight doesn't have much flexibility because of the two strict constraints: the orbit is periodic and the FWMAV can hover. Hence, in dynamic flight stage, we relaxed these two constraints to allow the FWMAV to fly in more flexible vertical orbits and expect the optimizer to find dynamic flight orbits with lower power consumption than static flight orbits. The entire flight is also composed of flight patterns. The flapping cycles inside the pattern are not necessarily periodic and at hovering status. But same with the static flight, the pattern still should be reproducible.

In the dynamic flight stage, as the flapping cycles inside the pattern are not necessarily periodic and at hovering status, we introduce more control inputs to control each flapping

cycle in the pattern separately. The control inputs of the FWMAV system are represented by the vector (4.10). In the control input vector, same with the static flight, the part *[φ(0) w(0) $\dot{\varphi}$(0)]* represents the starting state (at time = 0) of the pattern as well as the entire flight. Different starting state of the flight may lead the optimizer to different solution. The pattern includes 4 cycles in our experiments. The variables *[U_i α_{mi} ω_i] (i = 1,2,3,4)* are used to control the flight process of each cycle in the pattern. U_i is the amplitude of the periodic force $\tau_\varphi(t)$ shown in equation (2.2), α_{mi} is the angle of attack of the wings, and ω_i is the flapping frequency of the FWMAV for the i_{th} *(i = 1,2,3,4)* flapping cycle in the pattern, respectively.

$$\chi_{dynamic} = [U_i \quad \alpha_{mi} \quad \omega_i \quad \varphi(0) \quad w(0) \quad \dot{\varphi}(0)]^T \tag{4.10}$$

In our algorithm, the flight pattern is set as 4 flapping cycles and each flapping cycle is divided into 8 time segments. The optimization variables are composed of control inputs (4.10) and state vector (2.6) of all time segments as in equation (4.11). In equation (4.11), we have *i = 1,2,3,4* and *j = 1,2,3 ... , 31, 32*. Hence, there are *143* optimization variables in total. To get the initial values of the optimization variables for the optimizer, we first generate random values of the control inputs, then calculate the values of state vector for all time segments by using the values of randomly generated initial values of the control inputs.

$$v_{dynamic} = [U_i \quad \alpha_{mi} \quad \omega_i \quad \varphi(0) \quad w(0) \quad \dot{\varphi}(0) \quad z(j) \quad \varphi(j) \quad w(j) \quad \dot{\varphi}(j)]^T \tag{4.11}$$

The boundaries of the optimization variables used by the optimizer are shown in table 4.5. The initial values of the control inputs are random values generated within the boundaries listed in table 4.4. In both tables, we have *i = 1,2,3,4* and *j = 1,2,3 ... , 31, 32*.

Compared with the static flight stage, the boundary of the vertical velocity is enlarged from *[-0.2, 0.2]m/s* to *[-0.5, 0.5]m/s*. The boundary of the vertical velocity is enlarged for two

reasons. First, it gives a looser constraint to the vertical velocity. In dynamic flight, we allow a larger fluctuation of the displacement of the FWMAV, hence a looser constraint of the vertical velocity. It allows the optimizer to search more area and get a better solution.

Second, it makes the optimization more practical in the perspective of the running time of optimization. When we generate the initial values of the optimization variables, all *143* variables have to satisfy the lower bound and upper bound listed in table (4.5). If we keep using the *[-0.2, 0.2]m/s* as the bounds of vertical velocity, according to our experiments, the optimizer would take an extremely long time to generate the initial values of the optimization variables which don't violate the bounds in table (4.5). Increasing the boundary of the vertical velocity shortens this optimization running time a lot so that the experiments could be done in a reasonable time.

Parameters	Lower Bound	Upper Bound
$\varphi(0)$ *(rad)*	*-0.5π*	*0.5π*
$w(0)$ *(m/s)*	*-0.5*	*0.5*
$\dot{\varphi}(0)$ *(rad/s)*	*-50π*	*50π*
U_i	*0.0001*	*2*
α_{mi} *(deg)*	*10*	*60*
ω_i *(rad/s)*	*16π*	*100π*

Table 4.4: Bounds for generating random initial values of control inputs in dynamic flight stage

Parameters	Lower Bound	Upper Bound
z_j *(m)*	*-Inf*	*Inf*
φ_j *(rad)*	*-0.5π*	*0.5π*
w_j *(m/s)*	*-0.5*	*0.5*
$\dot{\varphi}_j$ *(rad/s)*	*-Inf*	*Inf*
U_i	*0*	*Inf*
α_{mi} *(deg)*	*10*	*80*
ω_i *(rad/s)*	*16π*	*100π*

Table 4.5: Bounds for optimization variables of dynamic flight

The experiments during this dynamic flight stage mainly focus on finding more flexible dynamic flight orbits which will introduce lower power consumption than static flight orbits.

We improve our experiments by adjusting cost functions and constraints.

4.4.1 Dynamic Flight with Flexible Displacement

Experiment 3: Dynamic Flight without Considering Battery Model We started the experiments by relaxing the constraint for hovering. The hovering status is measured by the displacement z. In this experiment, we allow a larger fluctuation of the displacement of the FWMAV. For this purpose, we divided the displacement z from other variables in the state vector when we calculate the residual. We define the state vector without displacement z as $x^{'}(t)$ in equation (4.12).

$$x^{'}(t) = [\varphi(t) \quad w(t) \quad \dot{\varphi}(t)]^T \tag{4.12}$$

When the battery model is not considered, the cost function of the optimization is adjusted to include three terms as in equation (4.13), where R' is the residual, $\overline{P_{act}}$ is the average of actual power consumption of the FWMAV, and δz is the change of the displacement. $\mathfrak{w}_{R'}$, $\mathfrak{w}_{Pact}$ and $\mathfrak{w}_z$ are weights for $\sum R'^2$, $\overline{P_{act}}$ and z, respectively.

$$Cost_3 = \mathfrak{w}_{Pact}\overline{P_{act}} + \mathfrak{w}_{R'}\sum R'^2 + \mathfrak{w}_z \delta z^2 \tag{4.13}$$

The R'^2 is a $1 \times S'_{sv}$ matrix, where S'_{sv} is the size of the state vector without displacement $x'(t)$ in equation (2.6). For easier evaluation, we use the sum of all elements inside the vector R'^2 ($\sum R'^2$) in the cost function. $\sum R'^2$ is calculated as in equation (4.14).

$$\sum R'^2 = \sum_{j=1}^{S'_{sv}} R'^2(j) \tag{4.14}$$

The R'^2 is calculated using the value of $x'(t)$ at each time segment as shown in equation

(4.15). $\mathfrak{w}_{x'}$ is the weight vector for each element inside *x'(t)* in equation (4.12). The value of $\mathfrak{w}'_x$ does not affect the trade off between the residual and power, as long as it keeps same during the entire experiment. It is set as *[1 1 10]* in our experiments. N_{seg} is the total number of time segments in a pattern, N_{cyc} is the total number of flapping cycles in a pattern, and N_{spp} is the number time segments in a flapping cycle. They have the relationship in equation (4.7). *x'(i*$\times T_{seg}$*)* is the value of *x'(t)* at time $i \times T_{seg}$ (starting time of the flight is 0), where T_{seg} is the time endurance of one single time segment as shown in equation (4.8).

$$R'^2 = \sum_{i=1}^{N_{seg}-N_{spp}} \mathfrak{w}_{x'}[x'((i+N_{spp}) \times T_{\text{seg}}) - x'(i \times T_{\text{seg}})]^2 \tag{4.15}$$

The change of displacement δz is calculated by equation (4.16), where *z(*$N_{seg} \times T_{seg}$*)* is the displacement at time $N_{seg} \times T_{seg}$ and *z(0)* is the initial value (time = 0) of the displacement.

$$\delta z = z(N_{seg} \times T_{\text{seg}}) - z(0) \tag{4.16}$$

In the dynamic flight stage, the constants used in the equations are listed in table 4.6.

Parameters	Value
S'_{sv}	*3*
N_{seg}	*32*
N_{cyc}	*4*
$\mathfrak{w}_{x'}$	*[1 1 10]*
z(0) (m)	*0*

Table 4.6: Constants in dynamic flight stage

Experiment 4: Dynamic Flight with Considering Battery Model By replacing the actual power consumption in equation (4.13) with the effective power consumption, we did the experiments to get more accurate solutions by considering the battery model. The cost function is adjusted and represented by equation (4.17), where P_{eff} is the average of the effective power consumption, which is recalculated by equation (3.6). The R'^2 and δz are

still calculated by equations (4.14) and (4.16), respectively.

$$Cost_4 = \varpi_{Peff}\overline{P_{eff}} + \varpi_{R'}\sum R'^2 + \varpi_z \delta z^2 \tag{4.17}$$

The effective power consumption $\overline{P_{eff}}$ is calculated according to equation (3.6). $\varpi_{R'}$ and ϖ_{Peff} are weights for $\sum R'^2$ and $\overline{P_{eff}}$, respectively. The R'^2 is calculated in the same method with the one in experiment 3, as in (4.14). The δz are still calculated by equation (4.16). The constants used in this experiment are also same with experiment 3, as in table 4.6.

4.4.2 Dynamic Flight with Flexible States

Experiment 5: Dynamic Flight without Considering Battery Model In this experiment, we give an even looser constraint to the flight. In experiment 3 and 4, we only do not require the displacement z be periodic for each cycle in the pattern. But other variables in the state vector (2.6) are still required to be periodic for each cycle. Here, in experiment 5, for all variables in the state vector (2.6), we don't require them be periodic for each cycle, but only for the pattern (4 cycles). In other words, the equation of the sum of the square of residual is reformed to equation (4.19). The cost function is now represented by equation (4.18). In the equations, $\overline{P_{act}}$ is the average of actual power consumption of the FWMAV. ϖ_{Rpat} and ϖ_{Pact} are weights for $\sum {R_{pattern}}^2$ and $\overline{P_{act}}$ respectively.

$$Cost_5 = \varpi_{Pact}\overline{P_{act}} + \varpi_{Rpat}\sum R^2_{pattern} \tag{4.18}$$

$$\sum R^2_{pattern} = \delta z^2 + \delta\phi^2 + \delta w^2 + \delta\dot{\phi}^2 \tag{4.19}$$

The δz is the change of the displacement z, which is calculated by equation (4.16). The $\delta\phi$, δw and $\delta\dot{\phi}$ are calculated by equations (4.20), (4.21) and (4.22), respectively. N_{seg} $\times$

T_{seg} is the ending time of the pattern. The constants used in this experiment are same with experiment 3, as in table 4.6.

$$\delta\phi = \phi(N_{seg} \times T_{\text{seg}}) - \phi(0) \tag{4.20}$$

$$\delta w = w(N_{seg} \times T_{\text{seg}}) - w(0) \tag{4.21}$$

$$\delta\dot{\phi} = \dot{\phi}(N_{seg} \times T_{\text{seg}}) - \dot{\phi}(0) \tag{4.22}$$

Experiment 6: Dynamic Flight with Considering Battery Model By replacing the actual power consumption in equation (4.18) with the effective power consumption, we did the experiments to get more accurate solutions by considering the battery model. The cost function is adjusted and represented by equation (4.23), where P_6 is the average of the effective power consumption, which is recalculated by equation (3.6). The $\sum {R_{pattern}}^2$ and z, ϕ, w, $\dot{\phi}$ are still calculated by equations (4.19) and (4.16), (4.20), (4.21), (4.22) respectively.

$$Cost_6 = \mathfrak{w}_{Peff}\overline{P_{eff}} + \mathfrak{w}_{Rpat}\sum {R_{pattern}}^2 \tag{4.23}$$

4.4.3 Dynamic Flight with Residual as a Constraint

Experiment 7: Dynamic Flight without Considering Battery Model As mentioned in previous sections, the pattern, which includes 4 cycles and 8 time segments for each cycle, should be reproducible so that this pattern can be applied to a long time vertical flight. For making the pattern reproducible, the smaller the sum of the square of residual of the pattern in equation (4.19) is, the better.

With previous methodology, a reproducible orbit is not easy to achieve by adjusting the weights of terms in the cost function. A more strait-forward method is to restrain the residual of the pattern as a nonlinear equality constraint. In other words, we make the last state of the pattern equal to the initial state in the nonlinear constraint file. Meanwhile, we modify the cost function to only contain the power consumption. When the battery model is not considered, the cost function is represented by equation (4.24).

$$Cost_7 = \overline{P_{act}} \tag{4.24}$$

To be noted, the bounds for flapping angle ϕ is also changed from *[-0.5π, 0.5π]* to a smaller range *[-0.4π, 0.4π]*. The optimizer only uses part of the data of state vector ($x(i \times T_{seg})(i=0,1,2...,31,32)$) instead of the complete data of state vector calculated by *ode45*. When we plot the orbit by using the complete data of the state vector, we may find that the flapping angle exceeds the boundaries *[-0.5π, 0.5π]*. And *[-0.5π, 0.5π]* are hard boundaries which should not be broken physically. This problem can be avoid by using smaller bounds for the flapping angle. This is also applicable to all previous experiments if the flapping angle also exceeds the boundaries *[-0.5π, 0.5π]*.

Experiment 8: Dynamic Flight with Considering Battery Model When the battery model is considered, the cost function is represented by equation (4.25). As the only term in the cost function is the power consumption, there should be no much difference of using actual power consumption and using effective power consumption.

$$Cost_8 = \overline{P_{eff}} \tag{4.25}$$

Chapter 5

Experimental Results

The flight dynamics of FWMAVs are represented by multi-body, Nonlinear Time-Periodic (NLTP) system models. In order to evaluate the contribution of including the battery model into the optimization, we implemented two FWMAV system models. The first one only includes the FWMAV mechanical dynamics. The second one also considers battery characteristics except for the FWMAV mechanical dynamics.

Through experiments, we found that the optimum solutions found by the optimizer greatly depend on the initial guess of starting point (state vector at time zero) of the flying orbit. If we give the optimizer a certain initial guess of starting point, the optimizer would always give out the periodic orbit which is close to the initial guess and very hard to find other periodic orbits. In other words, it is stuck in the local minimum. In this case, the optimizer only sees one periodic orbit because of the certain initial guess. The power consumption is then also settled down once the orbit is found. Hence, there is no flexibility at all for optimizing the power consumption while capturing a periodic orbit in this case. In the experiments for dynamic flight, although we allow the FWMAV to have more flexible flying strategy, the periodic orbit close to the initial guess of the starting point is still the best solution it can

find. Hence, in all of the experiments described in the following sections, we find the global minimum solution by the means of generating 100 appropriate random initial guesses of the starting point.

Also, the sampling rate (or, the number of time segments) is also adjusted to balance the accuracy of the model and the optimization running time. With a larger sampling rate, the model is more accurate, but the optimizer would need a longer time to find the solution. In our experiments, we set the flight time for the optimization as 4 flapping cycles and each flapping cycle uses 8 samples.

5.1 Experimental Setup

The FWMAV model equations defined in chapter 2 contain multiple parameters which are defined by the specifications of DelFlyII, which is designed by the Micro Air Vehicles Laboratory of Delft University of Technology. The morphological and the wing planform are shown in figure 2.2. The specification of DelFlyII used in this thesis are listed in table 2.1. The model of FWMAV and battery represented in chapter 2 and chapter 3, as well as our FWMAV flight control described in chapter 4 are implemented in MATLAB [17]. The flight dynamics of FWMAVs are represented by multi-body, NLTP system model, which is written in a state-space form as in equation (2.4). It is solved by the MATLAB *ode45* solver. The optimizations are preformed by the MATLAB *fmincon* solver. The average running time of the *fmincon* solver is around 3 hours per each optimization in our experiments.

5.2 Static Vertical Flight

During the static vertical flight, the control inputs, represented by the equation (4.2), stay the same during the flight pattern.

In our experiments, the flight pattern is set as 4 cycles. Each cycle is divided into 8 time segments. For each time segment, the state variables in equation (2.6) are calculated based on the model in equation (2.4) by using *ode45*. For experiments of static flight stage, the initial values of the control inputs are random values generated within the boundaries listed in table 4.1. The boundaries of the optimization variables of the optimizer are shown in table 4.2.

5.2.1 Static Flight Orbits Analysis

Experiment 1 - Static Flight without Considering Battery Model In this section, we presents the experimental results of experiment 1 described in chapter 4. The controller sets the control variables in the FWMAV for optimizing the cost function and reaching a target. When the battery model is not considered, the cost function of the optimization is composed of the two parts as in equation (4.4).

In order to find the global optimal solution, We run the optimizations *100* times with different random initial values of the control input variables, then choose the final solution out of the 100 sets of solutions which introduce minimal $Cost_1$. With different settings of the weights, we get different solutions as shown in table 5.1, as well as the values of random initial control inputs in table 5.2 which introduce the best solution out of *100* simulations.

Except the conditions that one of the $\mathfrak{w}_R$ and $\mathfrak{w}_{Pact}$ is zero, when we change the weights, the optimizer gives results which are very close to each other even when the values of random

initial variables vary a lot. This is because the static flight itself doesn't have much flexibility when we optimize the power consumption while capturing a static hovering periodic orbit. These two strict conditions, the orbit is periodic and the FWMAV can hover, make the poor flexibility of the optimization, hence the very close solutions.

$\mathfrak{w}_R$	$\mathfrak{w}_{Pact}$	U	α_m *(rad)*	ω *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	*0.0788*	*0.5831*	*93.4523*	*-0.5158*	*0.0408*	*21.3174*
0	*1*	*0.0000*	*0.4008*	*312.8364*	*1.5281*	*-0.0233*	*-69.6945*
100	*1*	*0.0494*	*0.3671*	*50.4013*	*-1.3595*	*0.0743*	*23.1387*
10	*1*	*0.0498*	*0.3461*	*50.2682*	*-1.2187*	*0.0752*	*21.8462*
1	*1*	*0.0496*	*0.3325*	*50.2655*	*-1.4513*	*0.1195*	*20.8708*
1	*10*	*0.0485*	*0.3296*	*50.2655*	*-1.2146*	*0.1990*	*20.5106*
1	*100*	*0.0485*	*0.3299*	*50.2655*	*-1.2166*	*0.1939*	*20.4793*

Table 5.1: Experiment 1 - Control inputs under different weights

$\mathfrak{w}_R$	$\mathfrak{w}_{Pact}$	U	α_m *(rad)*	ω *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	*0.0721*	*0.5663*	*88.8209*	*-0.3271*	*-0.1217*	*16.9859*
0	*1*	*0.1564*	*0.8960*	*261.7321*	*1.4324*	*-0.1573*	*-38.3165*
100	*1*	*0.1079*	*0.4620*	*137.6293*	*0.8453*	*-0.0818*	*-50.3165*
10	*1*	*0.1035*	*0.6120*	*120.2830*	*0.0256*	*-0.0795*	*-36.0361*
1	*1*	*0.0731*	*0.3470*	*81.5081*	*-1.0087*	*0.0149*	*9.4111*
1	*10*	*0.1307*	*0.5560*	*175.3444*	*-0.3793*	*-0.1151*	*17.8239*
1	*100*	*0.1503*	*0.4273*	*168.5789*	*-0.1530*	*0.1684*	*6.7778*

Table 5.2: Experiment 1 - Random initial control inputs under different weights

The average of actual power consumption $\overline{P_{act}}$, the average of effective power consumption $\overline{P_{eff}}$ and the sum of the square of residual $\sum R^2$ are calculated using the results in table 5.1 are presented in the table 5.3. Although the solutions are very close to each other, the tradeoff between the two parts in the cost function can still be seen. Figure 5.1 shows the tradeoff of the results more clearly. With smaller $\mathfrak{w}_{Pact}$, we get larger $\overline{P_{act}}$ and smaller $\sum R^2$.

The smaller $\sum R^2$ means a more periodic pattern of the flight orbit. When $\mathfrak{w}_{Pact}$ is given *0*, we get a very small residual value which gives a very nice periodic hovering orbit and a very small average of vertical velocity (*0.0004985 m/s*).

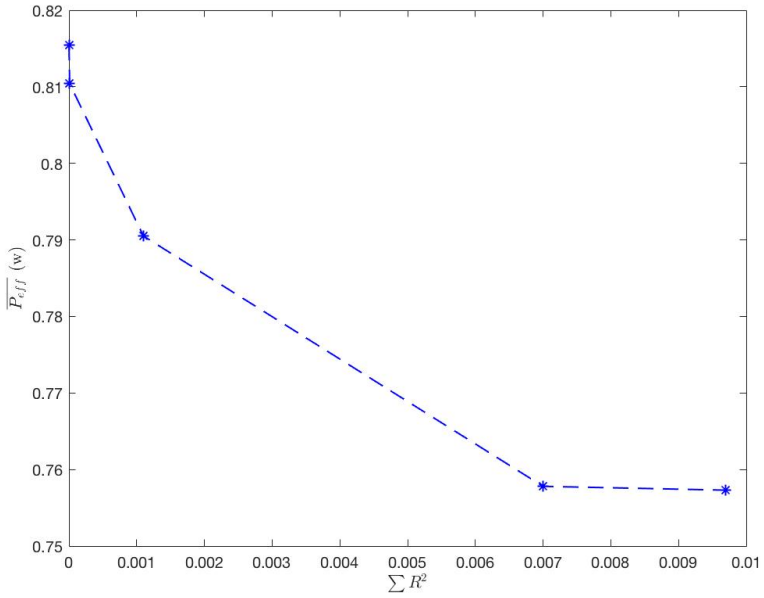

Figure 5.1: Experiment 1 - The tradeoff between actual power consumption and residual

ϖ_R	ϖ_{Pact}	$\overline{P_{act}}$ *(W)*	$\overline{P_{eff}}$ *(W)*	$\sum R^2$
1	*0*	*1.1531*	*1.5815*	*2.4549E-10*
0	*1*	*1.9987E-06*	*2.087E-07*	*3838.7*
100	*1*	*0.8155*	*1.0458*	*1.1991E-07*
10	*1*	*0.8104*	*1.0404*	*0.00001099*
1	*1*	*0.7905*	*1.0113*	*0.0011*
1	*10*	*0.7578*	*0.9611*	*0.007*
1	*100*	*0.7573*	*0.9603*	*0.0097*

Table 5.3: Experiment 1 - The power consumption and residual

We put the solution (when ϖ_{Pact}=0) into the FWMAV model in equation (2.4) and run the model for 20 flapping cycles using *ode45*. Figure 5.2 is the orbit we plot using part of the data (*x(i* $\times T_{seg}$*)(i=0,1,2...,31,32)*), which are also the part of the data the optimizer used. To be noted, this orbit may not accurately reflect the actual flight status. The reason is that the optimizer only use part data of the state vector *x(t)* instead of the full data of state vector *x(t)* calculated by *ode45*, for the purpose of getting less optimization running time. It's a tradeoff between the accuracy and optimization running time. Based on the solution (when ϖ_{Pact}=0) find by the optimizer, we also plot the orbit using all data of state vector

x(t) calculated by *ode45*, as shown in figure 5.3.

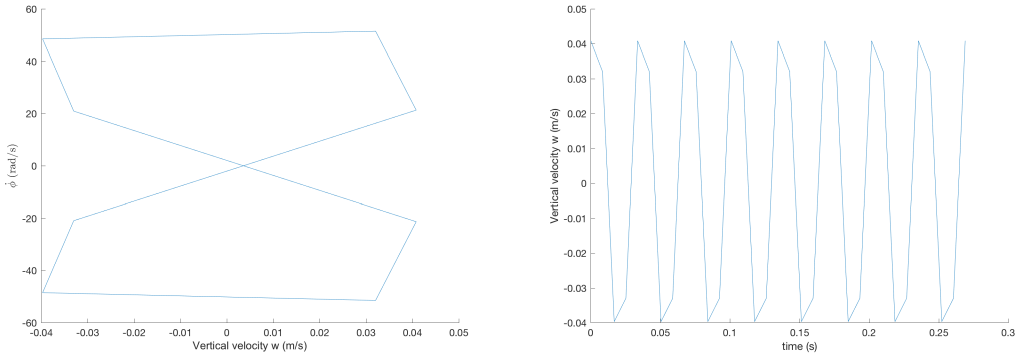

Figure 5.2: Experiment 1 - The orbit of 4 cycles using part state vector data ($\mathfrak{w}_{Pact}$=0)

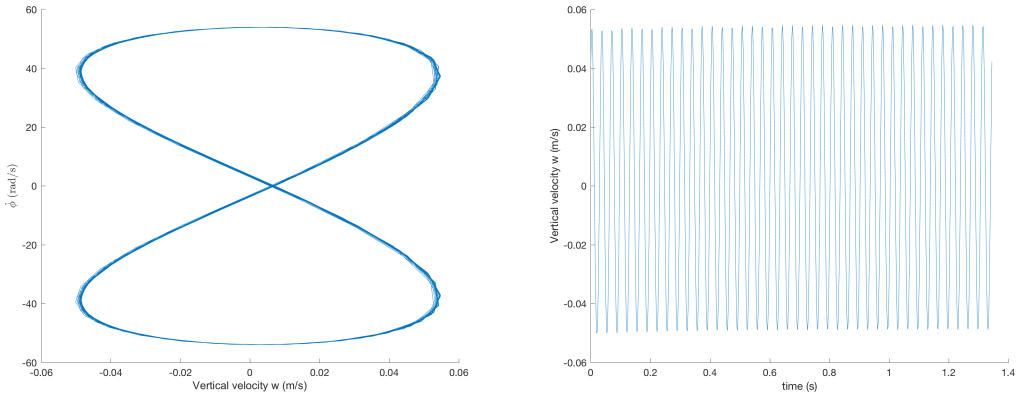

Figure 5.3: Experiment 1 - The orbit of 20 flapping cycles using complete state vector data ($\mathfrak{w}_{Pact}$=0)

5.2.2 Battery Model Effection Analysis

Experiment 2 - Static Flight with Considering Battery Model In this section, we presents the experimental results of experiment 2 described in chapter 4. To get more accurate solutions, the battery model is considered in experiment 2. The cost function is adjusted and represented by equation (4.9), where P_{eff} is the average of the effective power consumption, which is recalculated by equation (3.6). The term of residual $\sum R^2$ is still calculated by equations (4.5).

We still run the optimizations *100* times with different random initial values of the control input variables, then choose the solution out of the 100 sets of solutions which introduce minimal $Cost_2$. In addition, we also run this optimization by using the initial values in table 5.2 from the model without battery. By comparing this solution with the one chosen from 100 optimizations with random initial values, the final solution is the one which introduces the minimal $Cost_2$. With battery model considered, for different settings of the weights, we get different solutions as shown in table 5.4, as well as the values of initial control inputs in table 5.5.

$\mathfrak{w}_R$	$\mathfrak{w}_{Peff}$	U	α_m *(rad)*	ω *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	*0.0788*	*0.5831*	*93.4523*	*-0.5158*	*0.0408*	*21.3174*
0	*1*	*0.0000*	*0.4243*	*312.7124*	*1.5297*	*0.0162*	*-75.2778*
100	*1*	*0.0495*	*0.3569*	*50.2664*	*-1.2512*	*0.0731*	*22.5453*
10	*1*	*0.0496*	*0.3527*	*50.2694*	*-1.1583*	*0.0795*	*22.2601*
1	*1*	*0.0489*	*0.3418*	*50.2655*	*-1.1421*	*0.1478*	*21.4065*
1	*10*	*0.0485*	*0.3301*	*50.2655*	*-1.1979*	*0.1990*	*20.5391*
1	*100*	*0.0485*	*0.3297*	*50.2655*	*-1.1982*	*0.1916*	*20.4517*

Table 5.4: Experiment 2 - Control inputs under different weights

$\mathfrak{w}_R$	$\mathfrak{w}_{Peff}$	U	α_m *(rad)*	ω *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	*0.0721*	*0.5663*	*88.8209*	*-0.3271*	*-0.1217*	*16.9859*
0	*1*	*0.1564*	*0.8960*	*261.7321*	*1.4324*	*-0.1573*	*-38.3165*
100	*1*	*0.1319*	*0.4518*	*167.7397*	*-1.2026*	*-0.1682*	*-5.8743*
10	*1*	*0.1578*	*0.9067*	*189.5122*	*-0.7557*	*0.0368*	*-16.9896*
1	*1*	*0.0731*	*0.3470*	*81.5081*	*-1.0087*	*0.0149*	*9.4111*
1	*10*	*0.1307*	*0.5560*	*175.3444*	*-0.3793*	*-0.1151*	*17.8239*
1	*100*	*0.1503*	*0.4273*	*168.5789*	*-0.1530*	*0.1684*	*6.7778*

Table 5.5: Experiment 2 - Random initial control inputs under different weights

The average of actual power consumption $\overline{P_{act}}$, the average of effective power consumption $\overline{P_{eff}}$ and the sum of the square of residual $\sum R^2$ are presented in the table 5.6. The tradeoff between the two parts in the cost function can also be seen, as in figure 5.4. With smaller $\mathfrak{w}_{Peff}$, we get larger $\overline{P_{eff}}$ and smaller $\sum R^2$.

Compare table 5.3 and table 5.6, we can see that the actual power consumption of the system

$\mathfrak{w}_R$	$\mathfrak{w}_{Peff}$	$\overline{P_{act}}$ (W)	$\overline{P_{eff}}$ (W)	$\sum R^2$
1	*0*	*1.1531*	*1.5815*	*2.4553E-10*
0	*1*	*2.9336E-06*	*3.3164E-07*	*4932*
100	*1*	*0.813*	*1.043*	*4.1627E-07*
10	*1*	*0.8096*	*1.0383*	*2.7285E-05*
1	*1*	*0.7802*	*0.9941*	*0.0024*
1	*10*	*0.7577*	*0.961*	*0.007*
1	*100*	*0.7571*	*0.96*	*0.0122*

Table 5.6: Experiment 2 - The power consumption and residual

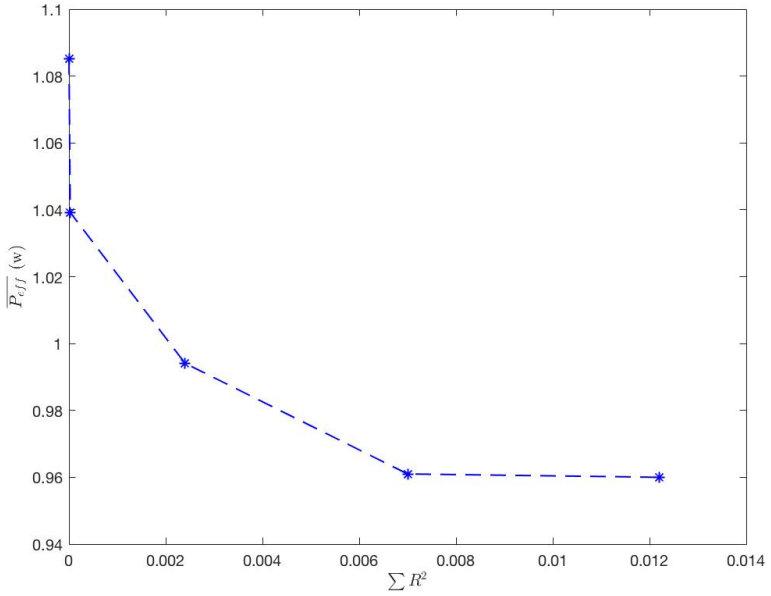

Figure 5.4: Experiment 2 - The tradeoff between effective power consumption and residual

when we consider the battery model is slightly smaller than the actual power consumption of the system when we did not consider the battery model. The effective power is larger than the actual power when its value is larger than the threshold. In our experiments, this threshold is *0.4810W*. It is calculated using the battery model by making effective power consumption equal to actual power consumption. Replacing the actual power consumption with the effective power consumption in the cost function is equivalent to giving a larger weight to the power consumption term, hence the optimizer gives a solution with a smaller power consumption. Although it is not a large improvement, it is always good to take the

battery model into consideration to get more accurate and better solution.

5.3 Dynamic Vertical Flight

The results of experiment 1 and experiment 2 have proved that the static vertical flight optimization doesn't have much flexibility because of the two strict constraints: the orbit is periodic and the FWMAV can hover. In static flight stage, the solutions under different settings of weights are very close to each other and there is no much improvement of power consumption. Hence, in dynamic flight stage, we relaxed these two constraints to allow the FWMAV to fly in more flexible vertical orbits and expect the optimizer to find dynamic flight orbits with smaller power consumption than static flight orbits.

In the dynamic flight stage, we introduce more control inputs to control each flapping cycle in the pattern separately. The control inputs now become (4.10), which is composed by the control inputs for each flapping cycle and the initial state of the entire flight.

5.3.1 Dynamic Flight with Flexible Displacement

Experiment 3 - Dynamic Flight without Considering Battery Model In this section, we presents the experimental results of experiment 3 described in chapter 4. When the battery model is not considered, the cost function of the optimization is shown as in equation (4.13).

Same as experiments for static flight, we run the optimizations *100* times with different random initial values of the control input variables, then choose the final solution out of the 100 sets of solutions which introduce minimal $Cost_3$. The solutions are shown in table 5.7 for different settings of the weights, as well as the values of random initial control inputs in

table 5.8 which introduce the best solution out of *100* simulations.

$\mathfrak{w}_{R'}$	$\mathfrak{w}_{Pact}$	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.1181*	*0.7094*	*149.2505*	*-0.3272*	*0.0229*	*17.5436*
		$i = 2$	*0.1181*	*0.7094*	*149.2503*			
		$i = 3$	*0.1181*	*0.7094*	*149.2502*			
		$i = 4$	*0.1181*	*0.7094*	*149.2502*			
0	*1*	$i = 1$	*0.0000*	*0.7678*	*305.0217*	*0.0120*	*-0.4295*	*-0.0003*
		$i = 2$	*0.0000*	*0.8481*	*274.7026*			
		$i = 3$	*0.0000*	*0.7911*	*299.4220*			
		$i = 4$	*0.0000*	*0.8220*	*292.7134*			
100	*1*	$i = 1$	*0.0001*	*1.3591*	*314.1590*	*0.0006*	*-0.3909*	*-0.0001*
		$i = 2$	*0.0001*	*1.3381*	*314.1591*			
		$i = 3$	*0.0001*	*0.2321*	*314.1591*			
		$i = 4$	*0.0001*	*0.2096*	*314.1590*			
10	*1*	$i = 1$	*0.0006*	*1.3944*	*314.1590*	*0.0042*	*-0.3817*	*-0.0006*
		$i = 2$	*0.0006*	*1.3842*	*314.1591*			
		$i = 3$	*0.0006*	*0.1860*	*314.1591*			
		$i = 4$	*0.0006*	*0.1762*	*314.1590*			
1	*1*	$i = 1$	*0.0492*	*0.3303*	*50.2660*	*-1.2228*	*0.1452*	*20.6494*
		$i = 2$	*0.0492*	*0.3301*	*50.2667*			
		$i = 3$	*0.0492*	*0.3300*	*50.2672*			
		$i = 4$	*0.0492*	*0.3298*	*50.2675*			
1	*10*	$i = 1$	*0.0497*	*0.3494*	*50.2707*	*-1.2303*	*0.0788*	*22.0468*
		$i = 2$	*0.0497*	*0.3494*	*50.2708*			
		$i = 3$	*0.0497*	*0.3494*	*50.2709*			
		$i = 4$	*0.0497*	*0.3494*	*50.2709*			
1	*100*	$i = 1$	*0.0484*	*0.4188*	*50.2694*	*-0.6513*	*0.0844*	*26.4360*
		$i = 2$	*0.0484*	*0.4188*	*50.2694*			
		$i = 3$	*0.0484*	*0.4188*	*50.2694*			
		$i = 4$	*0.0484*	*0.4188*	*50.2694*			

Table 5.7: Experiment 3 - Control inputs under different weights

The average of actual power consumption $\overline{P_{act}}$, the average of effective power consumption $\overline{P_{eff}}$, the sum of the square of residual $\sum R'^2$ and δz are calculated using the results in table 5.7 are presented in the table 5.9. As there are three items in the cost function, the tradeoff is not necessarily shown.

$\mathfrak{w}_{R'}$	$\mathfrak{w}_{Pact}$	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.0782*	*0.3591*	*229.4566*	*-0.6882*	*-0.2898*	*26.5908*
		$i = 2$	*0.0834*	*0.8742*	*211.6721*			
		$i = 3$	*0.1257*	*0.5227*	*193.8777*			
		$i = 4$	*0.2921*	*0.7888*	*185.1599*			
0	*1*	$i = 1$	*0.3816*	*0.3099*	*306.8500*	*-1.2730*	*0.4118*	*15.5052*
		$i = 2$	*0.2875*	*0.4537*	*287.7536*			
		$i = 3$	*0.4322*	*0.3114*	*293.3811*			
		$i = 4$	*0.3920*	*0.1922*	*296.2336*			
100	*1*	$i = 1$	*0.1687*	*0.6431*	*253.7519*	*-1.2669*	*-0.4464*	*15.6696*
		$i = 2$	*0.1142*	*0.9930*	*103.0078*			
		$i = 3$	*0.0262*	*0.9306*	*227.5075*			
		$i = 4$	*0.0225*	*0.4338*	*203.2910*			
10	*1*	$i = 1$	*0.2347*	*0.9919*	*300.9255*	*-0.8846*	*0.2722*	*12.8957*
		$i = 2$	*0.2677*	*0.7930*	*308.2358*			
		$i = 3$	*0.3424*	*0.8906*	*313.3759*			
		$i = 4$	*0.0077*	*0.7998*	*283.4552*			
1	*1*	$i = 1$	*0.0886*	*0.7798*	*269.1679*	*-1.1552*	*0.0103*	*60.8832*
		$i = 2$	*0.3810*	*1.0140*	*230.1440*			
		$i = 3$	*0.0627*	*0.9554*	*214.7371*			
		$i = 4$	*0.1739*	*0.4036*	*255.1217*			
1	*10*	$i = 1$	*0.1718*	*0.1904*	*236.6020*	*0.0797*	*-0.4305*	*16.4276*
		$i = 2$	*0.2037*	*0.4636*	*291.5922*			
		$i = 3$	*0.1046*	*0.4448*	*242.7237*			
		$i = 4$	*0.0443*	*0.3749*	*131.2082*			
1	*100*	$i = 1$	*0.0291*	*0.6217*	*170.7458*	*1.1884*	*-0.4675*	*-21.5364*
		$i = 2$	*0.2886*	*0.2143*	*233.8591*			
		$i = 3$	*0.3174*	*0.1848*	*309.7625*			
		$i = 4$	*0.2353*	*0.9810*	*219.9779*			

Table 5.8: Experiment 3 - Random initial control inputs under different weights

Experiment 4 - Dynamic Flight with Considering Battery Model In this section, we presents the experimental results of experiment 4 described in chapter 4. When the battery model is considered, the cost function of the optimization is shown as in equation (4.17).

We still run the optimizations *100* times with different random initial values of the control input variables, then choose the solution out of the 100 sets of solutions which introduce

$\mathfrak{w}_{R'}$	$\mathfrak{w}_{Pact}$	$\overline{P_{act}}$ *(W)*	$\overline{P_{eff}}$ *(W)*	$\sum R'^2$	δz *(m)*
1	*0*	*1.5474*	*2.2611*	*1.8377E-07*	*1.1529E-04*
0	*1*	*3.1957E-08*	*1.4926E-09*	*1.1225*	*7.322E-04*
100	*1*	*1.04E-07*	*6.42E-09*	*0.9604*	*8.41E-05*
10	*1*	*9.15E-06*	*1.38E-06*	*0.9602*	*8.19E-04*
1	*1*	*0.7781*	*0.9924*	*6.56E-05*	*0.0406*
1	*10*	*0.8092*	*1.0381*	*8.44E-07*	*0.0043*
1	*100*	*0.8302*	*1.0633*	*3.27E-08*	*6.32E-04*

Table 5.9: Experiment 3 - The power consumption and residual

minimal $Cost_4$. In addition, we also run this optimization by using the initial values in table 5.8 from experiment 3. By comparing this solution with the one chosen from 100 optimizations with random initial values, the final solution is the one which introduces the minimal $Cost_4$. With battery model considered, for different settings of the weights, we get solutions as shown in table 5.10, as well as the values of initial control inputs in table 5.11.

The average of actual power consumption $\overline{P_{act}}$, the average of effective power consumption $\overline{P_{eff}}$, the sum of the square of residual $\sum R'^2$ and δz are presented in the table 5.12.

As expected, by comparing table 5.9 and table 5.12, we can see that the actual power consumption of the system when we consider the battery model is slightly smaller than the actual power consumption of the system when we did not consider the battery model. Although it is not a large improvement, it is always good to take the battery model into consideration.

However, by comparing the results with the results for static flight, we still couldn't see more flexibility here. Inside the pattern, all cycles have almost same values for the control inputs. So, the results proved that the constraints are still too strict and are still restraining the flight flexibility. Hence, we made more improvements and designed the experiments in following sections.

$\mathfrak{w}_{R'}$	$\mathfrak{w}_{Peff}$	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.1147*	*0.7221*	*143.8430*	*-0.3360*	*0.0249*	*18.6190*
		$i = 2$	*0.1147*	*0.7221*	*143.8428*			
		$i = 3$	*0.1147*	*0.7221*	*143.8427*			
		$i = 4$	*0.1147*	*0.7221*	*143.8426*			
0	*1*	$i = 1$	*0.0000*	*1.3086*	*267.0573*	*-1.4446*	*0.3348*	*617.6857*
		$i = 2$	*0.0000*	*0.9669*	*304.8888*			
		$i = 3$	*0.0000*	*0.7399*	*299.6291*			
		$i = 4$	*0.0000*	*0.7311*	*205.0992*			
100	*1*	$i = 1$	*0.0004*	*1.3928*	*314.1590*	*-0.0005*	*-0.3852*	*-0.0005*
		$i = 2$	*0.0004*	*1.3819*	*314.1591*			
		$i = 3$	*0.0004*	*0.1874*	*314.1591*			
		$i = 4$	*0.0004*	*0.1778*	*314.1590*			
10	*1*	$i = 1$	*0.0023*	*1.3961*	*314.1591*	*-0.0066*	*-0.3479*	*0.0519*
		$i = 2$	*0.0023*	*1.2651*	*314.1590*			
		$i = 3$	*0.0023*	*0.2723*	*314.1592*			
		$i = 4$	*0.0023*	*0.1749*	*314.1430*			
1	*1*	$i = 1$	*0.0561*	*0.7700*	*314.1592*	*-0.0233*	*-0.2294*	*0.7216*
		$i = 2$	*0.0561*	*0.7027*	*314.0741*			
		$i = 3$	*0.0561*	*0.6393*	*314.0298*			
		$i = 4$	*0.0561*	*0.6045*	*313.9585*			
1	*10*	$i = 1$	*0.0495*	*0.3525*	*50.2820*	*-1.3415*	*0.0837*	*22.2284*
		$i = 2$	*0.0495*	*0.3525*	*50.2821*			
		$i = 3$	*0.0495*	*0.3525*	*50.2822*			
		$i = 4$	*0.0495*	*0.3525*	*50.2822*			
1	*100*	$i = 1$	*0.0497*	*0.3514*	*50.2688*	*-0.8677*	*0.0726*	*22.1924*
		$i = 2$	*0.0497*	*0.3514*	*50.2688*			
		$i = 3$	*0.0497*	*0.3514*	*50.2688*			
		$i = 4$	*0.0497*	*0.3514*	*50.2688*			

Table 5.10: Experiment 4 - Control inputs under different weights

5.3.2 Dynamic Flight with Flexible States

Experiment 5 - Dynamic Flight without Considering Battery Model In this section, we presents the experimental results of experiment 5 described in chapter 4. When the battery model is not considered, the cost function of the optimization is shown as in equation (4.18).

$\varpi_{R'}$	ϖ_{Peff}	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.1167*	*0.4003*	*151.6400*	*-0.7435*	*0.0355*	*35.2048*
		$i = 2$	*0.0863*	*0.9543*	*235.0855*			
		$i = 3$	*0.3957*	*0.6311*	*261.0183*			
		$i = 4$	*0.1145*	*0.2210*	*227.7031*			
0	*1*	$i = 1$	*0.0004*	*1.0412*	*112.0099*	*-1.4614*	*-0.2420*	*130.3366*
		$i = 2$	*0.4364*	*1.0157*	*286.8618*			
		$i = 3$	*0.0353*	*0.6392*	*263.0845*			
		$i = 4$	*0.1100*	*0.6582*	*161.3053*			
100	*1*	$i = 1$	*0.0867*	*0.7911*	*301.1131*	*-1.0584*	*-0.4578*	*-19.0106*
		$i = 2$	*0.2396*	*1.0430*	*300.3242*			
		$i = 3$	*0.0292*	*1.0071*	*238.6175*			
		$i = 4$	*0.3057*	*0.7728*	*254.3935*			
10	*1*	$i = 1$	*0.2259*	*0.5047*	*208.1798*	*0.5846*	*0.4300*	*-9.2962*
		$i = 2$	*0.3158*	*0.3833*	*291.4107*			
		$i = 3$	*0.1426*	*0.8521*	*303.9895*			
		$i = 4$	*0.5314*	*0.5409*	*309.1841*			
1	*1*	$i = 1$	*0.0335*	*0.8238*	*256.5005*	*1.4037*	*-0.2648*	*-66.8989*
		$i = 2$	*0.2322*	*0.1859*	*228.5266*			
		$i = 3$	*0.1140*	*0.4958*	*258.0570*			
		$i = 4$	*0.2798*	*0.9827*	*251.6505*			
1	*10*	$i = 1$	*0.2713*	*0.2519*	*280.6682*	*-0.9636*	*0.4134*	*13.9248*
		$i = 2$	*0.2050*	*0.4787*	*251.8486*			
		$i = 3$	*0.5066*	*0.5348*	*304.6421*			
		$i = 4$	*0.1458*	*0.7445*	*244.2045*			
1	*100*	$i = 1$	*0.0980*	*0.2631*	*164.0674*	*-0.6861*	*-0.1987*	*23.6976*
		$i = 2$	*0.4835*	*0.2046*	*298.7946*			
		$i = 3$	*0.1841*	*0.7375*	*281.8627*			
		$i = 4$	*0.0145*	*0.7401*	*278.3242*			

Table 5.11: Experiment 4 - Random initial control inputs under different weights

The optimizations are still run *100* times with different random initial values of the control input variables, then choose the solution out of the 100 sets of solutions which introduce minimal $Cost_5$. With different settings of the weights, we get solutions as shown in table 5.13, as well as the values of initial control inputs in table 5.14.

The average of actual power consumption $\overline{P_{act}}$, the average of effective power consumption $\overline{P_{eff}}$, the revised sum of square of residual $\sum {R_{pattern}}^2$ are presented in the table 5.15.

$\varpi_{R'}$	ϖ_{Peff}	$\overline{P_{act}}$ (W)	$\overline{P_{eff}}$ (W)	$\sum R'^2$	δz (m)
1	*0*	*1.5426*	*2.2486*	*1.86E-07*	*1.1637E-04*
0	*1*	*6.19E-07*	*5.12E-08*	*3.74E+05*	*-4.39E-04*
100	*1*	*4.06E-06*	*5.20E-07*	*0.9603*	*5.46E-04*
10	*1*	*1.63E-04*	*4.16E-05*	*0.9594*	*0.0035*
1	*1*	*0.0985*	*0.0904*	*0.8327*	*0.0111*
1	*10*	*0.8077*	*1.0353*	*1.87E-06*	*0.0065*
1	*100*	*0.8122*	*1.0425*	*2.83E-08*	*8.62E-04*

Table 5.12: Experiment 4 - The power consumption and residual

Experiment 6 - Dynamic Flight with Considering Battery Model In this section, we presents the experimental results of experiment 6 described in chapter 4. When the battery model is considered, the cost function of the optimization is shown as in equation (4.23).

We run the optimizations *100* times with different random initial values of the control input variables, then choose the solution out of the 100 sets of solutions which introduce minimal $Cost_6$. In addition, we also run this optimization by using the initial values in table 5.14 from the model without battery. By comparing this solution with the one chosen from 100 optimizations with random initial values, the final solution is the one which introduces the minimal $Cost_6$. With battery model considered, for different settings of the weights, we get solutions as shown in table 5.16, as well as the values of initial control inputs in table 5.17.

The average of actual power consumption $\overline{P_{act}}$, the average of effective power consumption $\overline{P_{eff}}$ and the sum of the square of residual $\sum {R_{pattern}}^2$ are presented in the table 5.18.

Comparing the results with the orbits with only flexible displacement, we get more flexible orbits with smaller power consumption. The patterns (4 cycles) are also reproducible to some extend. Figure 5.5 are the pattern (4 cycles) gotten from the FWMAV model when considering battery model as well as setting *[ϖ_{Rpat}=100, ϖ_{Peff}=1]*. And figure 5.6 and 5.7 are orbits by running the FWMAV model in 5 patterns (20 cycles) and 50 patterns (200

$\mathfrak{w}_{Rpat}$	$\mathfrak{w}_{Pact}$	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.1669*	*0.7451*	*274.8118*	*0.1694*	*-0.1002*	*-2.4122*
		$i = 2$	*0.1675*	*0.7344*	*281.0717*			
		$i = 3$	*0.1224*	*0.6195*	*138.8545*			
		$i = 4$	*0.1021*	*0.6913*	*99.1097*			
0	*1*	$i = 1$	*0.0000*	*0.6514*	*308.2942*	*0.9639*	*-0.4564*	*-41.8848*
		$i = 2$	*0.0000*	*0.7073*	*310.1534*			
		$i = 3$	*0.0000*	*0.8347*	*312.0582*			
		$i = 4$	*0.0000*	*0.4294*	*106.4339*			
100	*1*	$i = 1$	*0.0000*	*1.3135*	*314.1589*	*0.0060*	*-0.3824*	*0.1447*
		$i = 2$	*0.0000*	*1.2374*	*314.1589*			
		$i = 3$	*0.0000*	*0.1794*	*314.1589*			
		$i = 4$	*0.0000*	*0.1763*	*314.1589*			
10	*1*	$i = 1$	*0.0000*	*1.0475*	*314.1590*	*0.0005*	*-0.3821*	*-0.0130*
		$i = 2$	*0.0000*	*1.0351*	*314.1590*			
		$i = 3$	*0.0005*	*0.1796*	*314.1590*			
		$i = 4$	*0.0015*	*0.1757*	*314.1590*			
1	*1*	$i = 1$	*0.0000*	*0.5728*	*250.7047*	*0.9122*	*0.1612*	*28.0554*
		$i = 2$	*0.0554*	*0.2688*	*62.6762*			
		$i = 3$	*0.0000*	*0.3045*	*314.1213*			
		$i = 4$	*0.0000*	*0.4038*	*314.1212*			
1	*10*	$i = 1$	*0.0000*	*0.7198*	*262.4754*	*1.0482*	*-0.0163*	*24.8151*
		$i = 2$	*0.0629*	*0.3149*	*65.6189*			
		$i = 3$	*0.0000*	*0.3876*	*314.1589*			
		$i = 4$	*0.0000*	*0.5429*	*314.1589*			
1	*100*	$i = 1$	*0.0000*	*0.7247*	*257.0584*	*1.0930*	*-0.0481*	*21.9615*
		$i = 2$	*0.0626*	*0.3337*	*64.2646*			
		$i = 3$	*0.0000*	*0.4126*	*314.1256*			
		$i = 4$	*0.0000*	*0.6422*	*314.1275*			

Table 5.13: Experiment 5 - Control inputs under different weights

cycles), respectively. All of orbits in above figures are using complete data of state vectors calculated by *ode45*.

The δz (when *[*$\mathfrak{w}_{Rpat}$*=100,* $\mathfrak{w}_{Peff}$*=1]*) is *9.1258e-04* meters in the duration of 1 pattern (0.1668 seconds) when we calculate it using the sampled state vector data. It is not very large, but it will get a lot larger when we apply the control variables to the FWMAV model and plot the orbits with complete state vector data calculated by *ode45*. We can see the

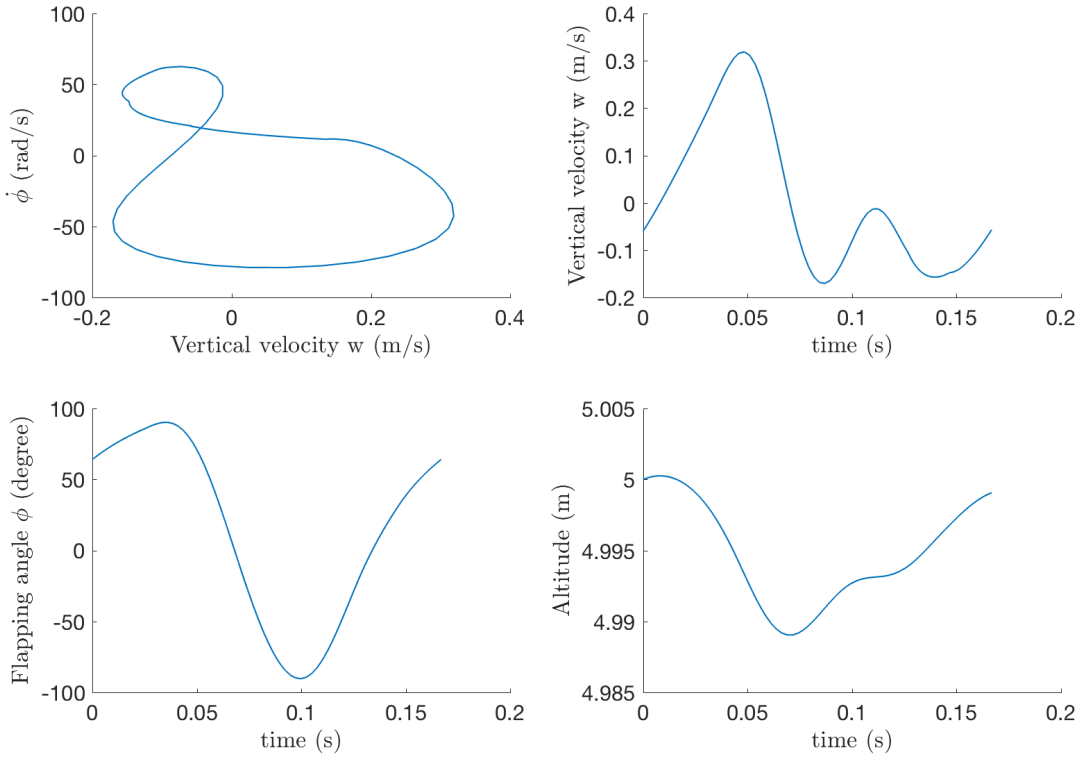

Figure 5.5: Experiment 6 - The dynamic flight with flexible states (1 pattern, *[*ϖ_{Rpat}*=100,* ϖ_{Peff}*=1]*)

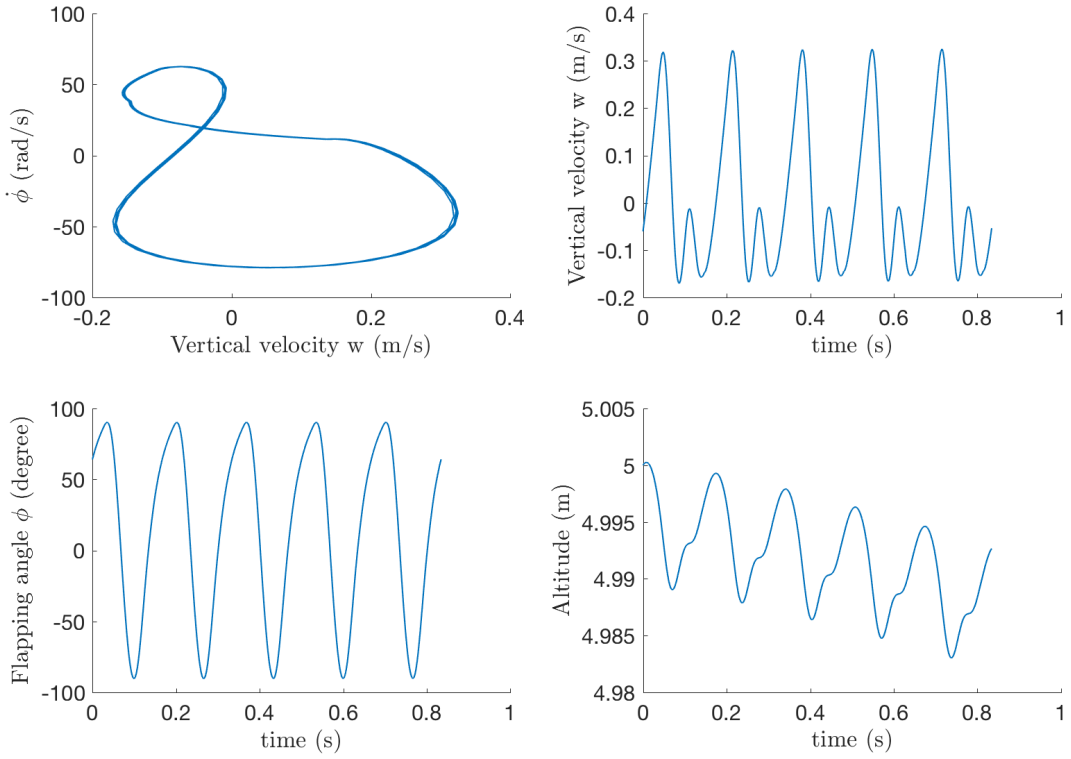

Figure 5.6: Experiment 6 - The dynamic flight with flexible states (5 patterns, *[*ϖ_{Rpat}*=100,* ϖ_{Peff}*=1]*)

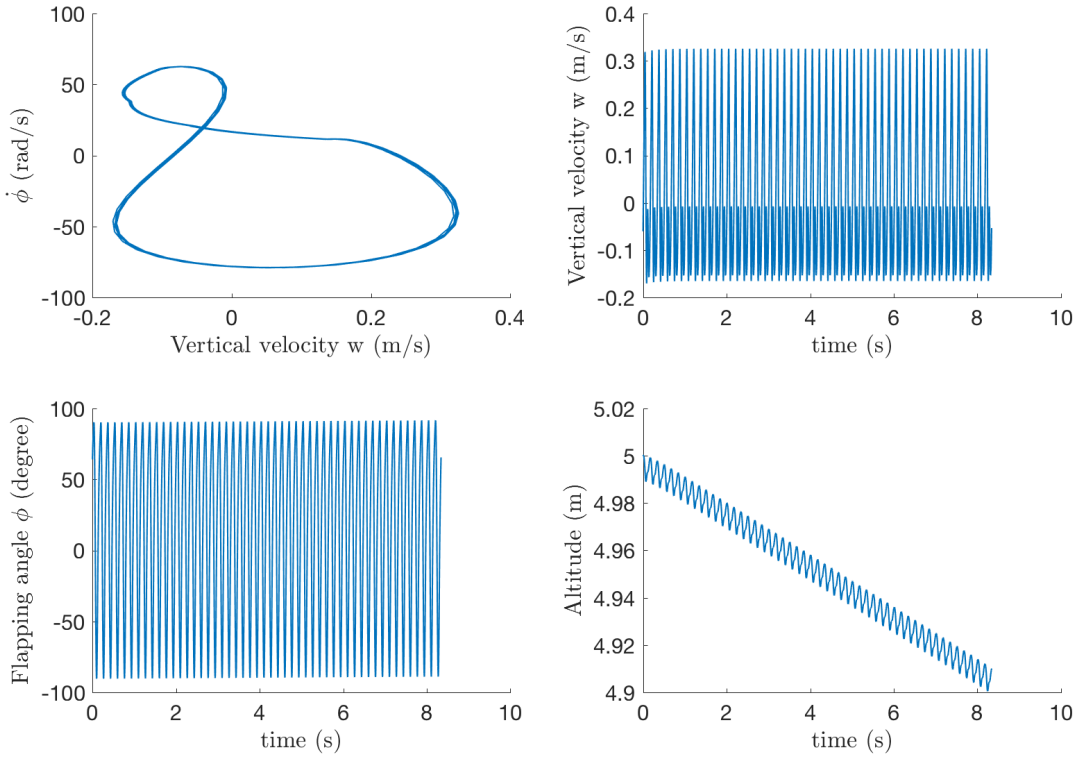

Figure 5.7: Experiment 6 - The dynamic flight with flexible states (50 patterns, *[ϖ$_{Rpat}$=100, ϖ$_{Peff}$=1]*)

$\mathfrak{w}_{Rpat}$	$\mathfrak{w}_{Pact}$	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.0118*	*0.9554*	*283.8770*	*-1.0707*	*0.3283*	*70.7451*
		$i = 2$	*0.3722*	*0.7068*	*286.4628*			
		$i = 3$	*0.0653*	*0.6108*	*150.9396*			
		$i = 4$	*0.1406*	*0.9382*	*92.5763*			
0	*1*	$i = 1$	*0.2711*	*0.8539*	*292.8144*	*-1.1979*	*0.4510*	*19.0157*
		$i = 2$	*0.3119*	*0.7108*	*251.2721*			
		$i = 3$	*0.1059*	*1.0187*	*261.6941*			
		$i = 4$	*0.0511*	*0.8007*	*117.2954*			
100	*1*	$i = 1$	*0.1199*	*0.9988*	*102.0776*	*0.1665*	*0.1927*	*-22.7769*
		$i = 2$	*0.1898*	*0.8234*	*259.6352*			
		$i = 3$	*0.1735*	*0.3505*	*140.7444*			
		$i = 4$	*0.0580*	*0.5794*	*106.9839*			
10	*1*	$i = 1$	*0.0943*	*0.6342*	*103.6874*	*0.0187*	*0.1114*	*-21.2570*
		$i = 2$	*0.3094*	*0.8412*	*262.3302*			
		$i = 3$	*0.2643*	*0.4163*	*262.3918*			
		$i = 4$	*0.1191*	*0.9908*	*225.3855*			
1	*1*	$i = 1$	*0.1446*	*0.9825*	*267.8893*	*-1.0626*	*-0.4511*	*33.0990*
		$i = 2$	*0.0105*	*0.3631*	*93.3155*			
		$i = 3$	*0.2695*	*0.8331*	*214.0811*			
		$i = 4$	*0.2946*	*0.8975*	*286.7589*			
1	*10*	$i = 1$	*0.2876*	*0.1879*	*285.9967*	*-0.3446*	*-0.2127*	*15.0246*
		$i = 2$	*0.1123*	*0.8318*	*85.2470*			
		$i = 3$	*0.2781*	*0.2018*	*235.0910*			
		$i = 4$	*0.1186*	*0.5578*	*283.5652*			
1	*100*	$i = 1$	*0.2426*	*0.7842*	*293.4574*	*-1.0495*	*0.3751*	*3.8779*
		$i = 2$	*0.1416*	*0.5227*	*117.0628*			
		$i = 3$	*0.1292*	*0.2805*	*245.8052*			
		$i = 4$	*0.1659*	*0.3807*	*243.8042*			

Table 5.14: Experiment 5 - Random initial control inputs under different weights

obvious drop of the altitude from the figure 5.7, which are drawn using the complete state vector data calculated by *ode45*. The altitude drops *0.0009* meters in 1 pattern duration (0.1668 seconds) and even *0.0073* meters in 5 pattern duration (0.8341 seconds). We let the FWMAV model run for 50 pattern duration (8.341 seconds), the altitude drops 0.0899 meters, which is quite large.

ϖ_{Rpat}	ϖ_{Pact}	$\overline{P_{act}}$ (W)	$\overline{P_{eff}}$ (W)	$\sum {R_{pattern}}^2$
1	*0*	*1.512*	*2.2166*	*5.82E-12*
0	*1*	*5.12E-08*	*2.49E-09*	*1.21E+03*
100	*1*	*1.26E-08*	*4.66E-10*	*0.6145*
10	*1*	*1.76E-05*	*3.69E-06*	*0.6143*
1	*1*	*0.2168*	*0.2978*	*1.80E-02*
1	*10*	*0.2903*	*0.4107*	*3.12E-04*
1	*100*	*0.3025*	*0.4266*	*3.15E-06*

Table 5.15: Experiment 5 - The power consumption and residual

5.3.3 Dynamic Flight with Residual as a Constraint

Experiment 7 and 8 - Dynamic Flight with Residual as a Constraint In this section, we presents the experimental results of experiment 7 and experiment 8 described in chapter 4.The cost function of the optimization is shown as in equation (4.24) and (4.25), respectively.

In table 5.19, we present the solutions which are gotten out of 100 times optimizations with different random initial values, with and without considering the battery model. Values in table 5.20 are the random initial values corresponding to the solutions in table 5.19.

Table 5.21 listed the average of actual power consumption $\overline{P_{act}}$, the average of effective power consumption $\overline{P_{eff}}$, the sum of square of residual for the pattern $\sum {R_{pattern}}^2$, as well as the value of δz, $\delta\phi$, δw, $\delta\dot{\phi}$ from the optimization with and without considering battery model. They are calculated by the sampled state vector data, which is used by the optimization.

The data in table 5.21 shows that the values of δz, $\delta\phi$, δw, $\delta\dot{\phi}$ are extremely small. Hence, the pattern should be reproducible. It is proved by applying the solutions in table 5.19 into the FWMAV model and plot the orbit using the complete data of state vector. Here, we use the solutions that we get when we take the battery model into account. Figure 5.8 shows the orbit, vertical velocity, flapping angle and the altitude of the FWMAV in only one pattern (4 cycles). We let the model also run for 5 patterns and 50 patterns, the results are presented

$\mathfrak{w}_{Rpat}$	$\mathfrak{w}_{Peff}$	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.1080*	*0.7788*	*140.9872*	*-0.3051*	*-0.0442*	*2.8505*
		$i = 2$	*0.1185*	*0.6464*	*140.6310*			
		$i = 3$	*0.1224*	*0.8042*	*207.5724*			
		$i = 4$	*0.0979*	*0.7382*	*101.2937*			
0	*1*	$i = 1$	*0.0000*	*0.7253*	*110.7192*	*-1.5533*	*0.0534*	*164.6981*
		$i = 2$	*0.0000*	*1.1432*	*313.1058*			
		$i = 3$	*0.0000*	*0.5643*	*288.4891*			
		$i = 4$	*0.0000*	*0.3731*	*177.5763*			
100	*1*	$i = 1$	*0.0000*	*1.1564*	*314.1590*	*0.0001*	*-0.3864*	*-0.0094*
		$i = 2$	*0.0000*	*0.1848*	*314.1590*			
		$i = 3$	*0.0003*	*0.1794*	*314.1590*			
		$i = 4$	*0.0007*	*0.1772*	*314.1590*			
10	*1*	$i = 1$	*0.0000*	*0.9210*	*314.1589*	*0.0014*	*-0.3689*	*-0.0534*
		$i = 2$	*0.0000*	*0.8948*	*314.1589*			
		$i = 3$	*0.0017*	*0.1755*	*314.1589*			
		$i = 4$	*0.0039*	*0.1749*	*314.1589*			
1	*1*	$i = 1$	*0.0000*	*0.6752*	*242.2963*	*0.9491*	*0.1201*	*26.6640*
		$i = 2$	*0.0497*	*0.2936*	*57.9665*			
		$i = 3$	*0.0000*	*0.3116*	*314.1590*			
		$i = 4$	*0.0000*	*0.4329*	*314.1590*			
1	*10*	$i = 1$	*0.0000*	*0.8167*	*261.9449*	*1.0804*	*-0.0231*	*22.8320*
		$i = 2$	*0.0588*	*0.3424*	*61.4924*			
		$i = 3$	*0.0000*	*0.4113*	*314.1590*			
		$i = 4$	*0.0000*	*0.5984*	*314.1590*			
1	*100*	$i = 1$	*0.0000*	*0.8323*	*259.5602*	*1.1184*	*-0.0603*	*20.7596*
		$i = 2$	*0.0599*	*0.3586*	*61.2325*			
		$i = 3$	*0.0000*	*0.4388*	*314.1585*			
		$i = 4$	*0.0000*	*0.6804*	*314.1578*			

Table 5.16: Experiment 6 - Control inputs under different weights

in figure 5.9 and 5.10, respectively.

When the battery model is considered, the δz in table 5.21 is *1.88E-15* meters in the duration of 1 pattern (0.1418 seconds). Like the experiment 6, it also gets larger when we apply the control variables to the FWMAV model and plot the orbits with complete data calculated by *ode45*. By applying the solution of experiment 8 to the FWMAV model, we get the complete and accurate data of the state vector. The displacement δz is shown in table 5.22. We can

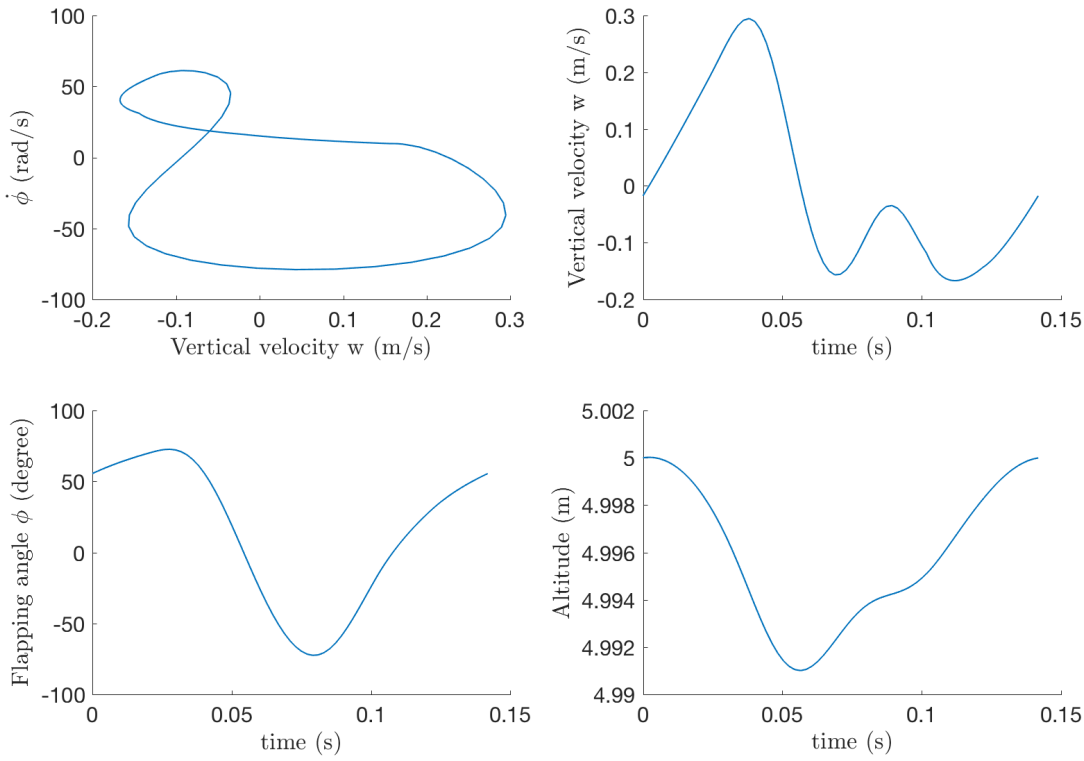

Figure 5.8: Experiment 8 - The dynamic flight orbit in 1 pattern

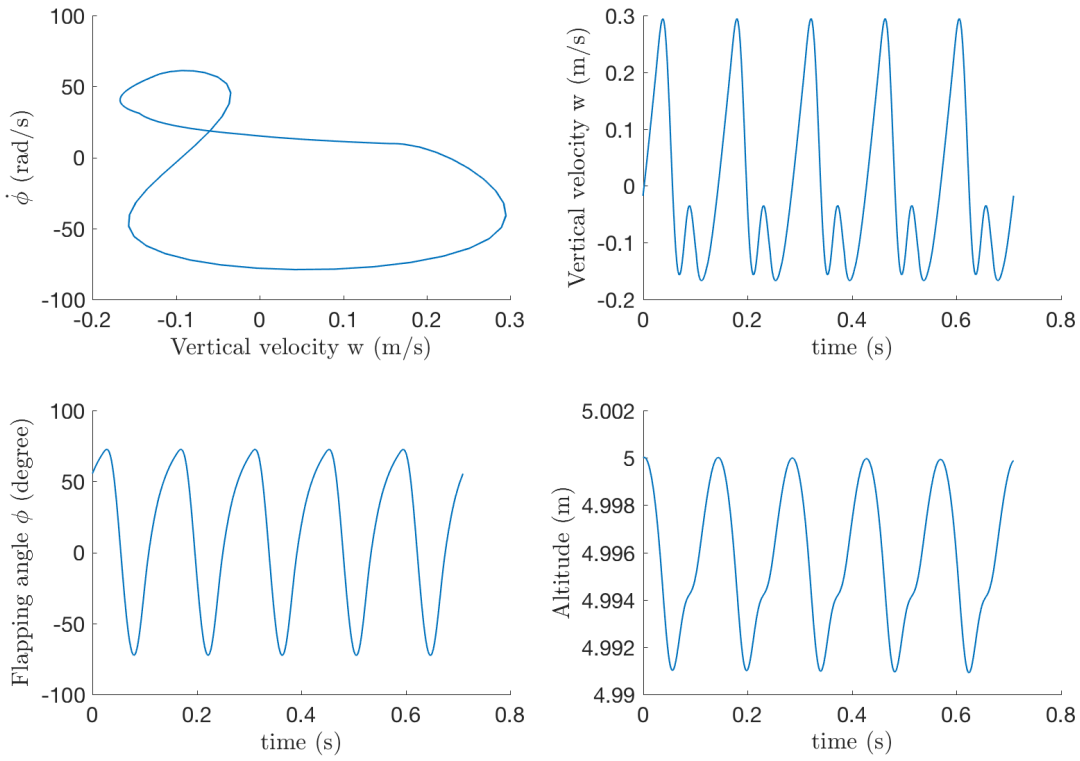

Figure 5.9: Experiment 8 - The dynamic flight orbit in 5 patterns

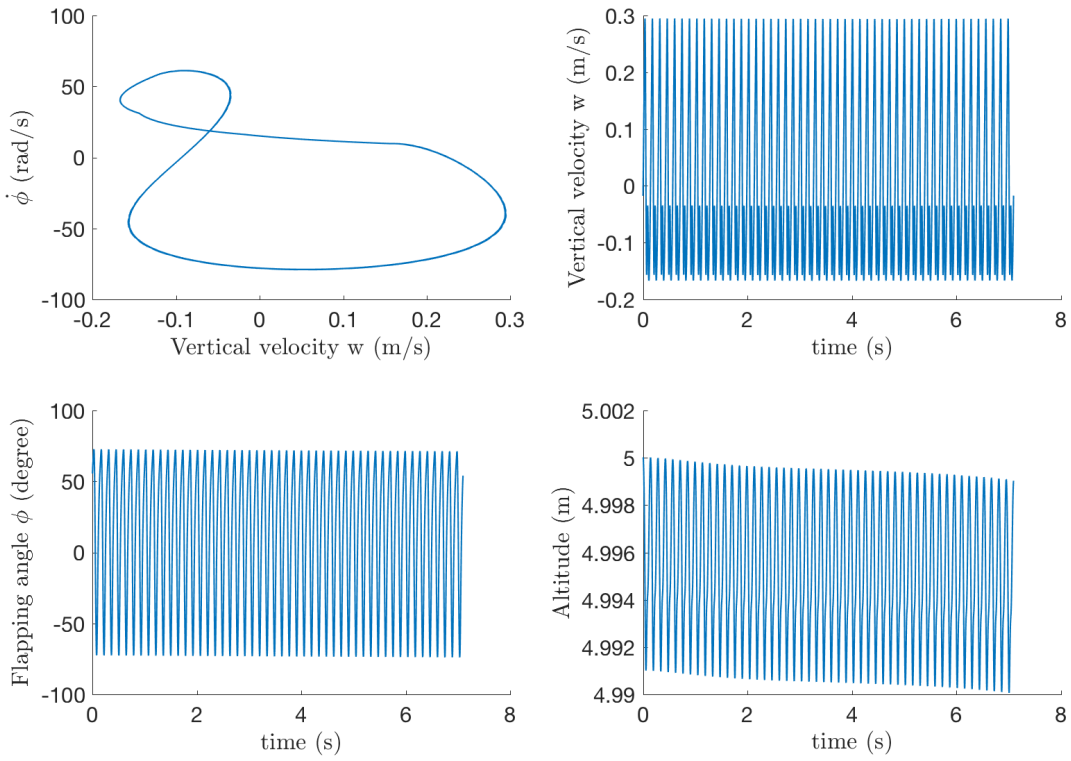

Figure 5.10: Experiment 8 - The dynamic flight orbit in 50 patterns

$\mathfrak{w}_{Rpat}$	$\mathfrak{w}_{Peff}$	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
1	*0*	$i = 1$	*0.1457*	*0.7020*	*197.1088*	*-0.1591*	*0.3797*	*-42.2735*
		$i = 2$	*0.1551*	*0.2273*	*149.2051*			
		$i = 3$	*0.2657*	*0.3344*	*225.1918*			
		$i = 4$	*0.1543*	*0.9523*	*119.6312*			
0	*1*	$i = 1$	*0.0551*	*0.7141*	*86.7590*	*-0.3410*	*0.3268*	*82.4673*
		$i = 2$	*0.2673*	*0.6326*	*263.4853*			
		$i = 3$	*0.0022*	*0.5880*	*305.5785*			
		$i = 4$	*0.1113*	*0.8119*	*182.0026*			
100	*1*	$i = 1$	*0.1664*	*0.7217*	*128.0523*	*-0.6472*	*0.0355*	*4.6730*
		$i = 2$	*0.0124*	*0.3685*	*123.1282*			
		$i = 3$	*0.2710*	*0.5283*	*257.5323*			
		$i = 4$	*0.0412*	*0.6232*	*125.9760*			
10	*1*	$i = 1$	*0.1001*	*0.9812*	*254.6646*	*0.2374*	*-0.1315*	*-40.7649*
		$i = 2$	*0.3339*	*0.9135*	*239.7476*			
		$i = 3$	*0.0049*	*0.2885*	*252.0848*			
		$i = 4$	*0.0279*	*0.8537*	*250.0119*			
1	*1*	$i = 1$	*0.2125*	*1.0327*	*141.6154*	*-0.2668*	*-0.0537*	*-24.0118*
		$i = 2$	*0.0104*	*0.9319*	*159.7887*			
		$i = 3$	*0.2587*	*0.5944*	*216.8422*			
		$i = 4$	*0.0742*	*0.8721*	*301.0206*			
1	*10*	$i = 1$	*0.0408*	*0.5829*	*304.0465*	*-1.3398*	*-0.4620*	*7.6814*
		$i = 2$	*0.0016*	*0.2805*	*179.2660*			
		$i = 3$	*0.2604*	*0.3010*	*237.1790*			
		$i = 4$	*0.3144*	*1.0142*	*255.2613*			
1	*100*	$i = 1$	*0.0707*	*0.8195*	*291.0683*	*-0.8912*	*-0.2073*	*16.6305*
		$i = 2$	*0.1941*	*0.2101*	*121.2359*			
		$i = 3$	*0.0583*	*0.6379*	*305.7286*			
		$i = 4$	*0.1306*	*0.4602*	*258.0886*			

Table 5.17: Experiment 6 - Random initial control inputs under different weights

see that, when the FWMAV model run for 50 pattern duration (7.0925 seconds), the altitude drops *9.7041E-04* meters. It is a lot better than the one in experiment 6.

ϖ_{Rpat}	ϖ_{Peff}	$\overline{P_{act}}$ (W)	$\overline{P_{eff}}$ (W)	$\sum {R_{pattern}}^2$
1	0	1.4701	2.128	4.5894E-12
0	1	6.32E-07	5.12E-08	2.52E+04
100	1	4.88E-06	7.56E-07	0.6145
10	1	1.38E-04	4.18E-05	0.6136
1	1	0.2026	0.2645	0.0368
1	10	0.2873	0.3920	7.61E-04
1	100	0.3058	0.4195	8.38E-06

Table 5.18: Experiment 6 - The power consumption and residual

Battery model	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
w/o	$i = 1$	0.0000	0.8995	313.9932	0.9593	-0.0134	16.8379
	$i = 2$	0.0748	0.3861	78.4983			
	$i = 3$	0.0000	0.5415	314.0499			
	$i = 4$	0.0000	0.7771	314.0471			
w	$i = 1$	0.0000	0.9605	314.1588	0.9698	-0.0176	16.1347
	$i = 2$	0.0734	0.3997	76.7651			
	$i = 3$	0.0000	0.5548	314.1591			
	$i = 4$	0.0000	0.8161	314.1591			

Table 5.19: Experiment 7 and 8 - Control inputs

Battery model	*cycle(i)*	U_i	α_{mi} *(rad)*	ω_i *(rad/s)*	$\varphi(0)$ *(rad)*	*w(0) (m/s)*	$\dot{\varphi}(0)$ *(rad/s)*
w/o	$i = 1$	0.1859	0.8502	305.7548	0.3063	-0.2420	-20.4034
	$i = 2$	0.0764	0.7416	101.2145			
	$i = 3$	0.2362	0.3483	194.4925			
	$i = 4$	0.3559	0.2812	256.5129			
w	$i = 1$	0.1917	0.6399	281.3580	0.2658	0.2530	-30.5192
	$i = 2$	0.0895	0.4221	89.1141			
	$i = 3$	0.1444	0.2402	222.9571			
	$i = 4$	0.2140	0.5661	240.9930			

Table 5.20: Experiment 7 and 8 - Random initial control inputs

Battery model	$\overline{P_{act}}$ *(W)*	$\overline{P_{eff}}$ *(W)*	$\sum {R_{pattern}}^2$	δz *(m)*	$\delta\phi$ *(rad)*	δw *(m/s)*	$\delta\dot{\phi}$ *(rad/s)*
w/o	0.3652	0.5315	3.09E-27	5.13E-16	-1.34E-14	8.13E-15	-5.33E-14
w	0.3683	0.5285	3.71E-26	1.88E-15	6.77E-15	1.71E-14	1.92E-13

Table 5.21: Experiment 7 and 8 - The power consumption and residual

Compare the results of experiment 7 and 8 with all previous experiments, we proved that the methodology of making the residual as a equality constraint can introduce best solutions with low power consumption along with a more periodic and hovering orbit. Especially, compared with the power consumption results of experiment 1 and 2 of the static flight stage in table 5.3 and 5.6, the power consumption improvement is larger than *40%* while keeping orbits periodic and hovering.

Number of patterns	*Duration (s)*	*δz (m)*
1	*0.1418*	*7.3393E-06*
5	*0.7092*	*1.1815E-04*
50	*7.0925*	*9.7041E-04*
500	*70.9248*	*0.0274*
5000	*709.2476*	*0.3331*

Table 5.22: Experiment 8 - The displacements

Chapter 6

Conclusion and Future Work

This chapter summarizes the findings from chapter 5 and gives recommendations for further research.

The balance and stability analysis for FWMAV systems is always quite challenging, while the flight duration is also becoming a big challenge. In this thesis, we analyzed the balance and stability of FWMAV system while considering the battery model for best energy usage. The residual described in chapter 2 is chosen to represent the balance and stability of the FWMAV system. In chapter 5, the experimental results of the flight control methodology described in chapter 4 are discussed.

There are two stages of the FWMAV project are covered in this thesis: static flight and dynamic flight. In the experimental results for the static flight, by adjusting the weights for power consumption and the residual in the cost function, we can see clear tradeoff between the power consumption and the residual. The effection of including battery model into the optimization is also reflected by the experimental results. When the battery model is considered, the effective power consumption is used in the cost function. Compared with the case that the battery model is not considered, the optimizer is able to find orbits with

lower power consumption. The results of the static flight stage also prove that the static vertical flight optimization doesn't have much flexibility because of the strict constraints of hovering and periodic flight. Hence, the power consumption isn't improved much when we increase the weight for the power consumption in the cost function. This problem is solved by the dynamic flight control. By relaxing the strict constraints of hovering and periodic flight, the dynamic vertical flight optimization gains more flexibility when finding the orbits. Compared with the power consumption results of the static flight stage, the power consumption improvement is even larger than *40%* while keeping orbits periodic and hovering.

There are still many possible further research which can be exploited in the future. In work [26] and [41], dynamic soaring is described by the authors. The transfer of energy from the moving air in the shear wind to a bird is considered as an energy source for dynamic soaring. The energy gain from the moving air is just used to compensate for the energy loss due to drag for a dynamic soaring cycle. It is also possible for the FWMAV to take advantage of the energy of the moving air in the wind to save the energy for flying and hovering.

The other possible energy source is the solar energy. The team of University of Maryland has had the flexible solar cells integrated into wings, tail, and body of FWAVs to harvest solar energy [22, 21]. The harvested solar energy is used to recharge batteries and increase the flight time of the FWMAV by supplementing the battery power. It also eliminates the need for external electrical power. Considering the solar energy, size of solar cells and wing area in our flight control, it is possible that the joint optimization could find the orbit and wing design which can maximize the flight endurance while keeping orbit hovering and periodic.

Bibliography

[1] M. A. Al Faruque and K. Vatanparvar. Modeling, analysis, and optimization of electric vehicle hvac systems. In *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*, pages 423–428. IEEE, 2016.

[2] Atomic Workshop. Battery specification. `http://www.atomicworkshop.co.uk/`, 2018.

[3] P. Chirarattananon and R. J. Wood. Identification of flight aerodynamics for flapping-wing microrobots. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1389–1396. IEEE, 2013.

[4] C. De Wagter, S. Tijmons, B. D. Remes, and G. C. de Croon. Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4982–4987. IEEE, 2014.

[5] G. Decroon, M. Percin, B. Remes, R. Ruijsink, and C. De Wagter. *The delfly: Design, aerodynamics, and artificial intelligence of a flapping wing robot*. 01 2015.

[6] W. Dednam and A. E. Botha. Optimized shooting method for finding periodic orbits of nonlinear dynamical systems. *Engineering with Computers*, 31(4):749–762, 2015.

[7] S. Deng, B. Van Oudheusden, B. Remes, M. Persin, H. Bijl, and H. Ruijsink. Experimental investigation of the flapping performance on'delfly micro'. 2013.

[8] D. B. Doman, M. W. Oppenheimer, and D. O. Sigthorsson. Wingbeat shape modulation for flapping-wing micro-air-vehicle control during hover. *Journal of guidance, control, and dynamics*, 33(3):724–739, 2010.

[9] H. Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, pages 1–15, 2011.

[10] J. Gerdes, A. Holness, A. Perez-Rosado, L. Roberts, A. Greisinger, E. Barnett, J. Kempny, D. Lingam, C.-H. Yeh, H. A. Bruck, et al. Robo raven: a flapping-wing air vehicle with highly compliant and independently controlled wings. *Soft Robotics*, 1(4):275–288, 2014.

[11] R. Hainsworth and L. Wolf. Hummingbird feeding. *Wildbird Magazine*, 1993.

[12] A. M. Hassan and H. E. Taha. Higher-order averaging analysis of the nonlinear time-periodic dynamics of hovering insects/flapping-wing micro-air-vehicles. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 7477–7482. IEEE, 2016.

[13] A. M. Hassan and H. E. Taha. A combined averaging-shooting approach for the trim analysis of hovering insects/flapping-wing micro-air-vehicles. In *AIAA Guidance, Navigation, and Control Conference*, page 1734, 2017.

[14] T. J. Kazmierski and S. Beeby. *Energy harvesting systems*. Springer, 2014.

[15] D. Lentink, S. R. Jongerius, and N. L. Bradshaw. The scalable design of flapping micro-air vehicles inspired by insect flight. In *Flying insects and robots*, pages 185–205. Springer, 2009.

[16] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340(6132):603–607, 2013.

[17] MathWorks. MATLAB. `https://www.mathworks.com/`, 2018.

[18] MAVLab at Delft University of Technology. DelFly series. `http://www.delfly.nl/`, 2018.

[19] N. O. Pérez-Arancibia, P.-E. J. Duhamel, K. Y. Ma, and R. J. Wood. Model-free control of a hovering flapping-wing microrobot. *Journal of Intelligent & Robotic Systems*, 77(1):95–111, 2015.

[20] N. O. Pérez-Arancibia, J. P. Whitney, and R. J. Wood. Lift force control of flapping-wing microrobots using adaptive feedforward schemes. *IEEE/ASME Transactions on Mechatronics*, 18(1):155–168, 2013.

[21] A. Perez-Rosado, H. A. Bruck, and S. K. Gupta. Enhancing the design of solar-powered flapping wing air vehicles using multifunctional structural components. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V05BT08A016–V05BT08A016. American Society of Mechanical Engineers, 2015.

[22] A. Perez-Rosado, H. A. Bruck, and S. K. Gupta. Integrating solar cells into flapping wing air vehicles for enhanced flight endurance. *Journal of Mechanisms and Robotics*, 8(5):051006, 2016.

[23] A. Perez-Rosado, R. D. Gehlhar, S. Nolen, S. K. Gupta, and H. A. Bruck. Design, fabrication, and characterization of multifunctional wings to harvest solar energy in flapping wing air vehicles. *Smart Materials and Structures*, 24(6):065042, 2015.

[24] L. Petricca, P. Ohlckers, and C. Grinde. Micro-and nano-air vehicles: State of the art. *International journal of aerospace engineering*, 2011, 2011.

[25] L. Roberts, H. A. Bruck, and S. K. Gupta. Autonomous loitering control for a flapping wing miniature aerial vehicle with independent wing control. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V05AT08A013–V05AT08A013. American Society of Mechanical Engineers, 2014.

[26] G. Sachs. Minimum shear wind strength required for dynamic soaring of albatrosses. *Ibis*, 147(1):1–10, 2005.

[27] H. E. Taha. A geometric control approach for the longitudinal flight stability of hovering insects/fwmavs. In *AIAA Guidance, Navigation, and Control Conference*, page 1552, 2015.

[28] H. E. Taha, S. Tahmasian, C. A. Woolsey, A. H. Nayfeh, and M. R. Hajj. The need for higher-order averaging in the stability analysis of hovering, flapping-wing flight. *Bioinspiration & biomimetics*, 10(1):016002, 2015.

[29] H. E. Taha, C. A. Woolsey, and M. R. Hajj. Geometric control approach to longitudinal stability of flapping flight. *Journal of Guidance, Control, and Dynamics*, 39(2):214–226, 2015.

[30] Z. E. Teoh and R. J. Wood. A flapping-wing microrobot with a differential angle-of-attack mechanism. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1381–1388. IEEE, 2013.

[31] K. Vatanparvar and M. A. Al Faruque. Eco-friendly automotive climate control and navigation system for electric vehicles. In *Cyber-Physical Systems (ICCPS), 2016 ACM/IEEE 7th International Conference on*, pages 1–10. IEEE, 2016.

[32] K. Vatanparvar and M. A. Al Faruque. Acqua: Adaptive and cooperative quality-aware control for automotive cyber-physical systems. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017.

[33] K. Vatanparvar and M. A. Al Faruque. Path to eco-driving: Electric vehicle hvac and route joint optimization. *IEEE Design & Test*, 2017.

[34] K. Vatanparvar, S. Faezi, I. Burago, M. Levorato, and M. A. Al Faruque. Extended range electric vehicle with driving behavior estimation in energy management. In *IEEE transactions on Smart Grid*, 2018.

[35] K. Vatanparvar, A. Faruque, and M. Abdullah. Battery lifetime-aware automotive climate control for electric vehicles. In *Proceedings of the 52nd Annual Design Automation Conference*, page 37. ACM, 2015.

[36] K. Vatanparvar, A. Faruque, and M. Abdullah. Otem: Optimized thermal and energy management for hybrid electrical energy storage in electric vehicles. In *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*, pages 19–24. EDA Consortium, 2016.

[37] K. Vatanparvar and M. A. A. Faruque. Electric vehicle optimized charge and drive management. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 23(1):3, 2017.

[38] K. Vatanparvar, J. Wan, and M. A. Al Faruque. Battery-aware energy-optimal electric vehicle driving management. In *Low Power Electronics and Design (ISLPED), 2015 IEEE/ACM International Symposium on*, pages 353–358. IEEE, 2015.

[39] J. Zhang, B. Cheng, B. Yao, and X. Deng. Adaptive robust wing trajectory control and force generation of flapping wing mav. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5852–5857. IEEE, 2015.

[40] J. Zhang, Z. Tu, F. Fei, and X. Deng. Geometric flight control of a hovering robotic hummingbird. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5415–5421. IEEE, 2017.

[41] Y. J. Zhao. Optimal patterns of glider dynamic soaring. *Optimal control applications and methods*, 25(2):67–89, 2004.