A Bio-inspired Sensing Approach to in-Gust Flight of Flapping Wing MAVs

Master Thesis

Chenyao Wang

Delft University of Technology

TUDelft

A Bio-inspired Sensing Approach to in-Gust Flight of Flapping Wing MAVs

by

Chenyao Wang

to obtain the degree of Master of Science,
at the Delft University of Technology,
to be defended publicly on Tuesday the 27^{th} of September, 2022 at 09:30.

Student Number:	4920104	
Project Duration:	December, 2021 - September, 2022	
Thesis Committee:	Prof. Dr. G. C. H. E. de Croon	TU Delft, committee chairman
	Prof. Dr. S. Hamaza	TU Delft, daily supervisor
	Prof. Dr. B.W. van Oudheusden	TU Delft, examiner
	Dr. ir. C. de Wagter	TU Delft, additional member
	Ir. Sunyi Wang	TU Delft, additional member

TU Delft

Contents

Nomenclature

Symbols

x, y, z	Position in the global frame $[m]$
ϕ, ψ, θ	Attitude angles $[deg]$
V_x, V_y, V_z	Velocities in the body frame $[m/s]$
a_x, a_y, a_z	Linear accelerations in the body frame $[m/s^2]$
p, q, r	Angular velocities in the body frame $[deg/s]$
V_{air}	Airflow sensing voltage from the airflow sensor
I	Current Intensity $[A]$
$\gamma_{command}$	Dihedral angle command $[deg]$
γ_{output}	Dihedral angle output $[deg]$
$V_{battery}$	Battery voltage $[V]$
X_G, Y_G, Z_G	Global frame
$X_{body}, Y_{body}, Z_{body}$	Body-fixed frame
$X_{inertial}, Y_{inertial}, Z_{inertial}$	Inertial frame
$\ddot{x}, \ddot{z}$	Linear acceleration along the $X_{inertial}$ and $Z_{inertial}$ axis
m	Mass $[Kg]$
g	Gravity $[m/s^2]$
T	Thrust force $[N]$
$F_{D_{wind}}$	Wind drag force $[N]$
K_{wind}	Identified effectiveness reduction coefficient
V_{wind}	Wind speed $[m/s]$
C_{corr}	Correction term for a better fit
$C_{D_{wind}}$	Wind drag coefficient
f_{flap}	Flapping frequency $[Hz]$
f	Flapping frequency $[Hz]$
K_{flap}	Conversion coefficient from PWM to flapping frequency
c_1, c_2	Fitted coefficients
x_{ref}	Reference position along X_{body} axis $[m]$
$x_{measured}$	Measured position along X_{body} axis $[m]$
e_x	Position error along X_{body} axis $[m]$
$K_{P_x}, K_{I_x}, K_{D_x}$	PID gains in the position controller
K_{FF_x}	Feedforward gain in the velocity controller
θ_{FF}	Pitch angle calculated from the feedforward block in the velocity controller $[deg]$
len	Horizon length for the gain selector
s_i	Current states of airflow readings
S_i	Length of time of the states in the gain selector remain unchanged
f_{gust}	Gust alternating frequency $[Hz]$
$RMSE_x$	Root Mean Square Error along the $X_{inertial}$ axis $[m]$
$RMSE_z$	Root Mean Square Error along the $Z_{inertial}$ axis $[m]$
$RMSE_\theta$	Root Mean Square Error of the pitch angle $[deg]$
U	Constant wing velocity
Φ	Flapping amplitude $[rad]$
b	Wingspan $[m]$
u	Free-stream velocity $[m/s]$
β	Force coefficients
$R_d^T R$	Attitude error
$e_{\omega b}$	Angular velocity tracking error
e_R	Attitude tracking error

Acronyms

AHRS	Attitude and Heading Reference System
AoA	Angle of Attack
CA	Classification Algorithm
CFD	Computational Fluid Dynamics
CoM	Center of Mass
CoP	Center of Pressure
DDGP	Deep Deterministic Policy Gradient
DOBC	Disturbance Observer Based Control
DoF	Degree of Freedom
EKF	Extended Kalman Filter
EoM	Equation of Motion
FF	Feed Forward
FWMAV	Flapping Wing Micro Air Vehicle
IMU	Inertial Measure Unit
LEV	Leading Edge Vortex
MAV	Micro Air Vehicle
ML	Machine Learning
NN	Neural Network
OLS	Ordinary Least Square
PG	Policy Gradient
PID	Proportional–Integral–Derivative (controller)
PWM	Pulse Width Modulation
RL	Reinforcement Learning
RMS	Root Mean Square
RMSE	Root Mean Square Error
SCM	Smart Composite Microstructures

List of Figures

List of Tables

1

Introduction

Starting from *Pterodactyls*, flying animals have been dominating sky for thousands of million years. Relying on their flapping wings to generate control forces, they are capable of hovering and agile maneuvers. Due to the similarity in the size, therefore, flying insects [45] and hummingbirds [50, 22] have inspired the design and development of flapping-wing micro air vehicles (FWMAVs) greatly. During the last years, researchers have developed several FWMAV platforms: *RoboBee* by Harvard University [45], *KU Beetle* [32], *Flapping Hummingbird* by Purdue University [42], *DelFly Nimble* by TU Delft [22] and the *GRIFFIN ornithopter* [27]. With weights from 80 milligrams [45] to 28.2 grams [22] and one or two pairs of wings, these platforms possess different and outstanding flight capabilities, such as: rapid banked turns of fruit flies [22], extreme evasive maneuvers of hummingbirds [14] and ling-endurance flights [33].

Regarding the flight control of micro air vehicles (MAVs), besides the vision based obstacle avoidance [10, 49], in-gust flight control and disturbance rejection for MAVs has always been a problem, particularly because aerodynamic forces resulting from gusts have a higher impact on flight stability on lightweight MAVs than multirotors. On the topic of gust disturbance rejection for quadrotors, [48] has proposed a dual closed-loop control framework with an extended state observer and active disturbance rejection control in the inner loop for attitude control. In [30], a deep learning–based trajectory tracking controller is implemented, which enables the quadrotor to learn how to quickly adapt to rapidly changing wind conditions. Furthermore, for fixed-wing type-of vehicles, [5] has introduced a bio-inspired gust rejection mechanism based on strain sensing feedback to improve the roll control of in-gust flights, which is inspired by the campaniform sensilla, the load and strain sensors of insects in their wings [39].

As a subset of MAVs, FWMAVs are of interest in recent years for their advanced maneuverability and agility, inspired by their biological counterparts [22]. However, they are more vulnerable to external disturbances such as gusts, due to unsteady aerodynamics of flapping wing flights [18]. So far, a few attempts to study and develop disturbance rejection methods for in-gust flight of FWMAVs have been introduced. With aim of disturbance rejection for $0.6\ m/s$ horizontal wind, both adaptive estimation and least square estimation methods are employed for RoboBee control [9]. [24] presented disturbance observer based control (DOBC) was also applicable for FWMAVs' disturbance rejection. A gain-scheduling control approach in the horizontal position controller enabled by onboard airflow sensing was introduced and implemented for in-wind flights of DelFly Nimble [43]. All these previous work present adequate solutions for their pre-defined wind conditions, however in nature, flying insects adjust their flight attitude and behaviors by sensing the gusts with their antennae, and then acting accordingly [16, 39].

In this thesis project, we introduce a bio-inspired sensing approach to in-gust flight for FWMAVs, comprising on-board airflow sensing and an adaptive PID and feedforward gain scheduling approach for gust disturbance rejection. Our main contribution here is the adaptive position and velocity control framework utilizing a thermistor-based airflow sensor [43]. With this adaptive position and velocity controller, the FWMAV could achieve a better stability for FWMAVs' in-gust flights with a similar energy consumption level as the original PID controller. Firstly, we focus on the modeling of FWMAVs' in-gust dynamics and capture the main effects brought by gusts on FWMAVs, which are wind drag forces and servo control effectiveness reduction. Then, we propose an airflow sensing based adaptive flight controller and validate it by stabilizing the FWMAV's in-gust hovering. Finally, multiple in-gust flight experiments are carried out with a time-varying wind intensity for validating the adaptive position and velocity controller, and their results are analyzed and compared with original cases.

The reminder of this report is organized as two Chapters. In Chapter 2, the main results of this thesis project are shown as a scientific paper. In Chapter 3, the literature study is shown in five chapters. In Section 3.1, several FWMAV platforms which has research progress in recent years are introduced. The aerodynamics and the in-gust dynamics of FWMAVs are explained in Section 3.2. In Section 3.3, the existing models of FWMAVs' dynamics are presented. Furthermore, in Section 3.4, classic, machine learning (ML) and reinforcement learning (RL) based control for FWMAVs are introduced. In Appendix A, the schematics of the connection of the extra hardware to the Flapper Drone's bolt are shown, while Appendix B presents the previous design of the adaptive feedforward velocity controller and its performance in several in-gust hovering flights.

2

Scientific Paper

A Bio-inspired Sensing Approach to in-Gust Flight of Flapping Wing Micro Air Vehicles

Chenyao Wang [1,*], **Sunyi Wang** [1,2], **Guido de Croon** [1], **Salua Hamaza** [1,2]

[1] *Micro Air Vehicle Laboratory, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands*
[2] *BioMorphic Intelligence Laboratory, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands*

Correspondence*:
Chenyao Wang
C.Wang-19@student.tudelft.nl

ABSTRACT

Flapping wing micro aerial vehicles (FWMAVs) are known for their flight agility and maneuverability. However, their in-gust flight performance and stability is still inferior to their biological counterparts. To this end, a simplified in-gust dynamic model, which could capture the main gust effects on FWMAVs, has been identified with real in-gust flights' data of a FWMAV, the Flapper Drone. Based on this model, an adaptive position and velocity controller was proposed with gain scheduling and implemented for in-gust flights under gust speeds up to $2.4\ m/s$. With this airflow-sensing based adaptive controller, the in-gust hovering stability of the Flapper Drone has been improved when the gust's intensity and frequency changes, comparing with the original fixed-gain cascaded PID controller case.

Keywords: flapping wing, bio-inspired aerial robotics, modeling, adaptive PID control, in-gust flights, onboard airflow sensing

1 INTRODUCTION

In-gust flight control and disturbance rejection for micro air vehicles (MAVs) has always been a problem, particularly because aerodynamic forces resulting from gusts have a higher impact on flight stability on lightweight MAVs than multirotors. On the topic of gust disturbance rejection for quadrotors, (Yang et al., 2017) has proposed a dual closed-loop control framework with an extended state observer and active disturbance rejection control in the inner loop for attitude control. In (O'Connell et al., 2022), a deep learning–based trajectory tracking controller is implemented, which enables the quadrotor to learn how to quickly adapt to rapidly changing wind conditions. Furthermore, for fixed-wing type-of vehicles, (Castano et al., 2014) has introduced a bio-inspired gust rejection mechanism based on strain sensing feedback to improve the roll control of in-gust flights, which is inspired by the campaniform sensilla, the load and strain sensors of insects in their wings Skordos et al. (2002).

As a subset of MAVs, flapping wing (FWAMVs) are of interest in recent years for their advanced maneuverability and agility, inspired by their biological counterparts (Karásek et al., 2018). However, they are more vulnerable to external disturbances such as gusts, due to unsteady aerodynamics of flapping wing flights (Ho et al., 2003). So far, a few attempts to study and develop disturbance rejection methods for in-gust flight of FWMAVs have been introduced. With aim of disturbance rejection for $0.6\ m/s$ horizontal wind, both adaptive estimation and least square estimation methods are employed for RoboBee control (Chirarattananon et al., 2015). (Lee et al., 2020) presented disturbance oberserver based control (DOBC) was also applicable for FWMAVs' disturbance rejection. A gain-scheduling control approach in the horizontal position controller enabled by onboard airflow sensing was introduced and implemented for in-wind flights of Delfly Nimble (Wang et al., 2022). All these previous work present adequate solutions for their pre-defined wind conditions, however in nature, flying insects adjust their flight attitude and behaviors by sensing the gusts with their antennae, and then acting accordingly (Fuller et al., 2014).

In this paper we introduce a bio-inspired sensing approach to in-gust flight for FWMAVs, comprising onboard airflow sensing and an adaptive PID and feedforward gain scheduling approach for gust disturbance rejection. Our main contribution here is the adaptive position and velocity control framework utilizing a thermistor-based airflow sensor (Wang et al., 2022). With this adaptive position and velocity controller, the FWMAV could achieve a better stability for FWMAVs' in-gust flights with a similar energy consumption level as the original PID controller. Firstly, we focus on the modeling of FWMAVs' in-gust dynamics and capture the main effects brought by gusts on FWMAVs, which are wind drag forces and servo control effectiveness reduction. Then, we propose an airflow sensing based adaptive flight controller and validate it by stabilizing the FWMAV's in-gust hovering. Finally, multiple in-gust flight experiments are carried out with a time-varying wind intensity for validating the adaptive position and velocity controller, and their results are analyzed and compared with original cases.

2 EXPERIMENTS SETUP

2.1 Aerial platform and Fan system

The FWMAV platform used here is Flapper Drone [1] which is developed based on Crazyflie STM32F405 platform. Comparing with DelFly Nimble which has a similar structure and design (Karásek et al., 2018), the dimensions of Flapper Drone are larger than that of DelFly Nimble, hence resulting in a heavier weight and a much larger maximum payload, as shown in Table 1. To generate continuous winds and dynamic gusts during experimental flights, a fan system consisting of an array of 135 axial fans (Olejnik et al., 2022) is employed. With pulse width modulation (PWM) based control, the fan system could generate various types of continuous winds and gusts with a total wind surface of approximately $1.00\ m^2$, which renders that the experimental flights could be conducted under continuous wind and gust conditions of different intensities and frequencies.

Parameters	DelFly Nimble	Flapper Drone
Wingspan [mm]	330	490
Weight [g]	29.85	112.17
Maximum payloads [g]	10.70	25.00
Battery capacity [mAh]	180	300

Table 1. DelFly Nimble and Flapper Drone parameters

[1] https://flapper-drones.com/wp/

2.2 Experiments set-up

All of the experimental flights are carried out in CyberZoo [2]. As shown in Figure 1, the Flapper Drone has been set to hover at a fixed pre-defined setpoint ($1.0m$ downstream from the fan surface of the fan system, $1.1m$ above the ground and the negative direction of body frame axis X_b pointing towards the center of the fan system) during the hovering flight experiments in front of the fan system, which ensures that it could be fully immersed in the freestream created by the fan system and hover right in the center of the flow.

For the guided position control flights, the OptiTrack motion capture system measures the accurate position of the Flapper Drone and feedbacks accurate position and attitude measurements to the flight control system. The in-flight data has been collected from different sources, as presented in Table 1. To generate repeatable gust disturbances of different intensities and frequencies for each flight experiment, the fan system PWM duty cycle has been programmed to stay rest for the first ten seconds, and alternate between a low wind speed value $0.5\ m/s$ (PWM duty cycle = 20%) and a high wind speed value $2.1\ m/s$ (PWM duty cycle = 60%) or $2.4\ m/s$ (PWM duty cycle = 70%) (Olejnik et al., 2022) at a pre-defined fixed frequency during the following 30 seconds.

Type sensor	Measurements obtained
Optitrack	Position (x, y, z) Attitude angles (ϕ, ψ, θ)
IMU AHRS	Velocities (V_x, V_y, V_z) Linear accelerations (a_x, a_y, a_z) Angular velocities (p, q, r)
RevP airflow sensor	Airflow sensing voltage (V_{air})
Current sensor	Current intensity (I)
On-board extra	Dihedral angle command $(\gamma_{command})$ Dihedral angle output (γ_{output}) Battery voltage $(V_{battery})$

Table 2. Overview of data obtained from Optitrack motion capture system and other on-board sensors.

3 THE MODELING OF THE FWMAV'S IN-GUST DYNAMICS

3.1 Model Structure

In several previous studies, the dynamics of FWMAV have been modelled and used for controller design. In (Kajak et al., 2019), a minimal longitudinal model has been proposed for controller design. In (Nijboer et al., 2020), a grey-box longitudinal dynamics model is derived based on free-flight data. Furthermore, in (Bains, 2020), the lateral body dynamics has been modeled with system identification approach. However, these models were derived mostly based on free-flight data with no external disturbance, hence couldn't capture the effects of gusts on FWMAV's system dynamics precisely.

[2] A flight arena located at the Faculty of Aerospace Engineering, TU Delft

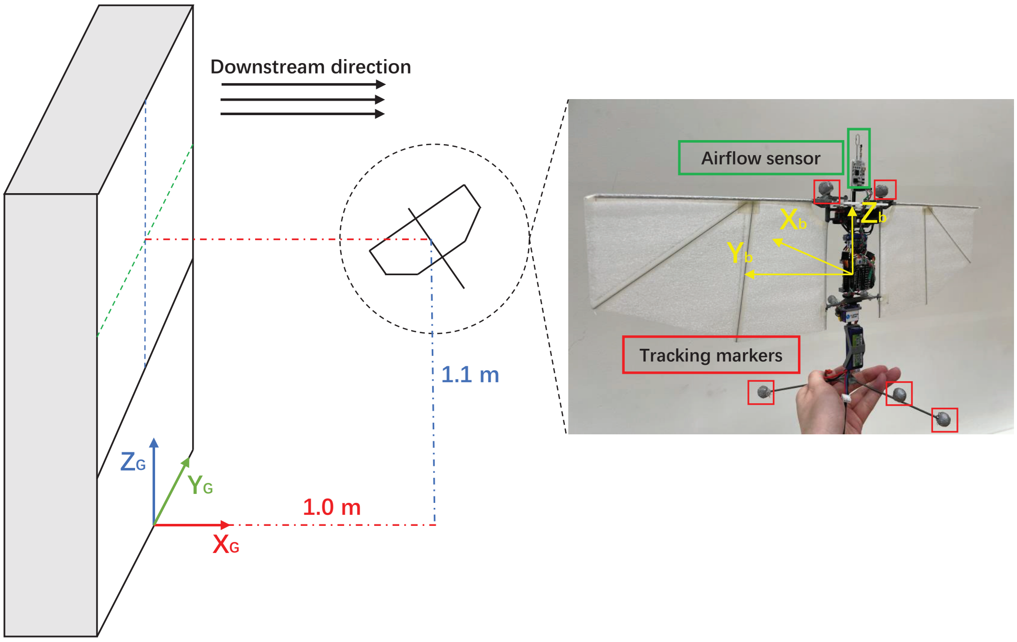


Figure 1. Flapper Drone hovering at a fixed pre-defined setpoint in front of the fan system, with the configuration of the airflow sensor illustrated. Reflective markers are added and used for the guided position control flights with Optitrack motion capture system. The global coordinate frame originates from the intersection point between the mid-line of the fan system and the ground plane.

The dynamics of FWMAV's in-wind flights have also been studied in the past few years. (Chirarattananon et al., 2017) presented an in-wind FWMAV dynamics model consists of equations of motion (EoM) and an additional vector τ_w describing the wind effects on FWMAV. In (Lee et al., 2020), the attitude dynamics of FWMAV has been modeled by modeling the moments acting on a flying FWMAV. Nevertheless, in these models, the wind effects on servo control effectiveness have not been modeled.

The model presented here is a simplified physical model focusing on in-wind longitudinal and translational dynamics, as shown in Equation 1 and Equation 2.

$$m\ddot{x} = Tsin\theta + F_{D_{wind}} \tag{1}$$

$$m\ddot{z} = Tcos\theta - mg \tag{2}$$

where m is the mass, T is the thrust generated by the pair of the flapping wings, $F_{D_{wind}}$ is the wind drag force and θ is the body pitch angle.

The free body diagram of this model has been shown in Figure 2. Both the thrust force T and the wind drag force $F_{D_{wind}}$ are acting directly on the center of mass (CoM) of the FWMAV. The pitch angle θ of FWMAV, which is controlled through the dihedral servo placed near the top of the body fuselage, is represented by the angle between the Z_{body} axis and the $Z_{inertial}$ axis. To include the wind effects on the FWMAV in this model, both in-gust actuator dynamics of the dihedral servo and the wind drag model have been modeled in Section 3.2 and Section 3.3 respectively. Furthermore, the thrust generated by the FWMAV is also modeled and presented in Section 3.4.

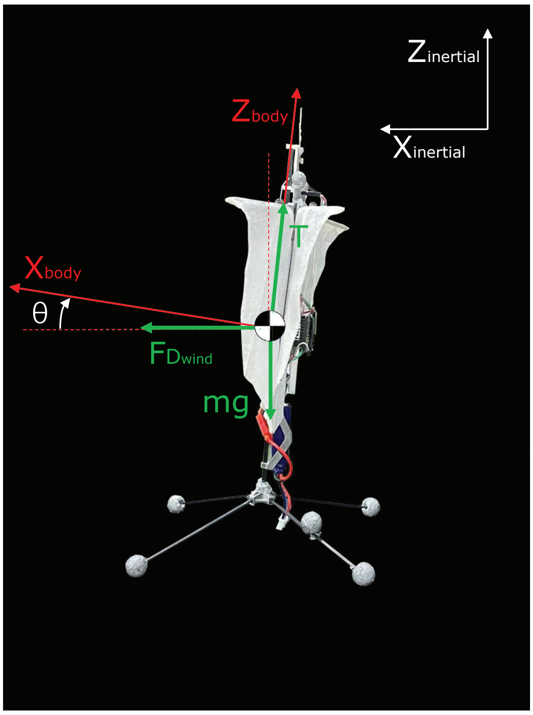


Figure 2. 2D longitudinal free body diagram of a FWMAV (side view). CoM represents the center of mass.

3.2 The Effects of Wind on Dihedral Servo Control Effectiveness

In (Kajak et al., 2019), the effects of forward flights on the dihedral servo of DelFly Nimble have been shown. Similarly as shown in Figure 3, when hovering under the influence of wind disturbance, the dihedral servo of Flapper Drone has also encountered control effectiveness reduction, which means that the actual output dihedral angle is always smaller than the desired dihedral angle due to the load imposed on the wings by gusts. To model this effect, a dihedral servo control effectiveness model has been proposed as Equation 3, where K_{wind} is the identified effectiveness reduction coefficient, V_{wind} the wind speed, γ_{output} the actual dihedral angle output, $\gamma_{command}$ the commanded dihedral angle and C_{corr} the correction term calculated with real-time airflow sensor reading aiming for a better fit.

$$\gamma_{output} = K_{wind}(V_{wind})\;\gamma_{command} + sign(V_{wind})C_{corr}(V_{wind}) \tag{3}$$

To model the dihedral servo control effectiveness reduction, several hovering tests have been conducted with the fan system's continuous wind speed setting between $0.5\ m/s$ and $2.7\ m/s$. The stable hovering positions and pitch angles of the Flapper Drone under different wind intensities are shown in Figure 4. When the wind speed is increasing, the FWMAV moves further downstream with an altitude increase during the stable hovering phase. Based on Equation 3, K_{wind} has been identified with ordinary least square (OLS) estimator by polyval() and polyfit() functions from MATLAB. As presented, during in-wind hovering, the dihedral servo could not follow the change of commands precisely, thus the output dihedral angles are always smaller than the commanded inputs. Theoretically, K_{wind} should decrease since the control effectiveness reduction is more severe when the wind speed is increasing Kajak et al. (2019). Though K_{wind} in Figure 4 has not always shown a clear trend, it is decreasing when the stable hovering positions in X_G axis are similarly large, such as the cases when continuous wind speed = $0.5\ m/s$ to $1.7\ m/s$ and

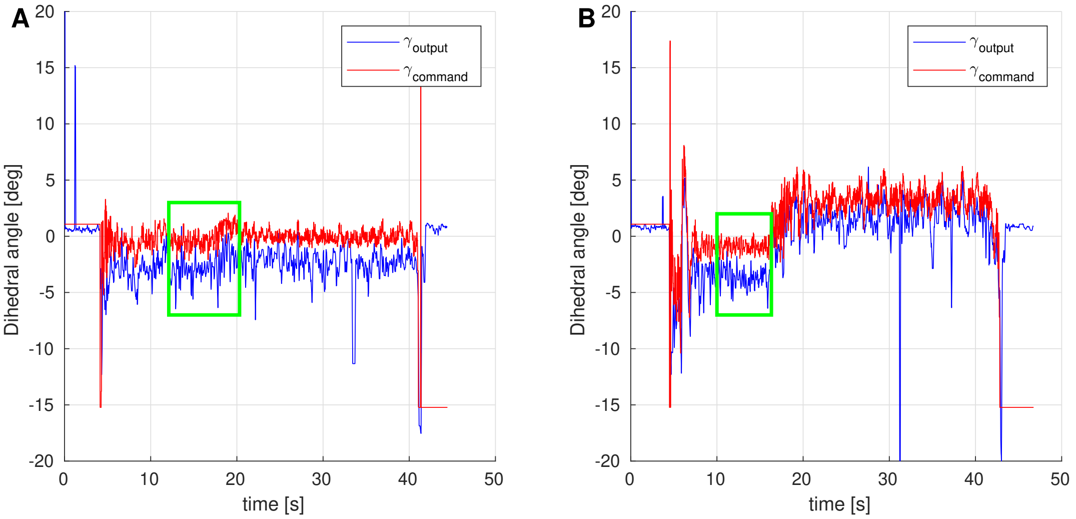


Figure 3. Time histories of the actual dihedral angle output γ_{output} (red lines) and the commanded dihedral angle $\gamma_{command}$ (blue lines) when hovering in wind stably under continuous wind speed = $0.9\ m/s$ (A) and $2.1\ m/s$ (B). The data within the green rectangular is selected for the dihedral servo control effectiveness model identification.

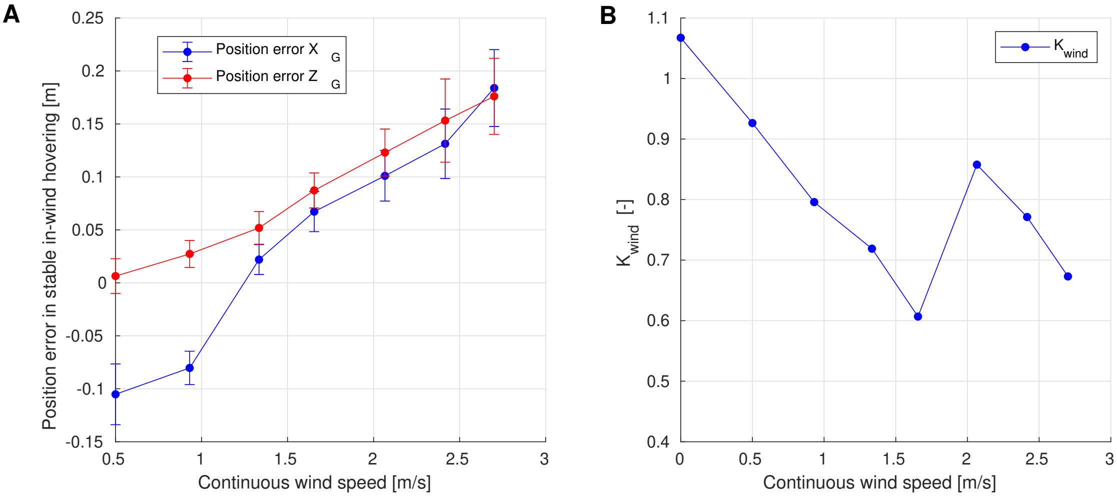


Figure 4. A: Average position errors in X_G and Z_G axis when hovering stably under different continuous wind speeds. B: Values of K_{wind} under different continuous wind speeds.

the cases when continuous wind speed = $2.1\ m/s$ to $2.7\ m/s$. This could be explained by that when the FWMAV is further away from the Fan System, the downstream flow field quality will always degrade though the wind speed has been set as a same value.

Therefore, referring to the dihedral servo control effectiveness model and stable in-wind hovering positions shown in Figure 3, when the wind speed is increasing, the stable positions of a stable in-wind hovering will become further and higher from the setpoints and the control effectiveness reduction will become severe, which indicates that during in-wind flights, the pitch angle should be increased to compensate control effectiveness reduction and to minimize the position error.

3.3 Drag Model

Based on the composition of forces, the wind drag forces of different continuous wind speeds have been estimated with the average values of pitch angles and the weight of the Flapper Drone, as shown by the blue line in Figure 5. Refer to the linear wind drag models proposed in (Kajak et al., 2019) and (Chirarattananon et al., 2017), the drag model for Flapper Drone's in-wind flights is structured as Equation 4, taking the effects of pitch angle θ on the projection area of the FWMAV along the wind speed direction into account.

$$F_{D_{wind}} = C_{D_{wind}} V_{wind}\, cos(\theta) \tag{4}$$

where C_{Drag} is the drag coefficient and V_{wind} is the continuous wind speed. With polyfit() function from MATLAB, the drag model has been identified as $C_{Drag} = 0.3444$. As shown by the red line in Figure 5(A), the identified model indicates that the wind drag is increasing approximately linearly when the wind speed is increasing, which indicates the pitch angle should increase correspondingly to compensate for the effects of increased drag and to minimize the position errors.

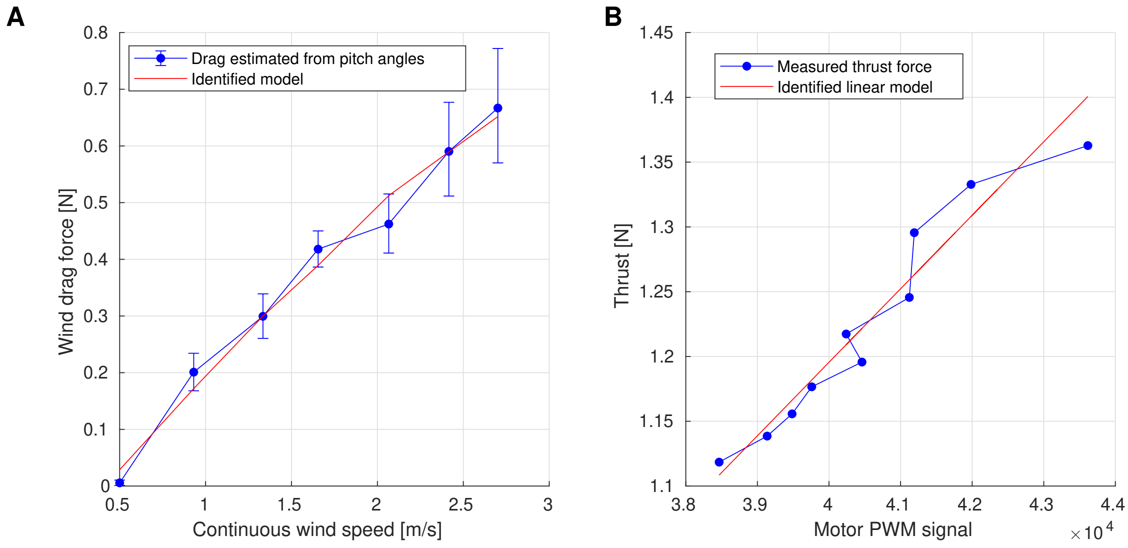


Figure 5. Estimation of wind drag forces and the identified linear drag model.

3.4 Thrust Model

Instead of measuring flapping frequencies directly, the thrust model has been modeled as a relationship between the PWM signal sent to the motors and the thrust generated by the flapping wing. It is important to notice that the thrust used here is measured by the sum of the Flapper Drone's weight and the payloads during stable hovering without any wind disturbance. The flapping frequency of this FWMAV could be correlated with the input PWM signals of the motors as Equation 5.

$$f_{flap} = K_{flap}\, PWM \tag{5}$$

where f_{flap} is the flapping frequency, K_{flap} the conversion coefficient, PWM the magnitude of the input PWM signals. Referring to the linear thrust assumptions in (Kajak et al., 2019), the thrust model has been proposed as a linear model in Equation 6, with both c_1 and c_2 are the fitted coefficients.

$$T = 2(c_1 f_{flap} + c_2) = 2c_1 K_{flap}\, PWM + 2c_2 \quad (6)$$

As shown in Figure 5(B), the thrust model has been identified with OLS estimator, in which the thrust force is approximately linear with the magnitudes of the motor's PWM signal.

4 ADAPTIVE FLIGHT CONTROLLER DESIGN

4.1 Airflow-sensing based adaptive PID controller

As shown in Section 3.2 and Section 3.3, the wind is effecting FWMAV's in-wind flights greatly with the wind drag and the dihedral servo control effectiveness reduction, which renders the FWMAV unable to reach the pre-defined setpoints when attempting to hover stably under the influence of wind disturbance. Based on this situation, an adaptive PID controller has been proposed with an adaptive feedforward (FF) gain in the velocity controller and an adaptive proportional gain in the position controller to compensate for the dihedral servo control effectiveness reduction, minimize the oscillation along X_G axis and improve the pitch stability when hovering. As shown in Figure 6, the adaptive position controller is implemented

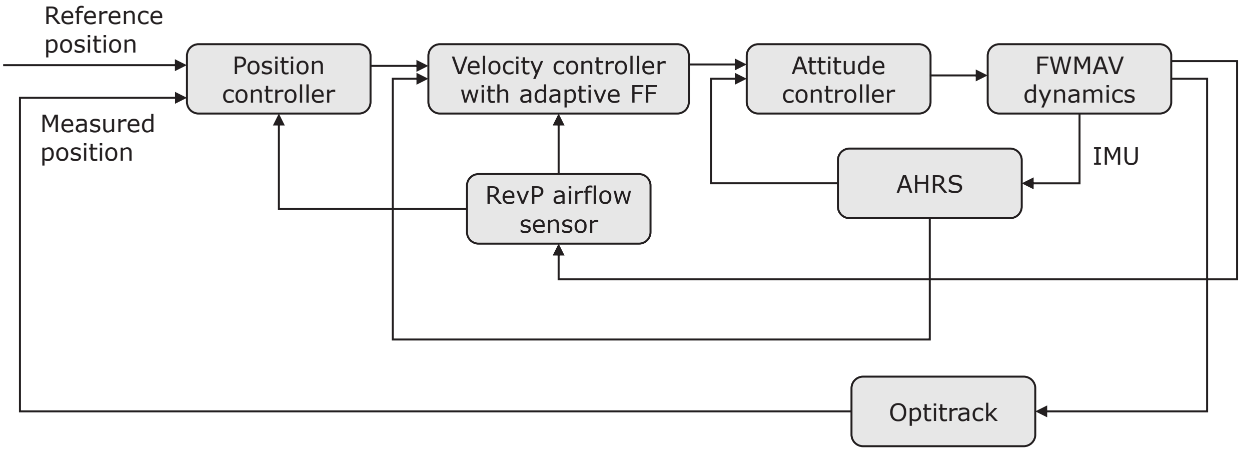


Figure 6. The block diagram showing the implementation of the adaptive-FF controller in FWMAV's control system.

with the measured position from Optitrack, the reference postion from the pre-defined setpoints and the airflow reading from the RevP airlfow sensor [3] as its inputs and will calculate the reference velocity for the adaptive velocity controller as its output.

Similarly, the adaptive velocity controller is implemented with the measured velocity from AHRS, the reference velocity from the position controller and the airflow sensor reading as its inputs and will calculate the reference attitude for the attitude controller as its output.

[3] https://moderndevice.com/products/wind-sensor-rev-p

4.1.1 Airflow-sensing based adaptive position controller

With the structure presented in Figure 7, in the adaptive position controller, with the position error in x_{body} axis e_x calculated in Equation 7, the reference velocity in x_{body} axis $V_{x_{ref}}$ is calculated as in Equation 8 at time t.

$$e_x = x_{ref} - x_{measured} \tag{7}$$

$$V_{x_{ref}}(t) = K_{P_x}\, e_x(t) \;+\; K_{I_x} \int_0^t e_x(\tau)\, d\tau \;+\; K_{D_x} \frac{de_x(t)}{dt} \tag{8}$$

where K_{P_x} is the proportional gain (1.5 in the original position controller), K_{I_x} the integral gain (0.0 in the original position controller) and K_{D_x} is the derivative gain (0.0 in the original position controller). Rather than the constant K_{P_x} value in the original position controller, the value of K_{P_x} is changed adaptively by the K_{P_x} selector during flights inreal time based on the filtered airflow sensor reading. The values of K_{P_x} corresponding to different intervals of wind speeds and filtered airflow sensor reading have been shown in Table 2. However, in real flights, the airflow sensor will output unreliable outlier readings occasionally, which results in rapid changes in K_{P_x} though the wind speed has not reached certain levels. Therefore, an extended Kalman Filter (EKF) and a K_{P_x} selector is employed to filter out the noise and select out an appropriate value for K_{P_x}. At the time $t = t_i$, the K_{P_x} selector works as Algorithm 1.

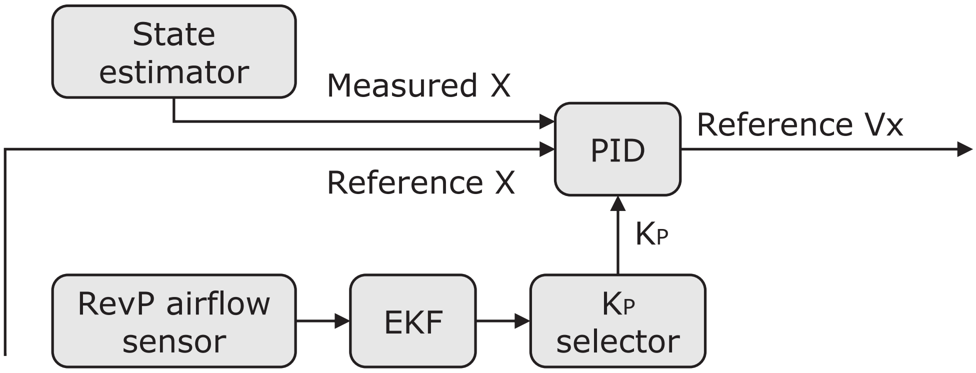


Figure 7. The block diagram of the adaptive position controller.

Wind speed [m/s]	Filtered airflow sensor reading [V]	K_{P_x} [-]	K_{FF_x} [-]	No. of intervals
[0, 0.780)	[0, 1.740)	1.5 (default)	10.0 (default)	1
[0.780, 1.087)	[1.740, 1.830)	1.65	18.5	2
[1.087, ∞)	[1.830, ∞)	1.65	21.5	3

Table 3. The values of K_{P_x} and K_{FF_x} with the corresponding ranges of wind speeds and filtered airflow sensor readings.

Algorithm 1 K_{P_x} and K_{FF_x} selector

```
while $t = t_i$ do
    initialize $s_i \leftarrow 0$, $S_i \leftarrow 0$ and $len \leftarrow 5$
    gain $s_{i-1}$ and $S_{i-1}$ from the previous time step
    read the filtered airflow sensor reading from EKF and assign it to $V_{air}$
    check which filtered airflow sensor reading interval the value of $V_{air}$ is in
    assign the corresponding number of intervals to $s_i$, as $s_i = 1, 2, 3$
end while
if $s_i = s_{i-1}$ then
    $S_i = S_{i-1} + 1$
else if $s_i = 1$ then
    $S = 0, K_{FF_x} = 10.0, K_{P_x} = 1.5$
else
    $S_i = 0$
end if
while $S_i > 0$ do
    $S_i = min(S_i, len)$
end while
if $S_i = len$ then
    update $K_P$ and $K_{FF_x}$ with the $K_{P_x}$ and $K_{FF_x}$ value corresponding to $s_i$ in Table 2
else
    keep the value of $K_{P_x}$ and $K_{FF_x}$ as the previous time step
end if
```

4.1.2 Airflow-sensing based adaptive feedforward velocity controller

As shown in Figure 8, in this adaptive velocity controller, a feedforward term is calculated based on the reference X velocity $V_{x_{ref}}$ as Equation 8 and summed with the output from the PID block. Instead of using a constant K_{FF_x} as the traditional feedforward controller, the value of K_{FF_x} is adjusted actively by the K_{FF_x} selector during the flight based on the filtered airflow sensor reading in this adaptive-ff controller.

$$\theta_{FF} = K_{FF_x} \, V_{x_{ref}} \tag{9}$$

As presented in and Section 3.3, the dihedral servo control effectiveness reduction is becoming more severe with greater wind drag forces when the wind speed is increasing. Therefore, in this controller, the K_{FF_x} has been designed as increasing step to step adaptively to the filtered airflow sensor readings. The values of K_{FF_x} corresponding to different intervals of wind speed and filtered airflow sensor reading has been shown in Table 2. Similar with the K_P case in real flights, K_{FF_x} will also change rapidly due to the unreliable outlier readings from the airflow sensor though the wind speed has not reached certain levels. Therefore, same as the adaptive position controller, an extended Kalman Filter (EKF) and a K_{FF_x} selector is also employed to filter out the noise and select out an appropriate value for K_{FF_x}. At the time $t = t_i$, the K_{FF_x} seletcor works as Algorithm 1.

Based on the airflow sensor, the adaptive position controller and the adaptive velocity controller, the FWMAV could estimate the wind speeds it has encountered with EKF-filtered airflow sensor readings then update K_P and K_{FF_x} in the pitch loop with proper values actively to compensate the effects brought by winds and gusts.

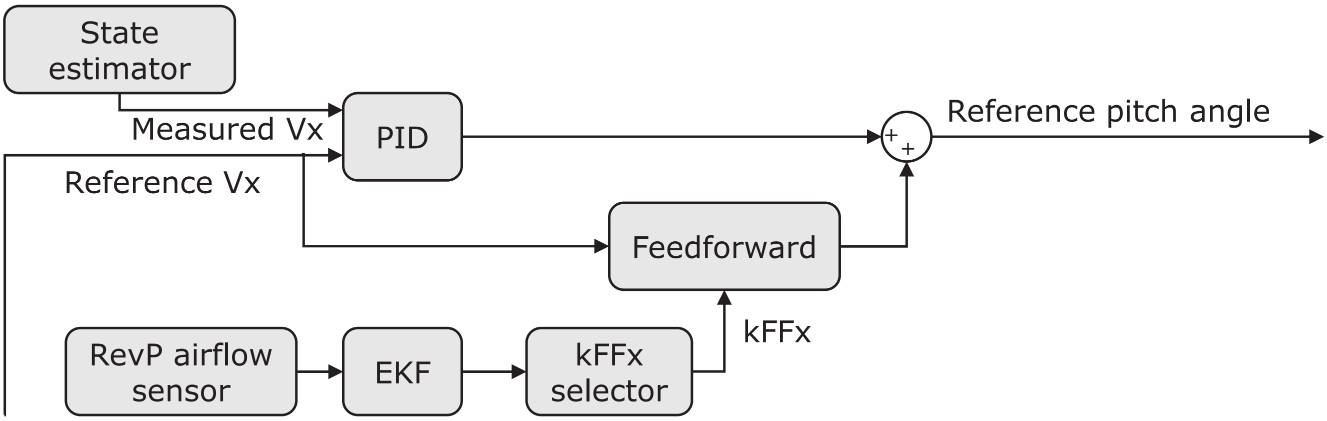


Figure 8. The block diagram of the adaptive velocity controller.

5 EXPERIMENTS

5.1 In-gust hover flights with original PID controller

Several in-gust hovering flights have been conducted with the original PID controller under the gusts alternating the wind speed between $0.5m/s$ and $2.4m/s$, and between $0.5m/s$ and $2.1m/s$ at the frequency of $0.25Hz$, $0.33Hz$, $0.50Hz$ and $0.75Hz$. The time histories of position errors in X_G and Z_G axis from the flights under the gusts alternating the wind speed between $0.5m/s$ and $2.4m/s$ have been shown in Figure 9.

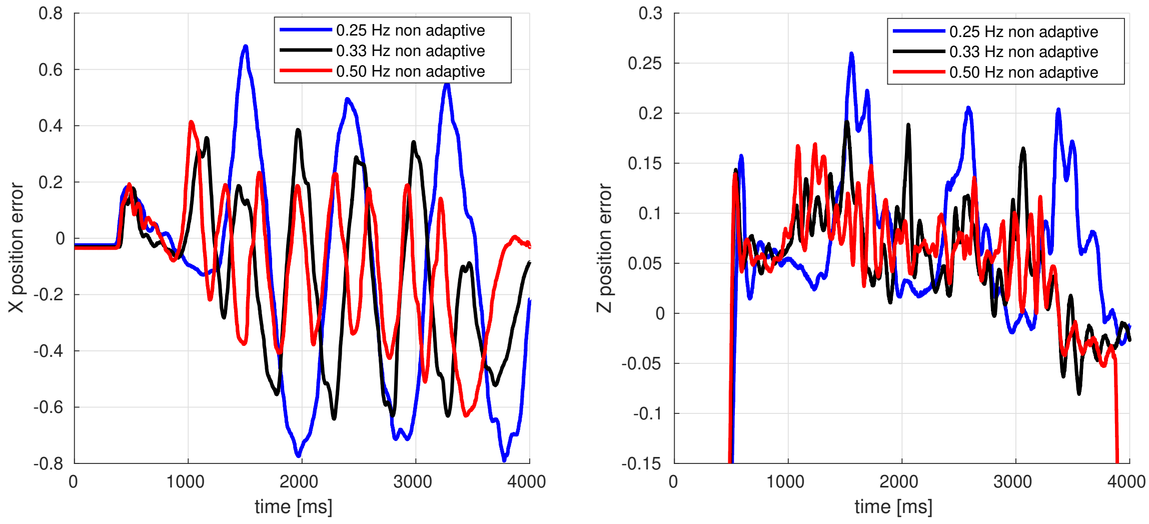


Figure 9. Time histories of X and Z axis position errors during the in-gust hovering flights with the original PID controller.

As shown in Figure 9, the FWMAV oscillated greatly under these dynamic gusts. From Table 4, the RMS errors of both X_{error} and Z_{error} are increasing, which indicates that the position control is degrading and the oscillation becomes more and more severe when the changing frequency of the gust f_{gust} is decreasing,

Wind speed	0.5 m/s and 2.4 m/s			0.5 m/s and 2.1 m/s		
	RMS errors					
f_{gust} [Hz]	X_G [m]	Z_G [m]	θ [deg]	X_G [m]	Z_G [m]	θ [deg]
0.25	0.4565	0.0798	9.5244	0.4044	0.0678	9.9005
0.33	0.3499	0.0516	7.3712	0.3445	0.0607	7.9331
0.50	0.2659	0.0475	5.6884	0.2091	0.0329	4.9303
0.75	0.2451	0.0447	5.1634	0.1697	0.0368	5.0958

Table 4. The RMS errors of the position in X_G and Z_G axis, and pitch angle θ from the in-gust hovering flights with the original PID controller under the gust alternating between $0.5m/s$ and $2.4m/s$, and between $0.5m/s$ and $2.1m/s$.

since within one period the gust of high wind speed always lasts longer (from 0.75 s to 2.00 s) before decreasing to 0.5 m/s.

Furthermore, when the maximum wind speed increases from 2.1 m/s to 2.4 m/s, the oscillation also becomes more severe because the gust intensity is increasing.

5.2 In-gust hover flights with adaptive PID controller

To validate the adaptive position and velocity controller, several in-gust hovering flights have been conducted under the gusts alternating the wind speed between $0.5m/s$ and $2.4m/s$, and between $0.5m/s$ and $2.1m/s$ at the frequency of $0.25Hz$, $0.33Hz$, $0.50Hz$ and $0.75Hz$. The time histories of position errors in X_G and Z_G axis from the flights under the gusts changing the wind speed between $0.5m/s$ and $2.4m/s$ have been shown in Figure 10.

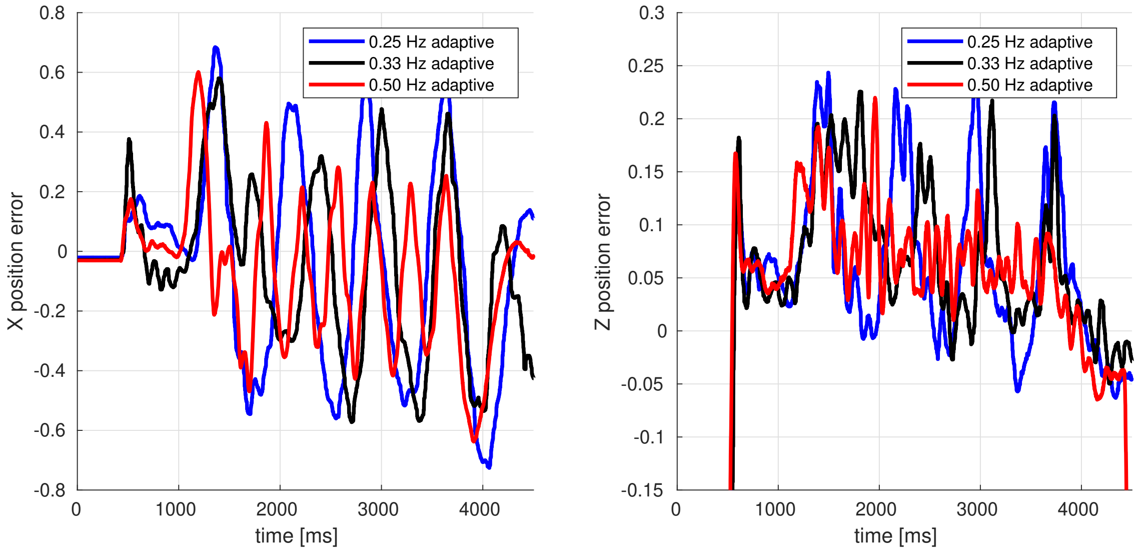


Figure 10. Time histories of X and Z axis position errors during the in-gust hovering flights with adaptive position and velocity controller.

Wind speed	$0.5\ m/s$ and $2.4\ m/s$			$0.5\ m/s$ and $2.1\ m/s$		
	RMS errors					
f_{gust} [Hz]	X_G [m]	Z_G [m]	θ [deg]	X_G [m]	Z_G [m]	θ [deg]
0.25	0.3435	0.0833	8.8880	0.3560	0.0671	9.2566
0.33	0.3184	0.0620	5.5373	0.2761	0.0483	7.1083
0.50	0.1947	0.0313	1.6880	0.1705	0.0332	5.9009
0.75	0.1235	0.0416	3.1116	0.1008	0.0321	5.0009

Table 5. The RMS of position errors in X_G and Z_G axis, and pitch angle θ from the in-gust hovering flights with the adaptive PID controller under the gust alternating between $0.5m/s$ and $2.4m/s$, and between $0.5m/s$ and $2.1m/s$..

Similar with the in-gust flight experiments in Section 4.3, the FWMAV is oscillating during these in-gust hover flights. As presented in Table 5 and Table 6, when the gust changing frequency f_{gust} is decreasing and the maximum wind speed increases from $2.1\ m/s$ to $2.4\ m/s$, the oscillation is becoming more and more severe.

6 PERFORMANCE ANALYSIS AND COMPARISON

The root mean square error (RMSE) of the X_G position, the Z_G position and pitch attitude angle θ from in-gust flights with both the original PID controller and the adaptive PID controller have been shown in Figure 11 and Figure 12, together with the average in-flight current intensities.

As shown, the RMSE value of the X_G position $RMSE_X$ is decreasing when the gust changing frequency is decreasing. Comparing with the original PID controller, $RMSE_X$ values of the cases with the adaptive PID controller are always lower which indicates a better performance in disturbance rejection and X_G position control. Furthermore, the RMSE values of the pitch attitude angle $RMSE_\theta$ of the adaptive PID cases are also always lower than the cases with the original PID controller, except the case of gusts alternating between $0.5\ m/s$ and $2.1\ m/s$ at $0.33Hz$, which presents that the pitch stability of FWMAV's in-gust flights has been improved.

The RMSE values of Z_G position $RMSE_Z$ are lower than the cases of original PID controller in the high gust changing frequency cases ($0.75Hz$) and remain the similar magnitudes in the lower cases ($0.25Hz$ and $0.33Hz$). Furthermore, for the flights with adaptive PID controller, the average in-flight current intensity values are slightly higher than the cases with the original PID controller when the gust changing frequencies are higher ($0.50Hz$ and $0.75Hz$), and are slightly lower than the cases with the original PID controller when the gust changing frequencies are lower ($0.33Hz$), which indicates the energy consumption levels remain similar in these in-gust flights.

Furthermore, as shown in Figure 13, the dihedral outputs of the in-gust flights with the adaptive PID controller is much closer to the level of the dihedral commands of the in-gust flights with the original PID controller, comparing with the dihedral outputs of the in-gust flights with the original PID controller. Therefore, the dihedral servo control effectiveness reduction has been alleviated by this adaptive PID controller efficiently.

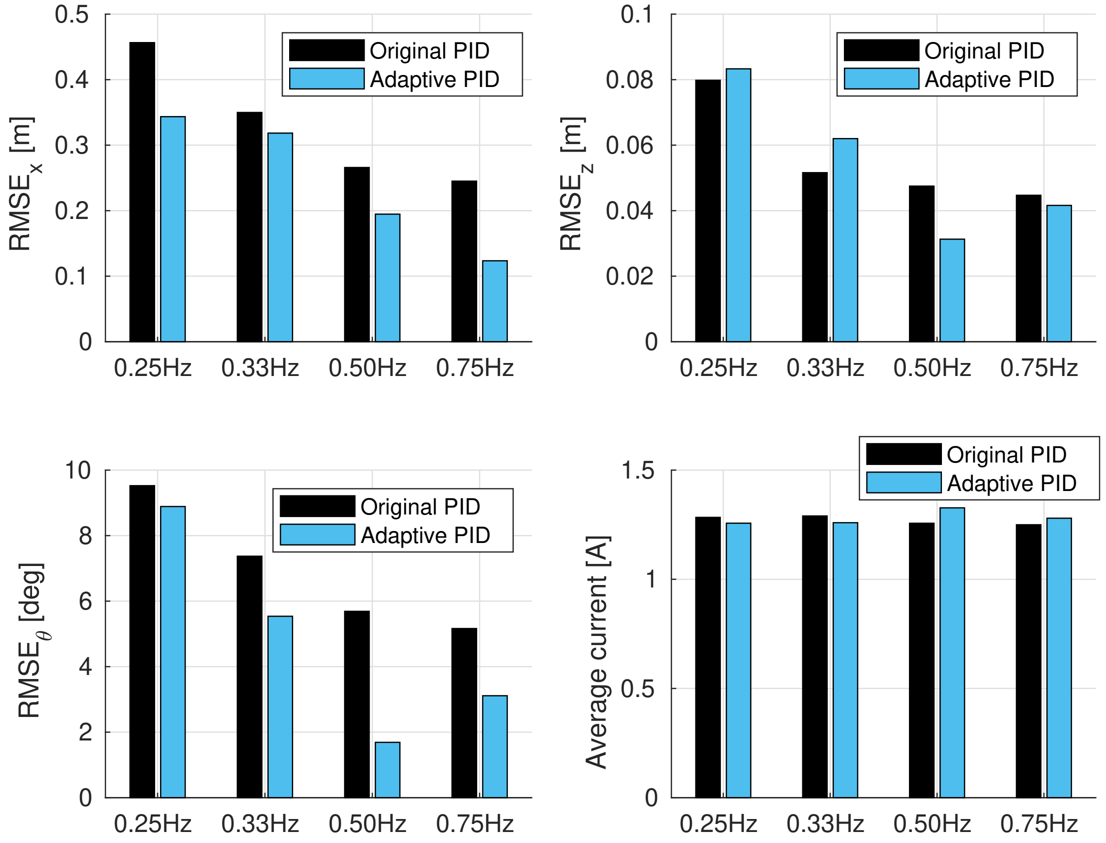


Figure 11. Average values of position errors in X_G axis and current intensities during the in-gust hovering flights under the gust changing between $0.5\ m/s$ and $2.4\ m/s$.

7 CONCLUSION

In this paper, two main aspects of wind effects on FWMAVs, which are wind drag force and servo control effectiveness reduction, have been identified in Chapter 3. An adaptive PID controller has been proposed in Chapter 4 and implemented with a RevP airflow sensor in the FWMAV's control system. This controller was validated with several hovering flights under the gusts changing the wind speed at different frequencies in Chapter 5.

Comparing with the original PID controller in which K_P and K_{FF_x} values are constant, this adaptive PID controller enables the FWAMV to damp the oscillation in X_G direction and improve the pitch stability efficiently by sensing the airflow speed actively and changing K_P and K_{FF_x} value, when encountering dynamic changing gusts up to $2.4\ m/s$ in X_G direction.

Future work will investigate the possibility to design a similar adaptive controller for the thrust loop of the FMWAV, which could improve the position control along Z_G axis and improve the energy efficiency. A better estimator for current wind speed could also be designed and implemented with the airflow sensor to replace the EKF in adaptive PID controller. Regarding the servo control effectiveness reduction, the possibility of implementing a low-level feedback controller for the dihedral servo to solve the control

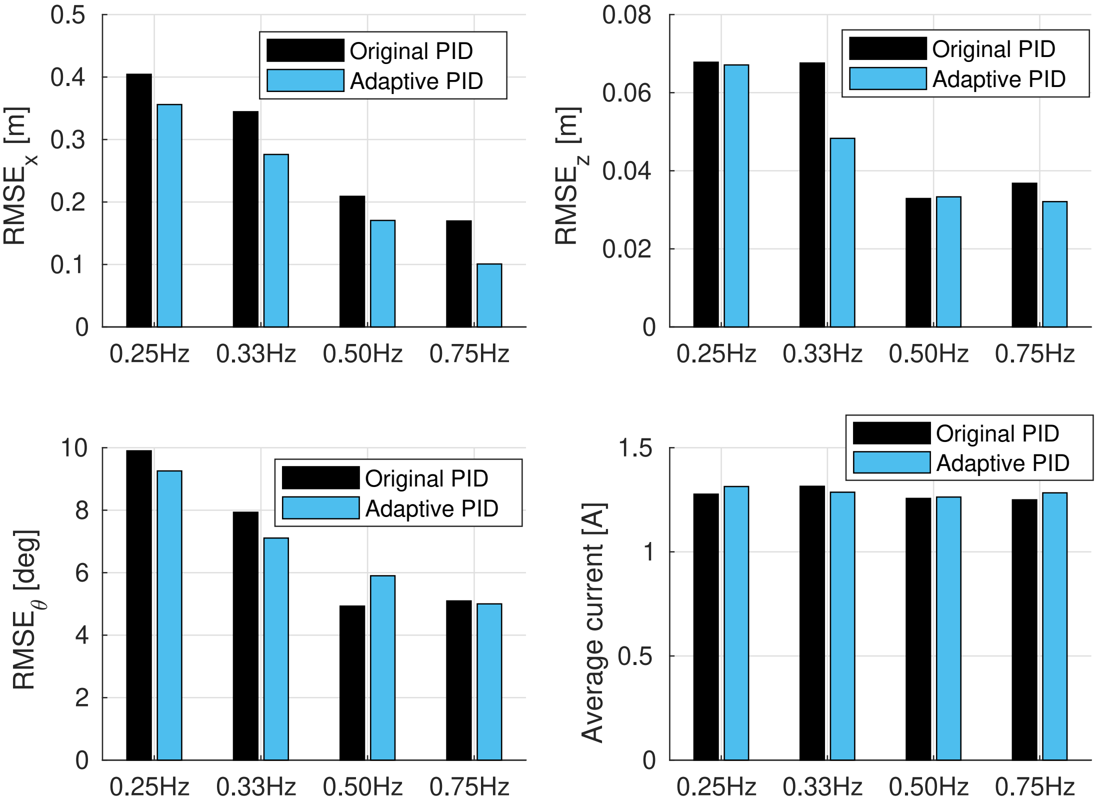


Figure 12. Average values of position errors in X_G axis and current intensities during the in-gust hovering flights under the gust alternating between $0.5m/s$ and $2.1m/s$.

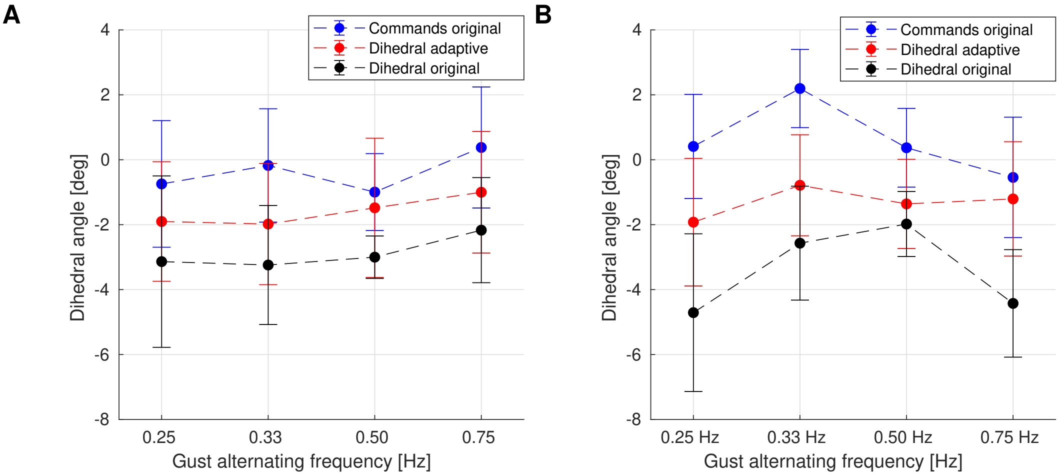


Figure 13. Comparison between dihedral commands from the flights with the original PID controller (blue), dihedral outputs from the flights with the adaptive PID controller (red) and dihedral outputs from the flights with the original PID controller (black) during the in-gust hovering flights under the gust alternating between $0.5m/s$ and $2.4m/s$ (A), and between $0.5m/s$ and $2.1m/s$ (B) at different frequencies.

effectiveness reduction thoroughly could be studied. Furthermore, to reduce the average value of position error, an integral gain could be introduced in the position controller in Figure 6.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

CW and SW conceptualized and designed the study under the supervision of GdC and SH. CW wrote the first draft of the manuscript. CW performed the robot experiments and analyzed the resulting data with the support of SW. CW designed the adaptive position and velocity controller, with the support of SW. All authors contributed to editing the manuscript, and approved the submitted version.

FUNDING

Details of all funding sources should be provided, including grant numbers if applicable. Please ensure to add all necessary funding information, as after publication this is no longer possible.

ACKNOWLEDGMENTS

We thank Martijn den Hoed and Hang Yu for helping in hardware maintenance, and Sven Pfeiffer for assisting the flight experiments.

SUPPLEMENTAL DATA

Supplementary Material should be uploaded separately on submission, if there are Supplementary Figures, please include the caption in the same file as the figure. LaTeX Supplementary Material templates can be found in the Frontiers LaTeX folder.

DATA AVAILABILITY STATEMENT

The datasets collected from experiment flights for this study can be found in the Adaptive-Flapper-Datasets repository .

REFERENCES

Bains, K. (2020). System identification of the delfly nimble: Modeling of the lateral body dynamics

Castano, L., Airoldi, S., McKenna, T., and Humbert, J. (2014). Gust rejection using force adaptive feedback for roll. In *14th AIAA Aviation Technology, Integration, and Operations Conference*. 2588

Chirarattananon, P., Chen, Y., Helbling, E. F., Ma, K. Y., Cheng, R., and Wood, R. J. (2017). Dynamics and flight control of a flapping-wing robotic insect in the presence of wind gusts. *Interface focus* 7, 20160080

Chirarattananon, P., Ma, K. Y., Cheng, R., and Wood, R. J. (2015). Wind disturbance rejection for an insect-scale flapping-wing robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE), 60–67

Fuller, S. B., Straw, A. D., Peek, M. Y., Murray, R. M., and Dickinson, M. H. (2014). Flying drosophila stabilize their vision-based velocity controller by sensing wind with their antennae. *Proceedings of the National Academy of Sciences* 111, E1182–E1191

Ho, S., Nassef, H., Pornsinsirirak, N., Tai, Y.-C., and Ho, C.-M. (2003). Unsteady aerodynamics and flow control for flapping wing flyers. *Progress in aerospace sciences* 39, 635–681

Kajak, K. M., Karásek, M., Chu, Q. P., and De Croon, G. (2019). A minimal longitudinal dynamic model of a tailless flapping wing robot for control design. *Bioinspiration & biomimetics* 14, 046008

Karásek, M., Muijres, F. T., De Wagter, C., Remes, B. D., and De Croon, G. C. (2018). A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns. *Science* 361, 1089–1094

Lee, J., Ryu, S., and Kim, H. J. (2020). Stable flight of a flapping-wing micro air vehicle under wind disturbance. *IEEE Robotics and Automation Letters* 5, 5685–5692

Nijboer, J., Armanini, S. F., Karasek, M., and de Visser, C. C. (2020). Longitudinal grey-box model identification of a tailless flapping-wing mav based on free-flight data. In *AIAA Scitech 2020 Forum*. 1964

Olejnik, D. A., Wang, S., Dupeyroux, J., Stroobants, S., Karasek, M., De Wagter, C., et al. (2022). An experimental study of wind resistance and power consumption in mavs with a low-speed multi-fan wind system. In *2022 International Conference on Robotics and Automation (ICRA)*. 2989–2994. doi:10.1109/ICRA46639.2022.9811834

O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., et al. (2022). Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics* 7, eabm6597

Skordos, A., Chan, P., Vincent, J., and Jeronimidis, G. (2002). A novel strain sensor based on the campaniform sensillum of insects. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 360, 239–253

Wang, S., Olejnik, D., de Wagter, C., van Oudheusden, B., de Croon, G., and Hamaza, S. (2022). Battle the wind: Improving flight stability of a flapping wing micro air vehicle under wind disturbance with onboard thermistor-based airflow sensing. *IEEE Robotics and Automation Letters* 7, 9605–9612

Yang, H., Cheng, L., Xia, Y., and Yuan, Y. (2017). Active disturbance rejection attitude control for a dual closed-loop quadrotor under gust wind. *IEEE Transactions on control systems technology* 26, 1400–1405

3 Literature Study

3.1. FWMAV Platforms

In nature, both birds and insects fly with their flapping wings. However, their underlying flight principles are rather different [2]. Except hummingbirds, most large/medium-scale birds gain aerodynamic forces by relatively low frequency and small downstroke motions, during which the flapping stroke plane is almost vertical to flight path [38]. In contrast, during flights of most insects, the amplitude of flapping is larger and the frequency is also much higher with a parallel stroke plane to flight path [12]. Therefore, insects' capability of precise hovering is not possessed by most birds. Furthermore, comparing with birds which could use their tails during flight, most insects produce control forces with only their wings by adjusting wing kinematics actively [41], which results in a huge difference in flight attitude control of birds and insects.

From the differences in aerodynamic mechanism, size and attitude control, FWMAV platforms could be separated into two categories: bird-inspired FWMAVs and insect/hummingbird-inspired tailless FWMAVs. This chapter introduces four FWMAV platforms which have active research progress in recent years: insect-inspired Robobee by Harvard University, hummingbird-inspired Flappy Hummingbird by Purdue University, hummingbird/insect-inspired DelFly Nimble by TU Delft and bird-inspired GRIFFIN ornithopter by University of Seville.

3.1.1. Robobee

As shown in Figure 3.1, Robobee, which was first introduced by Harvard University in 2013, is an 80-milligram, insect-scale fly-inspired FWMAV[26]. As the source of form and function inspiration for Robobee, *Diptera* was used as a model system for its simple flight apparatus (two wings) and exemplary aerial agility[11]. The flight of *Diptera*, in which wings undergo a complex trajectory with three rotational degrees of freedom[15], is simplified in Robobee fly as a reciprocating flapping motion with pitch rotation regulated by passive compliant flexures[46], for the sake of practicality of mechanism design and manufacture.

To mimic the aerial prowess of flies in Robobee, tiny but high-efficiency mechanical components which could tackle with miniaturization challenges from force-scaling laws is required. Therefore, to solve this issue, high-power-density flight muscles and unique manufacturing methodology was developed and implemented in Robobee[26]. Firstly, voltage-driven piezoeletric bimorphs which could generate bidirectional forces were used as flight muscles and geometrically optimized for energy density[47]. Secondly, a manufacturing methodology capable of producing articulated and flexure-based sub-millimeter mechanisms efficiently, smart composite microstructures (SCM), was developed and applied for manufacturing all electromechanical elements of Robobee.

The wing-flapping motion of Robobee is driven by a four-bar linkage serving as a lever arm to amplify the small displacement of the piezoelectric flight muscle[26]. Besides, a passive elastic flexure hinge at wing base was employed for wing pitch motion. With the structure shown in Figure 3.2, the wing

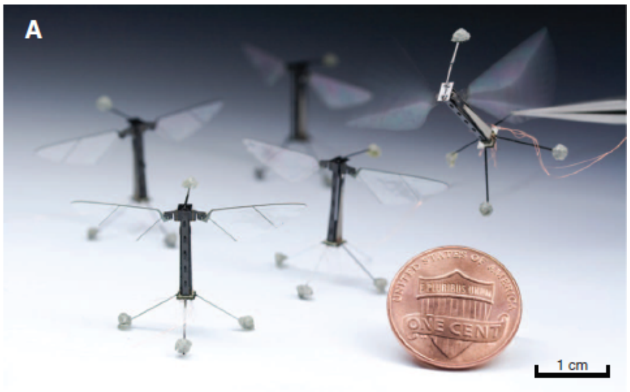


Figure 3.1: Robobee shown alongside with a U.S. penny for scale

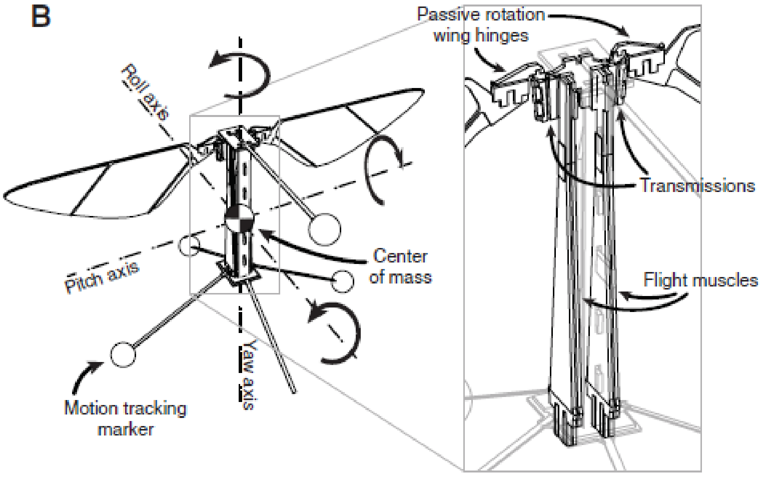


Figure 3.2: Robobee structure

kinematics of Robobee could resemble wing motions of insect flight and generate sufficient lift force for flight[25]. So far, hovering flight[26], basic maneuvers[6], acrobatic flight of perching on a vertical wall[7] and disturbance-rejection control[9, 8] has been demonstrated on Robobee.

3.1.2. Flappy Hummingbird

Inspired by the sustained stable hovering and extremely acrobatic maneuvering of hummingbirds, Flappy Hummingbird was developed and first introduced by researchers from Purdue University in 2017[50], shown in Figure 3.3. Under the conditions of stringent size, weight, and power constraints, it is a challenging task to design a hummingbird-inspired at-scale tailless FWMAV with only two actuators. Therefore, a systematic optimization approach was proposed in [42], and covered the complete system models and analysis of wing-actuation dynamics, control authorities, body dynamics, mechanical limitations and electrical constraints.

During the designing phase, three optimization tasks were conducted[42]:

- lift-to-weight ratio optimization
- control bandwidth optimization
- control authority optimization

From these optimization tasks, three different prototypes of Flappy Hummingbird were derived. Since the prototype from lift-to-weight ratio optimization which demonstrates the general design could generate sufficient lift to fly, flight experiments were carried out on two other prototypes and showed that

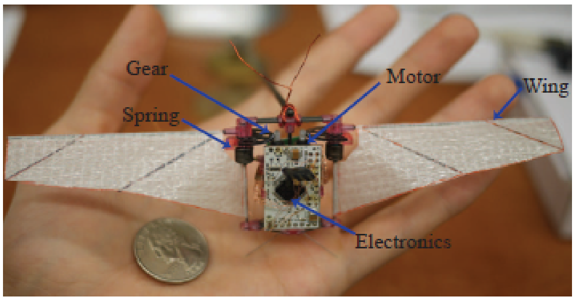


Figure 3.3: Flappy Hummingbird prototype

the prototype from control authority optimization possessed a better flight capability. As shown in Figure 3.4, the main structure of Flappy Hummingbird, the wing-actuation system consists of a brushless motor as actuator, a pair of reduction gears for torque transmission, a energy-restoring spring and a pair of 2-DoF cambered wings[42]. Furthermore, the wings are driven actively only for flapping motion and are able to rotate passively from the coupling effects of aerodynamics and inertial loading.

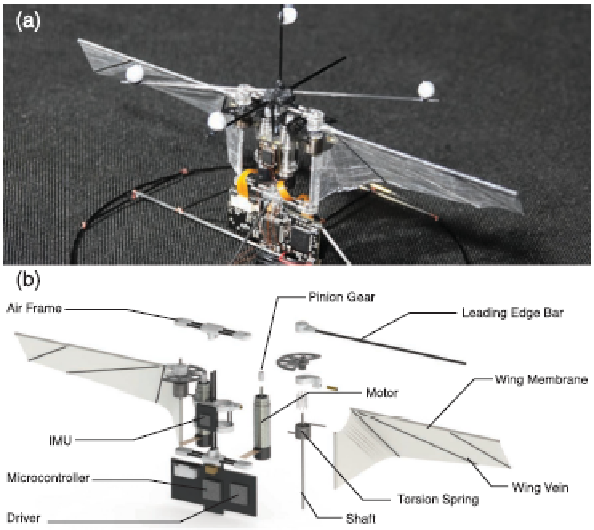


Figure 3.4: Flappy Hummingbird structure

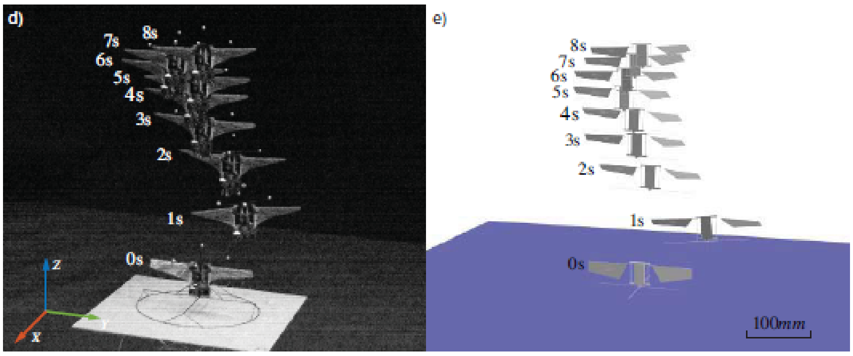


Figure 3.5: Flappy Hummingbird simulator

Based on this structure and prototypes, Flappy Hummingbird was developed successfully and demonstrated its stable hovering, trajectory following, payload carrying[50] and RL-based extreme evasive maneuvers[14]. Furthermore, an open-source dynamic simulator of FWMAV was developed based on identified dynamic model of Flappy Hummingbird[13], as shown in Figure 3.5.

3.1.3. DelFly Nimble

As the successor of DelFly and DelFly II, DelFly Nimble was developed by researchers of MAVLab, TU Delft and Wageningen University and Research in 2018[22]. Being a fly-inspired tailless FWMAV, DelFly Nimble could accurately imitate the rapid escape turn-over maneuvers of flies despite 55 times the size of a fruit fly. However, without mimicking any natural flyers in details, the size, wing morphology and wing kinematics of DelFly Nimble were optimized for maximal power efficiency since it is driven by brushless motors but not powerful animal flight muscles. Consequentially, DelFly Nimble has a weight of 28.2 g and a 33 cm wingspan, and the flapping frequency of its 14-cm-long wings could reach approximately 17 Hz while hovering. Besides, its hovering duration is 5 min and the flight range is more than 1 Km, on a single battery charge, which indicates its power efficiency.

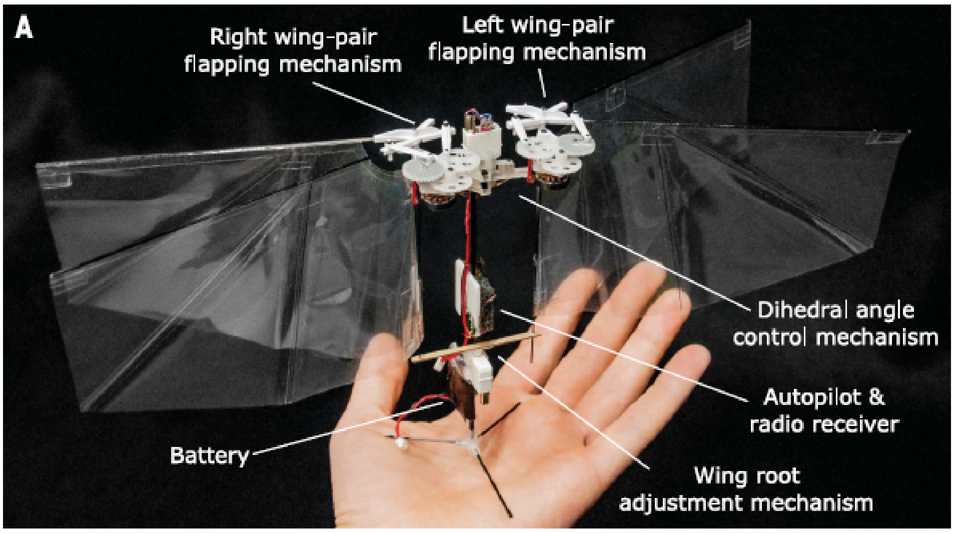

A
Right wing-pair flapping mechanism
Left wing-pair flapping mechanism
Dihedral angle control mechanism
Autopilot & radio receiver
Battery
Wing root adjustment mechanism

Figure 3.6: DelFly Nimble

As shown in Figure 3.6, DelFly Nimble is driven by two motors for left and right wing-pair respectively, and two servos for wing root angle control and dihedral angle control. In the attitude control of DelFly Nimble, yaw torque control is realised by wing root adjustment mechanism in left of Figure 3.7, and pitch torque is controlled by dihdral angle between left and right wing-pairs in right of Figure 3.7. The roll torque is actuated by difference in the flapping frequency between left and right wing-pairs.

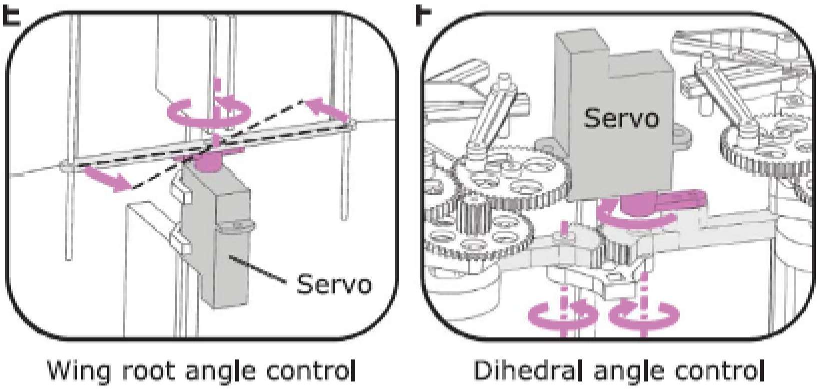


Figure 3.7: Left: wing root adjustment mechanism for yaw torque control, Right: dihedral angle control mechanism for pitch torque control

Based on the simple structure and great flight capability of DelFly Nimble, its commercial version, Flapper Drone was introduced to market, as shown in Figure 3.8. In this project, Flapper Drone is also selected as the platform for FWMAV in-gust dynamic modelling, flight control strategy design and flight experiments.

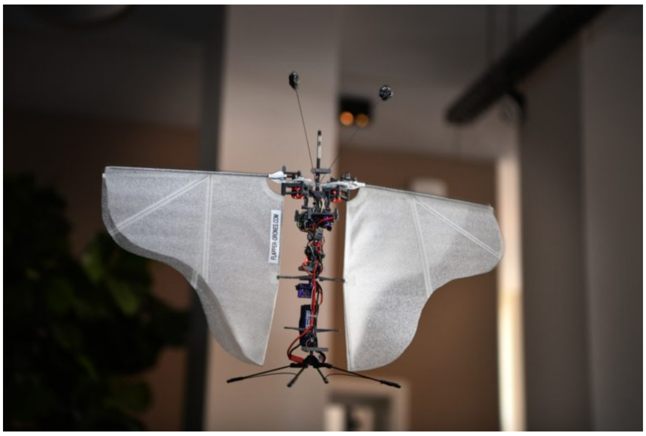


Figure 3.8: Flapper Drone

3.1.4. GRIFFIN Ornithopter

GRIFFIN Ornithopter in Figure 3.9 is the bird-inspired FWMAV platform developed for GRIFFIN-ERC project whose aim is to enable bird-inspired FWMAV to fly minimizing energy consumption, to perch on curved surfaces and to perform dexterous manipulation[1]. It was first introduced in [28] by researchers from University of Seville. Inspired by birds, GRIFFIN Ornithopter has one pair of wings with a 50-cm-long wingspan and a tail whose angle is controlled by a corresponding servo.

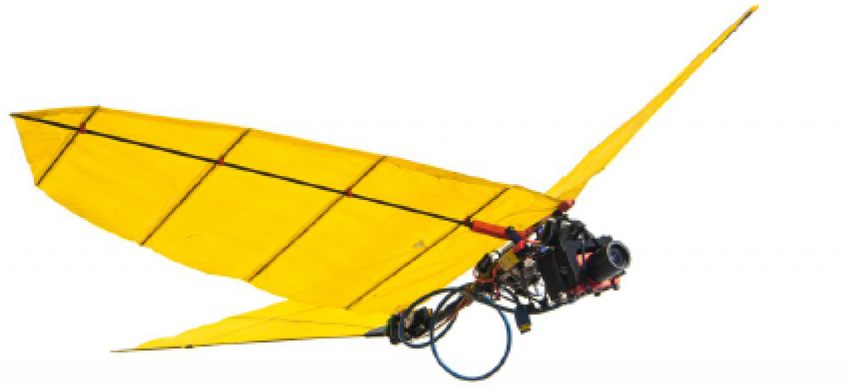


Figure 3.9: GRIFFIN Ornithopter

Due to its structure, recent research on this platform was carried out in its longitudinal control. A longitudinal dynamic model for GRIFFIN Ornithopter was proposed in [28]. Research in energy-efficient kino-dynamics planning and NN-based trajectory optimization was conducted based on this model in [34] and [31]. Different from the objective of agility, robustness and maneuverability in those research

[1] https://griffin-erc-advanced-grant.eu/project-overview/

mentioned above, the research on perching/landing motion on GRIFFIN Ornithopter was basically towards the objective of in-flight manipulation, including cooperative manipulation, with small-scale compliant dual-arm mechanism in [40].

3.2. FWMAV Aerodynamics and In-Gust Dynamics

The energy-efficient flight, agility and maneuverability of birds and insects has always been an obsession for biologists and engineers. Starting from Leonardo da Vinci in 15th century, there have many attempts for understanding the underlying principles of flapping wings. In 20th century, as theoretical aerodynamics became rather mature, the lift-generation mechanism of flapping wings was discovered as Leading Edge Vortex (LEV). In past decades, research on generation and characteristics of LEV was mainly conducted by both CFD simulation[37, 19, 44, 35] and experiments observation[44, 11]. In this chapter, the formation mechanism and lift-generation of LEV is introduced in subsection 3.2.1, and in-gust dynamics of flapping wings is stated in subsection 3.2.2.

3.2.1. Leading edge vortex

Formation mechanism

Similar to conventional airfoils, the lift generated due to LEV is from the pressure difference between upper and lower surface of airfoils. However, the formation of low-pressure area in LEV cases is more complex. As the airfoil increases its angle of attack (AoA) after certain critical angle, the flow stream going over the airfoil separates near leading edge and reattaches before trailing edge[37], which is the phenomenon of stall. In the insect/FWMAV flight regime, Reynolds number Re is always of O(10_4) or lower, which indicates fluid inertial forces dominate flow field of these flights. In such cases of low Reynolds number, a leading edge vortex forms and could keep stable in the separation zone as shown in Figure 3.10. For the reason that the airfoil translates at a rather high AoA, a greater downward momentum imparted to flow enhances the vortex substantially, which means the vorticity is being removed by viscous diffusion within LEV at same rate as its generating rate at leading edge[44] and results in stable LEV structure shown in Figure 3.11.

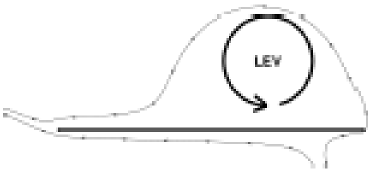


Figure 3.10: LEV formation.

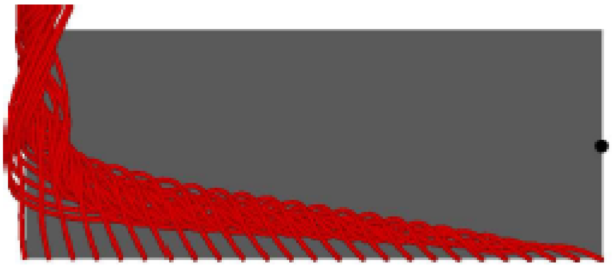


Figure 3.11: A stable LEV attached on upper surface of a thin airfoil.

In theory, the formation of LEV could be interpreted as the balance between the pressure gradient, the centrifugal force and the Coriolis force in the momentum equation of aerodynamics[37], which also influences the stability of LEV.

Lift generation of LEV

In general, LEV attached to upper surface of airfoil generates a lower pressure area as shown in Figure 3.12 and lift is mostly generated from a large suction on the upper surface brought by great pressure difference[37]. Furthermore, in the cases of low AoA in which no LEV is formed, flow moving exclusively towards trailing edge on the upper surface results in a large skin friction. In contrast, in the cases with LEV, flow moves over only a part of upper surface, which means the magnitude of skin friction force is reduced[44]. The lift and drag force brought by LEV is illustrated in Figure 3.13.

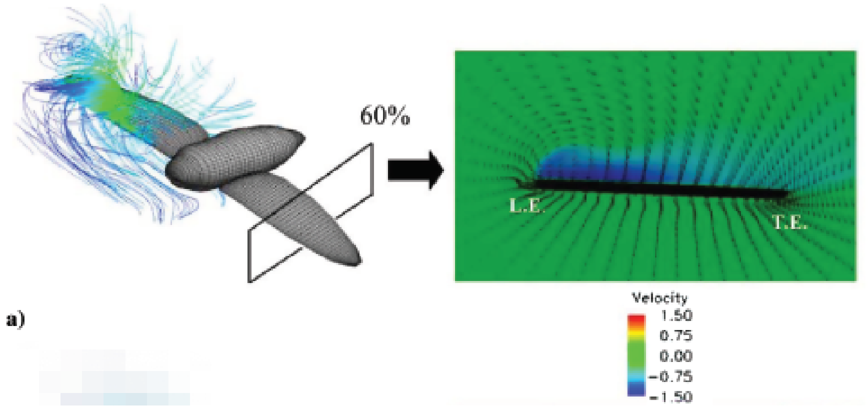


Figure 3.12: Low pressure area brought by LEV.

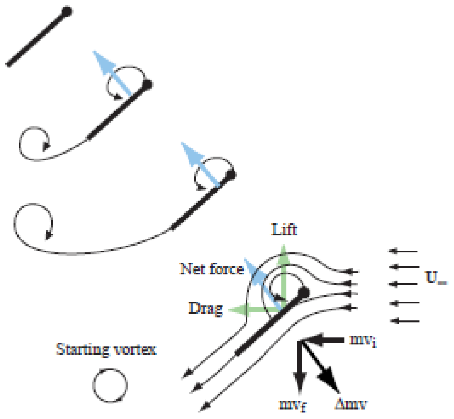


Figure 3.13: Lift and drag force brought by LEV.

The lift-generation mechanism of Flapper Drone's wing-pairs is shown in Figure 3.14. In Figure 3.14, flow lines are indicated by black lines, and induced velocity is shown by deep blue arrows while light blue arrows show the resultant force acting on the wing.

The clapping motion is shown in A-C part of Figure 3.14. In A, the wings come near to each other dorsally, and their leading edges touch or reach closest positions initially along with the rotation of wing around leading edge in B. In C, vorticity detached from the trailing edge turns into stopping vortices and dissipates in to the wake while the LEV also loses strength. Moreover, an additional thrust is generated by the closing of gap between the two wings, which presses out air between two wings. The flinging motion is shown in D-F part of Figure 3.14. In D, by means of wing rotation around trailing edge, two wings fling apart. From D to E, as the gap becomes larger when the leading edge translates away from each other, air rushes in and fills the gap again, which results in an initial boost in circulation around two wings. In F, LEV forms again during the translation of two wings with high AoA. The vortices near the trailing edge counteract each other due to opposite circulation[35].

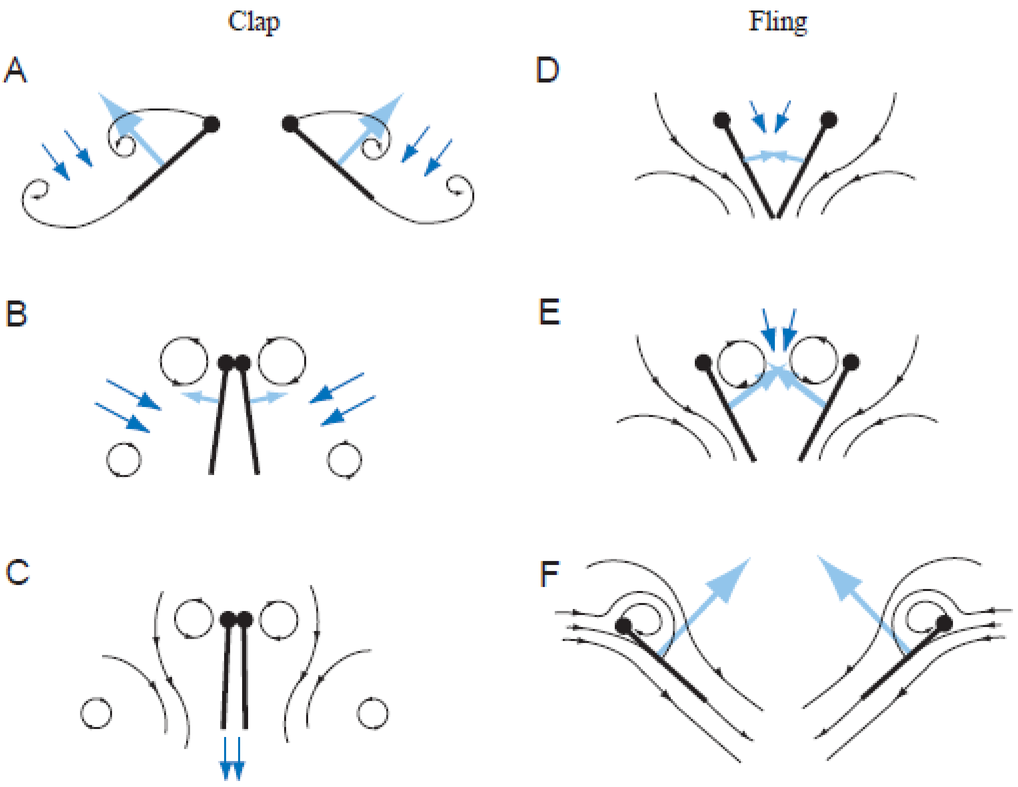


Figure 3.14: LEV and lift generation in clapping and flinging motion of Flapper Drone's wing-pairs.

3.2.2. In-gust dynamics of flapping wings

Generally for the sake of simplicity and practicality, the effects on FWMAVs by external disturbance, i.e. gusts were mostly computed and evaluated numerically in the stroke-averaged form, which is the average force generated per stroke due to relatively high flapping frequency. The stroke-averaged forces contributed by wing-flapping could be described as a function of incoming frontal (along $\hat{y}$ axis in Figure 3.15) and lateral (along $\hat{x}$ axis in Figure 3.15) gust speed[8].

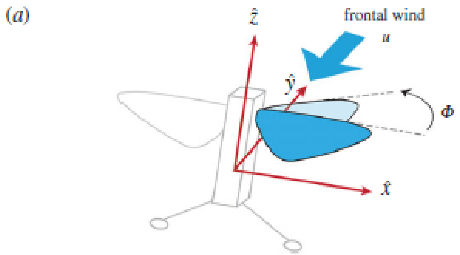


Figure 3.15: Body-fixed coordinates definition of a FWMAV (Robobee).

Based on findings in [1], the drag forces is assumed as linearly proportional to relative gust speed, and this relationship is shown to be approximately valid for a FWMAV at the scale of Robobee encountering both frontal and lateral gusts of low speeds (less than 1.5 m/s). And both frontal and lateral gusts result in additional lift (in $\hat{x}$ axis in Figure 3.15) which is quadratic to relative gust speeds. These relationships are shown in Figure 3.16 and were proven as accurate in further model-based disturbance-rejection control strategy design.

Besides aforementioned research, CFD simulation were carried out to study in-gust dynamics of flapping wings in [20]. As shown in Figure 3.17, both lift and drag forces increase when the speed of frontal

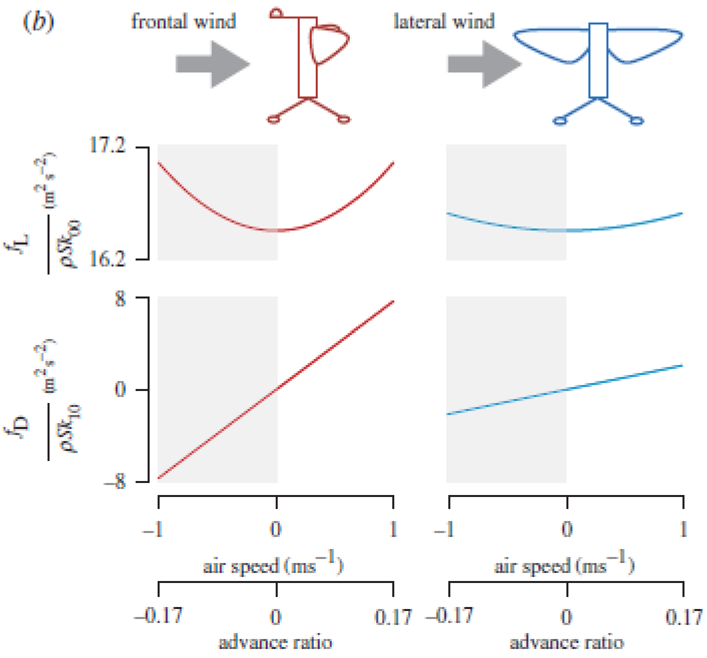


Figure 3.16: Relationships between forces and relative gust speeds.

gusts encountered increases. And the magnitude of effects on lift is slightly larger than that on drag, and the relationship between peak value of lift, and drag, and frontal gust speed are approximately linear.

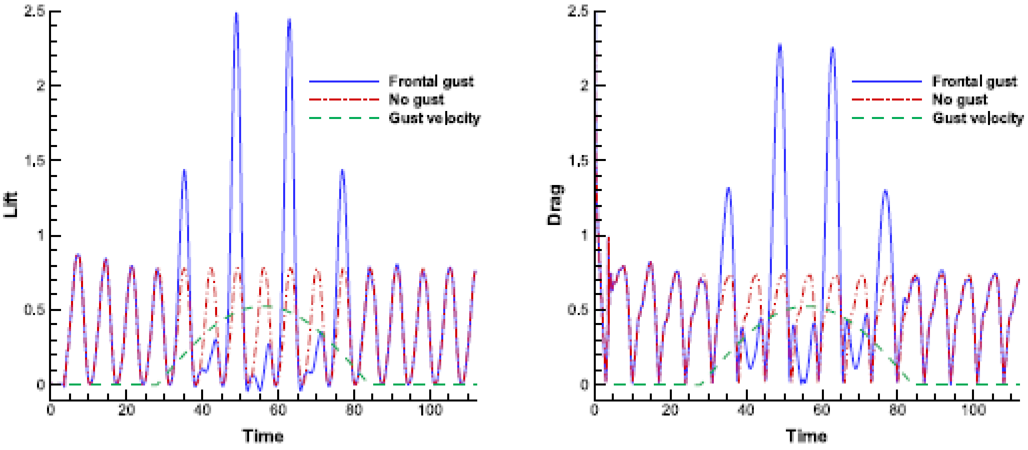


Figure 3.17: Effects on flapping-wing lift and drag forces by frontal gusts.

Comparing with Figure 3.17, effects brought by lateral gusts are much less significant, as shown in Figure 3.18. Both lift and drag increase when the speed of lateral gust encountered increases. However, the effect on lift is much smaller than that on drag in this case of lateral gust.

In summary, both research shows that when encountering frontal and lateral gusts, lift and drag generated by flapping wings will increase. But considering the size difference of experimental FWMAV/flapping-wing platform, the magnitude of effect/change on forces is quite different. Therefore, it’s quite obvious that in this project both studying the gust effects on Flapper Drone and accurately modelling in-gust dynamics are necessary steps before control strategy design.

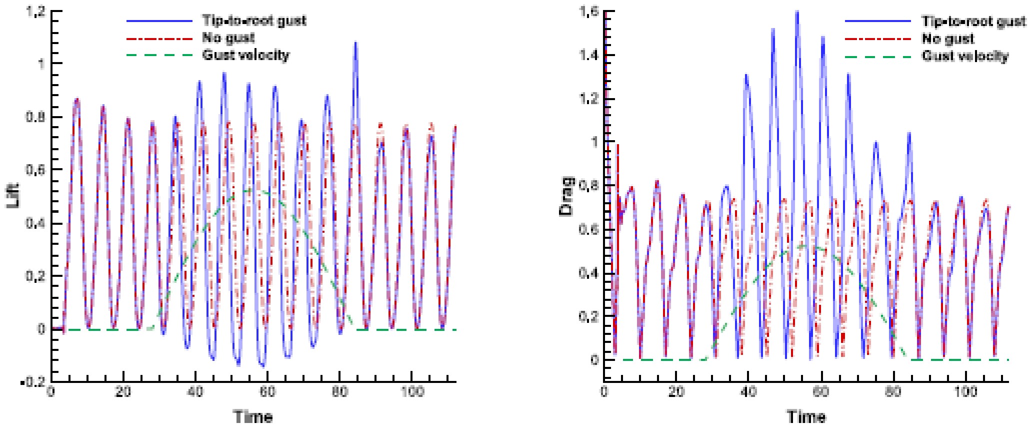


Figure 3.18: Effects on flapping-wing lift and drag forces by lateral gusts.

3.3. Dynamics Modelling of FWMAV/Drones

As discussed in 3.2, it is necessary to model the in-gust dynamics of FWMAVs before further steps of control strategy design. In previous thesis[23, 21, 3, 29], both longitudinal and lateral dynamics of DelFly Nimble have been studied, and several models for DelFly Nimble are developed from system identification (black-box models) and parameter estimation (white-box models). However, these models are not complete for this project because no in-gust dynamics are included. Furthermore, due to the high linearity in in-gust dynamics of FWMAV, it is more practical to model in-gust dynamics with ML/NN-based methods, which are strong tools for global nonlinear function approximation. In subsection 3.3.1, DelFly Nimble's models already existed are introduced, and ML/NN-based dynamics modelling of drones are introduced in subsection 3.3.2.

3.3.1. Dynamic models of DelFly Nimble

The modelling of FWMAV dynamics mostly requires initial proposal on model structure, parameter estimation and model validation. Depending on the accuracy and complexity of the model required, the model could be derived from fundamental dynamic analysis or from high-level approximation on data collected during flight experiments.

In [21], a minimal longitudinal dynamic model of DelFly Nimble is proposed and verified for control design purpose. The stroke-averaged drag force is modelled based on the assumption that flapping wings flap with a saw-tooth profile and its center of pressure is located at half the wing length. The constant wing velocity is expressed as Equation 3.1.

$$U = 2\Phi f \frac{b}{4} \tag{3.1}$$

where Φ is the flapping amplitude in radians, f is the flapping frequency and b is the wingspan.

Based on this, when the wing velocity due to flapping is higher than the free-stream velocity, the drag force during the downstroke and upstroke is modelled as Equation 3.2 and Equation 3.3 respectively with u as free-stream velocity, β as force coefficients.

$$f_d = -\beta(U + u)^2 \tag{3.2}$$

$$f_d = \beta(U - u)^2 \tag{3.3}$$

Since these two strokes have an equal time duration, the average force over a flapping cycle could be expressed as Equation 3.4.

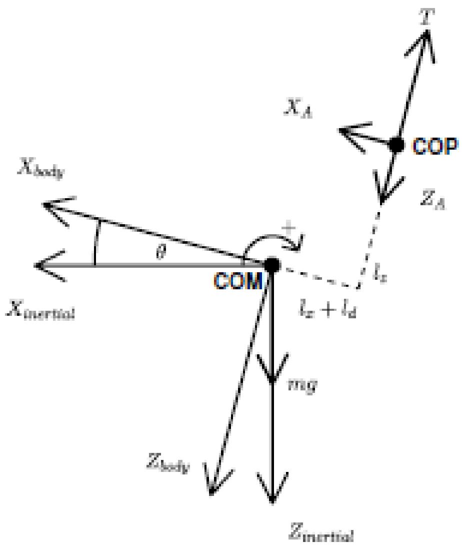


Figure 3.19: Longitudinal dynamics diagram of DelFly Nimble. COM - mean center of mass.

$$\overline{f_d} = \frac{1}{2}\beta[(U-u)^2 - (U+u)^2] = -2\beta U u \tag{3.4}$$

Based on above, the model for longitudinal dynamics of DelFly Nimble is given as Equation 3.5 to Equation 3.9 and illustrated in Figure 3.19.

$$u_{COP} = u - l_z\dot{\theta} - \dot{l_d} \tag{3.5}$$

$$w_{COP} = w - (l_d + l_z)\dot{\theta} \tag{3.6}$$

$$X = -2b_x u_{COP} \tag{3.7}$$

$$Z = -2T - 2b_z w_{COP} \tag{3.8}$$

$$M = -Xl_z + Z(l_d + l_z) \tag{3.9}$$

In [23], based on Equation 3.4, a complete white-box model for DelFLy Nimble is proposed. The nonlinear EoM of the rigid body dynamics are Figure 3.20. And the forces and torques are modelled as Figure 3.20, and the frames and forces are illustrated in Figure 3.21.

However, for the platform of this project, Flapper Drone, the force coefficients and the relationship between thrust and flapping frequency is still to-be-identified by flight experiments and parameter estimation techniques.

$$
\begin{aligned}
f'_{b,x} &= D_{l,x} = -d_{l,x} f_l\left(u - l_{l,z}q + l_{l,y}r + \dot{l}_{l,x}\right)\\
f'_{b,y} &= D_{l,y} = -d_{l,y} f_l\left(v + l_{l,x}r + l_{l,z}p + \dot{l}_{l,y}\right)\\
f'_{b,z} &= D_{l,z} - F_L(f_l) = -d_{l,z} f_l\left(w - l_{l,x}q - l_{l,y}p + \dot{l}_{r,z},\right) - F_L(f_l)
\end{aligned}
$$

$$
\begin{aligned}
\tau'_{b,x} &= D_{l,y} l_{l,z} - \left(D_{l,z} - F_L(f_l)\right) l_{l,y}\\
&= -d_{l,y} f_l(v + l_{l,x}r + l_{l,z}p + l_{l,y}) l_{l,z} - \left(-d_{l,z} f_l(w - l_{l,x}q - l_{l,y}p + l_{r,z}) - F_L(f_l)\right) l_{l,y}\\
\tau'_{b,y} &= -D_{l,x} l_{l,z} - \left(D_{l,z} - F_L(f_l)\right) l_{l,x}\\
&= d_{l,x} f_l(u - l_{l,z}q + l_{l,y}r + l_{l,x}) l_{l,z} - \left(-d_{l,z} f(w - l_{l,x}q - l_{l,y}p + l_{r,z}) - F_L(f_l)\right) l_{l,x}\\
\tau'_{b,z} &= D_{l,y} l_{l,x} + D_{l,x} l_{l,y}\\
&= -d_{l,y} f_l(v + l_{l,x}r + l_{l,z}p + l_{l,y}) l_{l,x} - d_{l,x} f_l(u - l_{l,z}q + l_{l,y}r + l_{l,x}) l_{l,y}
\end{aligned}
$$

Figure 3.20: Force and torque model in body frame of DelFly Nimble

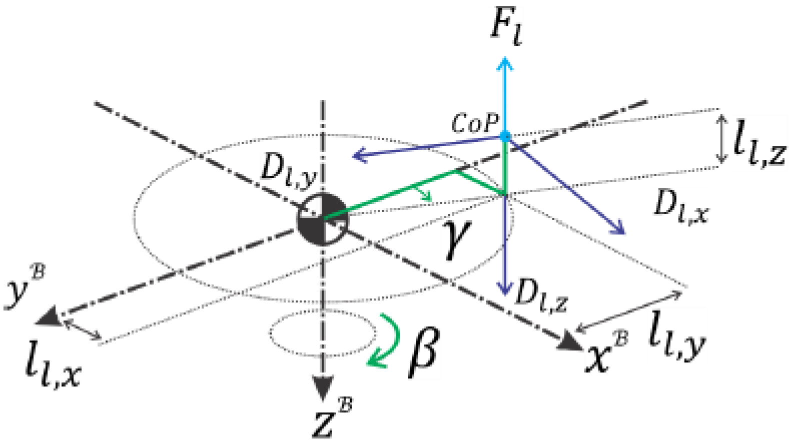


Figure 3.21: Overall view of DelFly Nimble with forces of left wing as example. The force acting on COP is decomposed in drag forces in directions in body frame (purple) and thrust forces.

3.3.2. ML/NN-based modelling of drones

As discussed in subsection 3.2.2 and subsection 3.3.1, the in-gust dynamics of a FWMAV is highly-nonlinear and hard to model with classic system identification methods. Being another popular flight platform, the dynamics of quadrotors are also highly-nonlinear and include coupling effects between different motions. On the other hand, neural networks are strong tools for global nonlinear function approximation. In [4], a ML/NN -based modelling method is proposed for flight dynamic model of quadrotors. With current states and control inputs as the input of a Rectified Linear Unit (ReLU) neural network shown in Figure 3.22, the network was trained on real flight data and could predict states of next step accurately. The model accuracy is shown in Figure 3.23.

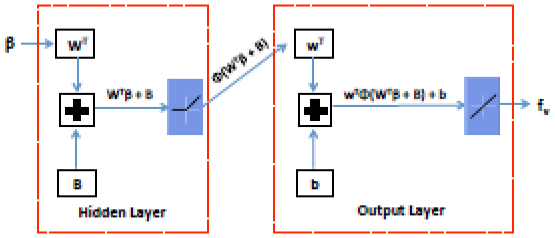


Figure 3.22: The ReLU NN architecture used to learn quadrotor's dynamics

Furthermore, in [4], it was shown that NN could generalize and learn nonlinear dynamic couplings between translational and rotational motions, even when the training data hasn't captured these coupling effects significantly. And in [36], a stable drone landing control strategy was designed based on NN-

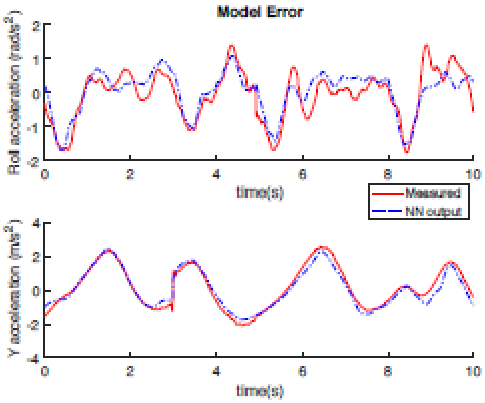


Figure 3.23: Comparison between real flight data and NN outputs.

learned dynamic model and verified in reality. Therefore, in this project, ML/NN-based methods would be employed to model the high-nonlinear in-gust dynamics of Flapper Drone and the learned model would also be used for further step of control strategy design after being validated.

3.4. Control of FWMAV

After modelling of in-gust dynamics of Flapper Drone, the following step as control strategy design is also an essential step for improving robustness and energy-efficiency of in-gust flights. In subsection 3.4.1, classic control strategies for FWMAV are introduced. And in subsection 3.4.2, the ML/RL-based control strategy for FWMAV is introduced.

3.4.1. Classic control

J. Kelbling's work in DelFly Nimble control

In [23], nonlinear geometric control is implemented for attitude control and position control, as shown in 3.24. The attitude error is defined as $R_d^T R$ and the an attitude error function is defined as Equation 3.10, which represents the Euclidean distance of $R_d^T R$.

$$\Psi(R, R_d) = \frac{1}{2} tr[I - R_d^T R] \tag{3.10}$$

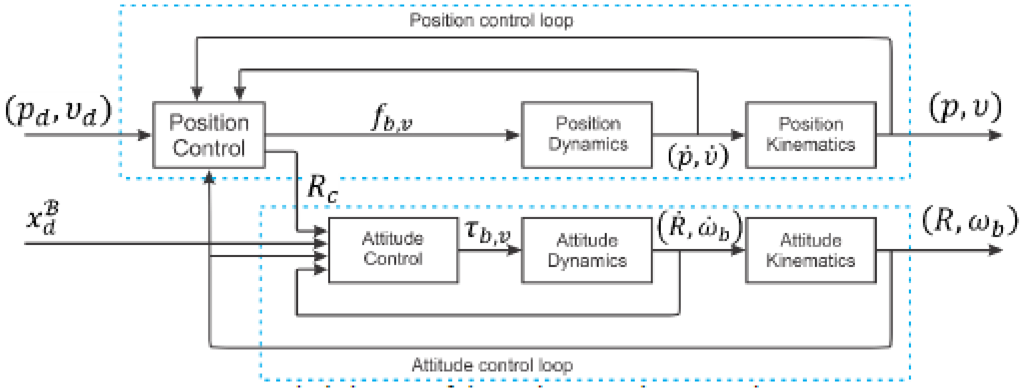


Figure 3.24: Block diagram of the nonlinear tracking control structure.

Based on Equation 3.10, the tracking error for the angular velocity is selected as Equation 3.11, and the tracking error of the attitude is selected as Equation 3.12.

$$e_{\omega_b} = \omega_b - R_d^T R \omega_{b_d} \tag{3.11}$$

$$e_R = \frac{1}{2}(R_d^T R - R^T R_d) \tag{3.12}$$

After substituting the nonlinear control input defined in Equation 3.13, the attitude error dynamics are given as Equation 3.14. With positive gains, the error could be eliminated theoretically.

$$\tau_{b,v} = -k_R e_R - k_{\omega_b} e_{\omega_b} + \omega_b \times J\omega_b - J(\omega_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega_{b,d}}) \tag{3.13}$$

$$J\dot{e_{\omega_b}} = -k_R e_R - k_{\omega_b} e_{\omega_b} \tag{3.14}$$

Chirarattananon et al. work in disturbance rejection control of Robobee.

In [8], based on the gust effect model in Figure 3.16 and the adaptive tracking flight controller in [9], adaptive estimation in Equation 3.15 and least-square estimation in Equation 3.16 methods for estimating disturbance torques are applied on Robobee.

$$\tau_w = \begin{bmatrix} -\hat{y}^T & \mathbf{0} \\ \mathbf{0} & \hat{x}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} a_x \mathbf{v} \\ a_y \mathbf{v} \end{bmatrix} \tag{3.15}$$

$$\tau[t_{i-1}] = \frac{1}{1-\gamma}(\Psi[t_i] - \gamma\Psi[t_{i-1}]) \tag{3.16}$$

where Ψ stands for results after passing a low-pass filter.

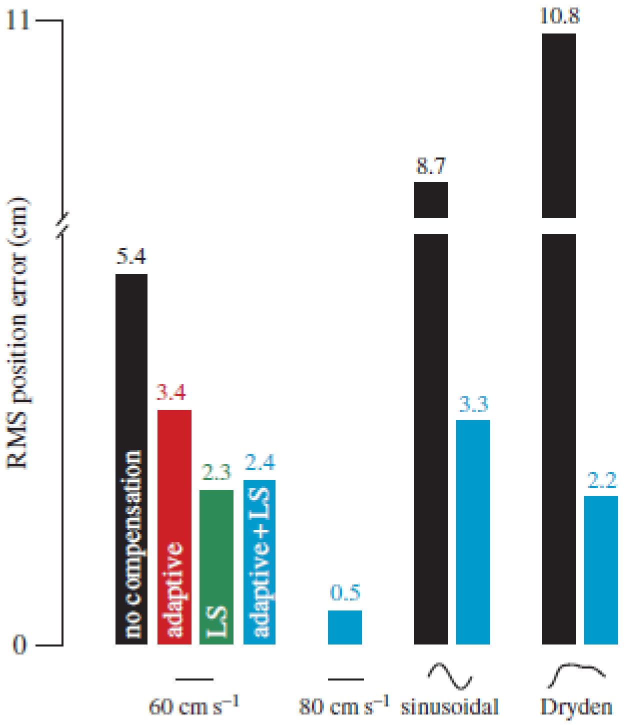


Figure 3.25: Position error comparison from all experiment set-ups.

As shown in Figure 3.25, the position error with adaptive/LS-estimation is much lower than the original case without any compensation. Furthermore, [8] admitted it was still yet to find out the reason that single LS estimation preformed even slightly better than combination (adaptive + LS) cases.

Lee et al. work in disturbance rejection control of FWMAV

In [24], disturbance-observer-based control (DOBC) is applied to the control of a simple FWMAV. Their work is mainly to estimate the disturbance and add a feed-forward control term in the inner loop of the nominal controller. The disturbance observer is defined as Equation 3.18.

$$\dot{z} = -\frac{\partial p}{\partial x}(x)g_2(x)z - \frac{\partial p}{\partial x}(x)(f(x) + g_1(x) + g_2(x)p(x)) \tag{3.17}$$

$$\hat{d} = z + p(x) \tag{3.18}$$

With disturbance observer, a flight mode selector is designed and will be activated when the output of disturbance observer reaches the critical value. Once activated, the feed-forward control term is added in to the control input to reject disturbances encountered.

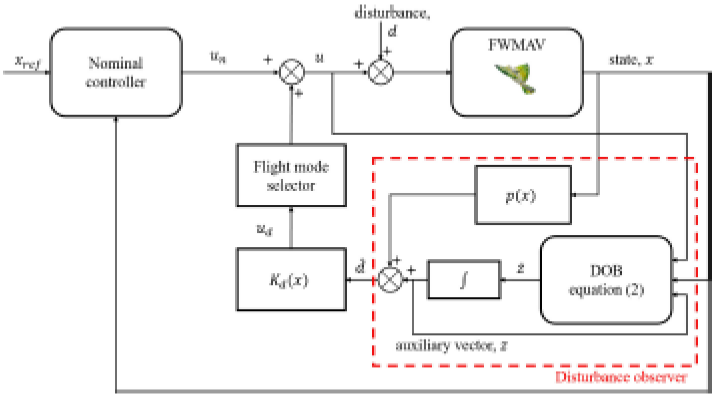


Figure 3.26: Block diagram of DOBC control for the FWMAV.

The experiment flight paths are defined as Figure 3.27. Right after encountering disturbance in red area, the flight mode selector is activated. The result comparison is shown in Figure 3.28. The oscillation and tracking errors of FWMAV is obviously smaller than nominal case without compensation.

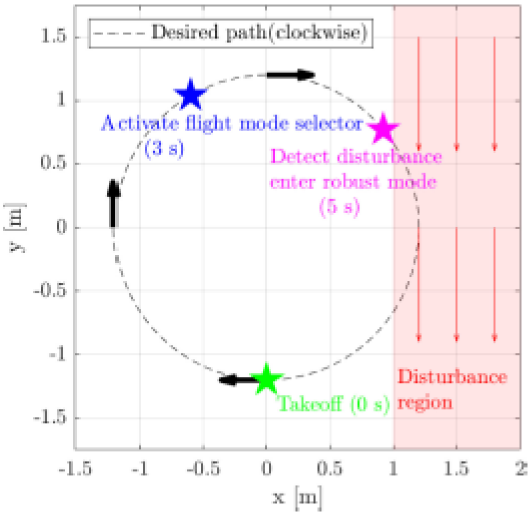


Figure 3.27: Experiment flight path for DOBC validation.

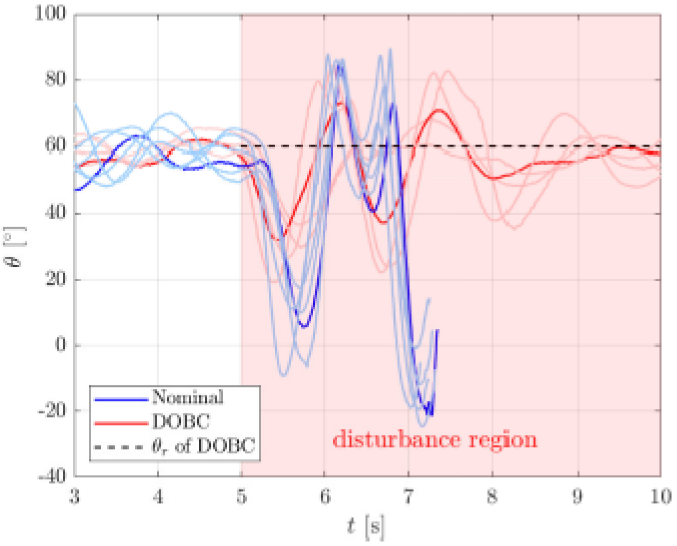
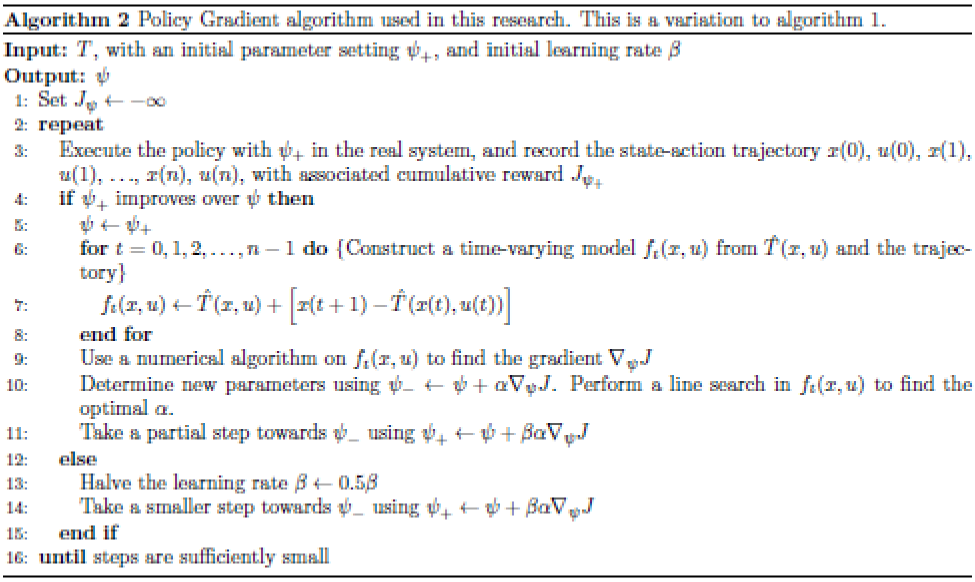


Figure 3.28: Result comparison for DOBC validation.

3.4.2. ML/RL-based control

Goedhart's work in ML/RL-based control of DelFly II

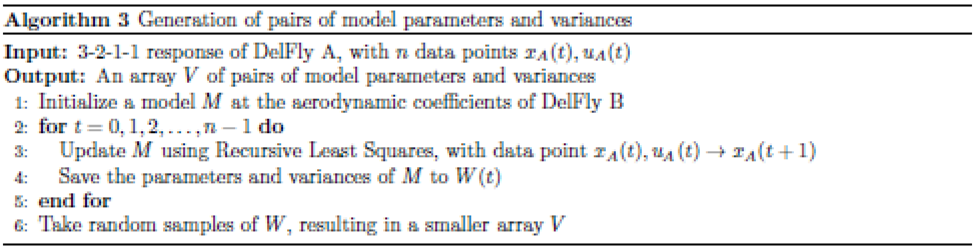
In [17], two ML/RL-based methods are applied for tuning gains of PI controller in airspeed controller. Firstly, Policy Gradient algorithm in RL is implemented as shown in Figure 3.29 in the actor-only form. Secondly, the Classification Algorithm in ML is applied for selecting pairs of gains in the predefined gain set, as shown in Figure 3.30.



```
Algorithm 2 Policy Gradient algorithm used in this research. This is a variation to algorithm 1.
Input: $T$, with an initial parameter setting $\psi_+$, and initial learning rate $\beta$
Output: $\psi$
Set $J_\psi \leftarrow -\infty$
repeat
  Execute the policy with $\psi_+$ in the real system, and record the state-action trajectory $x(0)$, $u(0)$, $x(1)$, $u(1)$, ..., $x(n)$, $u(n)$, with associated cumulative reward $J_{\psi_+}$
  if $\psi_+$ improves over $\psi$ then
    $\psi \leftarrow \psi_+$
    for $t = 0, 1, 2, \dots, n-1$ do {Construct a time-varying model $f_t(x,u)$ from $\hat{T}(x,u)$ and the trajectory}
      $f_t(x,u) \leftarrow \hat{T}(x,u) + \left[x(t+1) - \hat{T}(x(t), u(t))\right]$
    end for
    Use a numerical algorithm on $f_t(x,u)$ to find the gradient $\nabla_\psi J$
    Determine new parameters using $\psi_- \leftarrow \psi + \alpha \nabla_\psi J$. Perform a line search in $f_t(x,u)$ to find the optimal $\alpha$.
    Take a partial step towards $\psi_-$ using $\psi_+ \leftarrow \psi + \beta\alpha\nabla_\psi J$
  else
    Halve the learning rate $\beta \leftarrow 0.5\beta$
    Take a smaller step towards $\psi_-$ using $\psi_+ \leftarrow \psi + \beta\alpha\nabla_\psi J$
  end if
until steps are sufficiently small
```

Figure 3.29: Policy gradient algorithm for PI tuning

Baseline tracking tests were conducted for comparison of rewards per episode with different methods, as the results shown in Figure 3.31. In PI cases, the score of PG algorithm is slightly higher than two other cases. In cascaded PI cases, the performance of both PG and CA looks comparable to fixed-gain cases, but the tracking accuracy is actually much better than fixed-gain cases.



```
Algorithm 3 Generation of pairs of model parameters and variances
Input: 3-2-1-1 response of DelFly A, with $n$ data points $x_A(t), u_A(t)$
Output: An array $V$ of pairs of model parameters and variances
Initialize a model $M$ at the aerodynamic coefficients of DelFly B
for $t = 0, 1, 2, \ldots, n-1$ do
    Update $M$ using Recursive Least Squares, with data point $x_A(t), u_A(t) \rightarrow x_A(t+1)$
    Save the parameters and variances of $M$ to $W(t)$
end for
Take random samples of $W$, resulting in a smaller array $V$
```

Figure 3.30: Classification algorithm for PI gain selection.

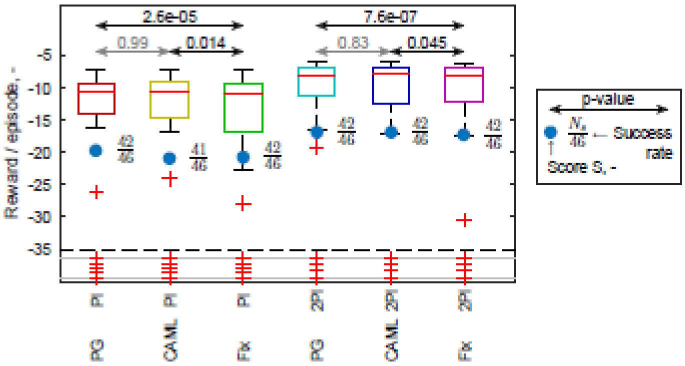


Figure 3.31: Comparison of rewards per episode with different methods

Fei et al. work in RL-based control of Flappy Hummingbird

In [13], an open source dynamic simulator for Flappy Hummingbird is introduced based on identified dynamic model of this FWMAV platform. With this simulator, Flappy Hummingbird has learned extreme hummingbird maneuvers with help of Deep Deterministic Policy Gradient (DDGP) algorithm in RL.

The proposed control strategy is a hybrid one which combines model-based nonlinear control with model-free RL algorithm, as shown in Figure 3.32. The model-based nonlinear controller could guarantee flight stability and the model-free maneuvering policy learns to destabilize the system for extreme agile maneuvers.

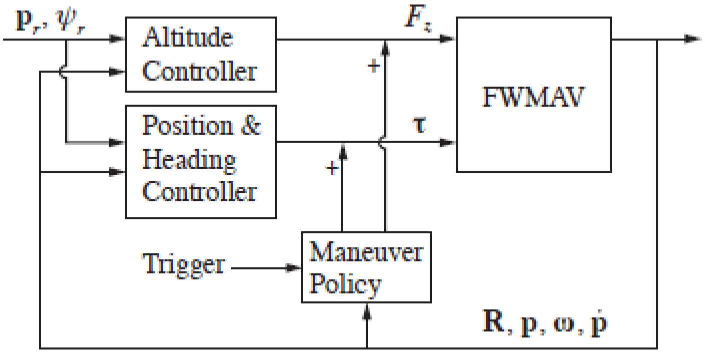

$\mathbf{p}_r, \psi_r$
Altitude Controller
Position & Heading Controller
F_z
+
τ
+
FWMAV
Trigger
Maneuver Policy
$\mathbf{R}, \mathbf{p}, \boldsymbol{\omega}, \dot{\mathbf{p}}$

Figure 3.32: Control architecture for Flappy Hummingbird.

Based on training in simulator and real transfer, Flappy Hummingbird has learned the extreme evasive maneuver of hummingbird. The real flight of hummingbird is shown in Figure 3.33, with the learned maneuvers in simulation as Figure 3.34 and real maneuvers conducted by Flappy Hummingbird as Figure 3.35.

In this project, based on literature study, the control strategy could be a combination of a controller for

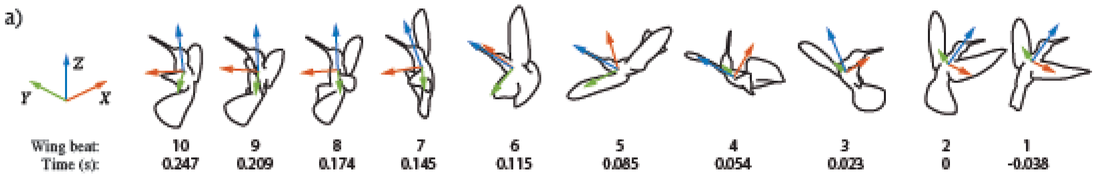


Figure 3.33: Maneuvers of hummingbirds

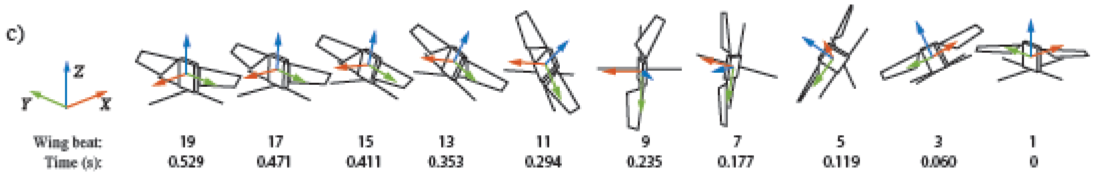


Figure 3.34: Control architecture for Flappy Hummingbird.

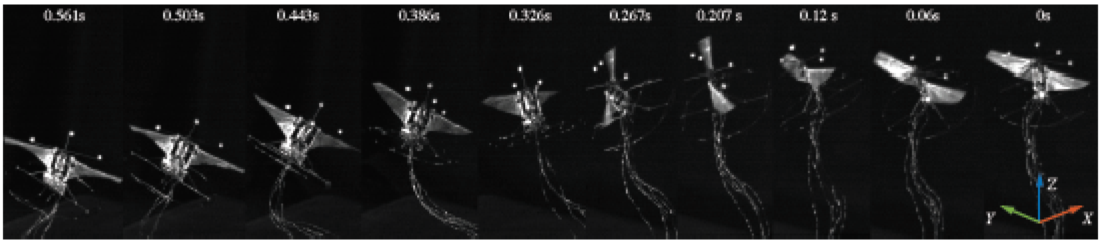


Figure 3.35: Control architecture for Flappy Hummingbird.

in-gust stable hovering and another controller for additional control input in in-gust way-point following task.

3.5. Conclusion

The conclusions from the literature study are the research questions and objectives of this project.

3.5.1. Research question

The findings of literature study in previous chapters are concluded as:

- The dynamic models of DelFly Nimble existing so far are not complete and adequate for this project, due to the reason that no in-gust dynamics are included.
- ML/NN algorithms are strong tools for nonlinear function approximation and the response speed of ML/NN-based controllers are faster than most classic controllers.

Based on these findings, the research question if ML/NN methods are applied on in-gust dynamic modelling and control strategy design, and airflow data from the sensor is utilized in control system, could the robustness and energy-efficiency of in-gust flights of Flapper Drone be improved and how much could we improve it?

3.5.2. Research objectives

From this research question, the research objectives of this project are as follows:

- To create a more accurate model of Flapper Drone in gust dynamic conditions.
- To use this model in design of controller with ML method, for in-gust flight control of Flapper Drone.
- To validate the new controller in gust dynamic conditions to examine the robustness (deviation from desired position and real trajectory, response speed and repeatability) and energy-efficiency(battery life/flight time) of Flapper Drone's in-gust flights.
- To compare validation results with current cases with OptiTrack/simple on-board PID controllers.

References

[1] Natalie Agre et al. “Linear drag law for high-Reynolds-number flow past an oscillating body”. In: *Physical Review Fluids* 1.3 (2016), p. 033202.

[2] David E Alexander. *Nature’s flyers: birds, insects, and the biomechanics of flight*. JHU Press, 2002.

[3] Karan Bains. “System Identification of the Delfly Nimble: Modeling of the Lateral Body Dynamics”. In: (2020).

[4] Somil Bansal et al. “Learning quadrotor dynamics using neural network for flight control”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 4653–4660.

[5] Lina Castano et al. “Gust rejection using force adaptive feedback for roll”. In: *14th AIAA Aviation Technology, Integration, and Operations Conference*. 2014, p. 2588.

[6] Pakpong Chirarattananon, Kevin Y Ma, and Robert J Wood. “Adaptive control of a millimeter-scale flapping-wing robot”. In: *Bioinspiration & biomimetics* 9.2 (2014), p. 025004.

[7] Pakpong Chirarattananon, Kevin Y Ma, and Robert J Wood. “Fly on the wall”. In: *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. IEEE. 2014, pp. 1001–1008.

[8] Pakpong Chirarattananon et al. “Dynamics and flight control of a flapping-wing robotic insect in the presence of wind gusts”. In: *Interface focus* 7.1 (2017), p. 20160080.

[9] Pakpong Chirarattananon et al. “Wind disturbance rejection for an insect-scale flapping-wing robot”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 60–67.

[10] Gangik Cho, Jongyun Kim, and Hyondong Oh. “Vision-based obstacle avoidance strategies for mavs using optical flows in 3-d textured environments”. In: *Sensors* 19.11 (2019), p. 2523.

[11] Michael H Dickinson, Fritz-Olaf Lehmann, and Sanjay P Sane. “Wing rotation and the aerodynamic basis of insect flight”. In: *Science* 284.5422 (1999), pp. 1954–1960.

[12] Robert Dudley. *The biomechanics of insect flight: form, function, evolution*. Princeton University Press, 2002.

[13] Fan Fei et al. “Flappy hummingbird: An open source dynamic simulation of flapping wing robots and animals”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 9223–9229.

[14] Fan Fei et al. “Learning extreme hummingbird maneuvers on flapping wing robots”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 109–115.

[15] Steven N Fry, Rosalyn Sayaman, and Michael H Dickinson. “The aerodynamics of free-flight maneuvers in Drosophila”. In: *Science* 300.5618 (2003), pp. 495–498.

[16] Sawyer Buckminster Fuller et al. “Flying Drosophila stabilize their vision-based velocity controller by sensing wind with their antennae”. In: *Proceedings of the National Academy of Sciences* 111.13 (2014), E1182–E1191.

[17] Menno Goedhart et al. “Machine learning for flapping wing flight control”. In: *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*. 2018, p. 2135.

[18] Steven Ho et al. “Unsteady aerodynamics and flow control for flapping wing flyers”. In: *Progress in aerospace sciences* 39.8 (2003), pp. 635–681.

[19] Hua Huang and Mao Sun. “Dragonfly forewing-hindwing interaction at various flight speeds and wing phasing”. In: *AIAA journal* 45.2 (2007), pp. 508–511.

[20] Martin Jones and Nail K Yamaleev. “Effect of lateral, downward, and frontal gusts on flapping wing performance”. In: *Computers & Fluids* 140 (2016), pp. 175–190.

[21] Karl Martin Kajak et al. “A minimal longitudinal dynamic model of a tailless flapping wing robot for control design”. In: *Bioinspiration & biomimetics* 14.4 (2019), p. 046008.

[22] Matěj Karásek et al. “A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns”. In: *Science* 361.6407 (2018), pp. 1089–1094.

[23] Jelle Kelbling. “Trajectory control and dynamic modeling of a tailless flapping-wing robot”. In: (2020).

[24] Jonggu Lee, Seungwan Ryu, and H Jin Kim. “Stable flight of a flapping-wing micro air vehicle under wind disturbance”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5685–5692.

[25] Kevin Y Ma, Samuel M Felton, and Robert J Wood. “Design, fabrication, and modeling of the split actuator microrobotic bee”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 1133–1140.

[26] Kevin Y Ma et al. “Controlled flight of a biologically inspired, insect-scale robot”. In: *Science* 340.6132 (2013), pp. 603–607.

[27] A. Martín-Alcántara et al. “A Simple Model for Gliding and Low-Amplitude Flapping Flight of a Bio-Inspired UAV”. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2019, pp. 729–737. DOI: 10.1109/ICUAS.2019.8798233.

[28] A Martín-Alcántara et al. “A simple model for gliding and low-amplitude flapping flight of a bio-inspired UAV”. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2019, pp. 729–737.

[29] Jorgen Nijboer et al. “Longitudinal Grey-Box Model Identification of a Tailless Flapping-Wing MAV Based on Free-Flight Data”. In: *AIAA Scitech 2020 Forum*. 2020, p. 1964.

[30] Michael O’Connell et al. “Neural-Fly enables rapid learning for agile flight in strong winds”. In: *Science Robotics* 7.66 (2022), eabm6597.

[31] MA Pérez-Cutiño et al. “Neural networks algorithms for ornithopter trajectory optimization** This work is partially supported by the Spanish Ministry of Economy and Competitiveness (MTM2016-76272-R AEI/FEDER, UE) and European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement# 734922.” In: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2021, pp. 1665–1670.

[32] Hoang Vu Phan, Taesam Kang, and Hoon Cheol Park. “Design and stable flight of a 21 g insect-like tailless flapping wing micro air vehicle with angular rates feedback control”. In: *Bioinspiration & biomimetics* 12.3 (2017), p. 036006.

[33] Hoang Vu Phan et al. “Towards the Long-Endurance Flight of an Insect-Inspired, Tailless, Two-Winged, Flapping-Wing Flying Robot”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5059–5066. DOI: 10.1109/LRA.2020.3005127.

[34] Fabio Rodríguez et al. “Kinodynamic planning for an energy-efficient autonomous ornithopter”. In: *Computers & Industrial Engineering* (2021), p. 107814.

[35] Sanjay P Sane. “The aerodynamics of insect flight”. In: *Journal of experimental biology* 206.23 (2003), pp. 4191–4208.

[36] Guanya Shi et al. “Neural lander: Stable drone landing control using learned dynamics”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 9784–9790.

[37] Wei Shyy and Hao Liu. “Flapping wings and aerodynamic lift: the role of leading-edge vortices”. In: *AIAA journal* 45.12 (2007), pp. 2817–2819.

[38] Wei Shyy et al. *An introduction to flapping wing aerodynamics*. Vol. 37. Cambridge University Press, 2013.

[39] Alex Skordos et al. “A novel strain sensor based on the campaniform sensillum of insects”. In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 360.1791 (2002), pp. 239–253.

[40] Alejandro Suarez et al. “Small-Scale Compliant Dual Arm with Tail for Winged Aerial Robots”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 208–214.

[41] Graham K Taylor. “Mechanics and aerodynamics of insect flight control”. In: *Biological Reviews* 76.4 (2001), pp. 449–471.

[42] Zhan Tu et al. “An at-scale tailless flapping-wing hummingbird robot. I. Design, optimization, and experimental validation”. In: *IEEE Transactions on Robotics* 36.5 (2020), pp. 1511–1525.

[43] Sunyi Wang et al. “Battle the Wind: Improving Flight Stability of a Flapping Wing Micro Air Vehicle Under Wind Disturbance With Onboard Thermistor-Based Airflow Sensing”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 9605–9612.

[44] PC Wilkins and K Knowles. “The leading-edge vortex and aerodynamics of insect-based flapping-wing micro air vehicles”. In: *The Aeronautical Journal* 113.1142 (2009), pp. 253–262.

[45] Robert Wood, Radhika Nagpal, and Gu-Yeon Wei. “Flight of the robobees”. In: *Scientific American* 308.3 (2013), pp. 60–65.

[46] Robert J Wood. “The first takeoff of a biologically inspired at-scale robotic insect”. In: *IEEE transactions on robotics* 24.2 (2008), pp. 341–347.

[47] Robert J Wood, E Steltz, and RS Fearing. “Optimal energy density piezoelectric bending actuators”. In: *Sensors and Actuators A: Physical* 119.2 (2005), pp. 476–488.

[48] Hongjiu Yang et al. “Active disturbance rejection attitude control for a dual closed-loop quadrotor under gust wind”. In: *IEEE Transactions on control systems technology* 26.4 (2017), pp. 1400–1405.

[49] Dong-Wan Yoo, Dae-Yeon Won, and Min-Jea Tahk. “Optical flow based collision avoidance of multi-rotor uavs in urban environments”. In: *International Journal of Aeronautical and Space Sciences* 12.3 (2011), pp. 252–259.

[50] Jian Zhang et al. “Design optimization and system integration of robotic hummingbird”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5422–5428.

A

Extra Hardware Added to the FWMAV

A.1. Schematics of the connection

In this thesis, an airflow sensor, a current sensor and a $9\ V$ voltage regulator are added to the Flapper Drone. The schematic which shows the connection of the airflow sensor and the current sensor to the Flapper Drone's Crazyflie expansion deck is as follow.

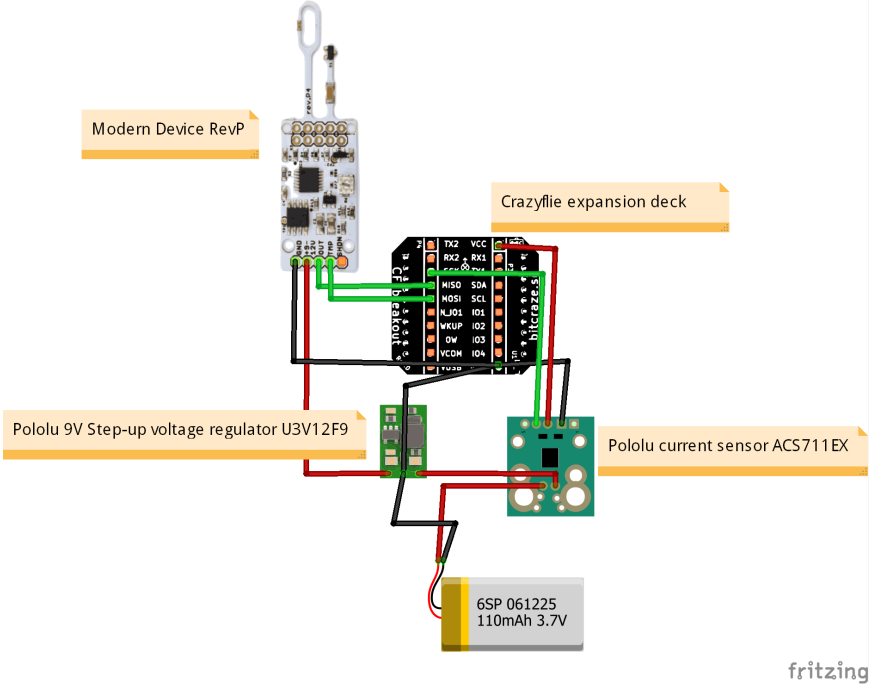


Figure A.1: The schematic of the connection of the airflow sensor and the current sensor to the Flapper Drone's Crazyflie expansion deck.

The specifications of the sensors and the voltage regulator are as Table A.1.

Hardware type	Model	Supply voltage [V]	Output
Airflow sensor	Rev.P6	$9.0 - 12.0$	$\leq 3.3\ V$
Current Sensor	Pololu ACS711EX	$3.0 - 5.5$	$\leq 31.0\ A$
Step-Up Voltage Regulator	Pololu U3V12F9	$2.5 - 9.0$	$9.0\ V$

Table A.1: The specifications of the sensors and the voltage regulator

A.2. Drivers of the sensor

The drivers for the airflow sensor is shown as follow.

```c
/**
Airflow Sensor Driver
*/

#define DEBUG_MODULE "AIRFLOW"

#include "FreeRTOS.h"
#include "task.h"
#include "stabilizer_types.h"
#include "deck.h"
#include "system.h"
#include "debug.h"
#include "log.h"
#include "param.h"
#include "airflowdeck.h"

static float wind_volts_last = 0;
static float airspeed_last = 0;

static bool isInit;

void airflowDeckInit(DeckInfo* info)
{
  if (isInit)
    return;

  xTaskCreate(airflowDeckTask, AIRFLOWDECK_TASK_NAME, AIRFLOWDECK_TASK_STACKSIZE, NULL, AIRFLOWDECK_TASK_PRI, NULL);

  isInit = true;
}

bool airflowDeckTest()
{
  bool testStatus;
  testStatus = true;

  if (!isInit)
    return false;

  return testStatus;
}

void airflowDeckTask(void* arg)
{
  systemWaitStart();
  TickType_t xLastWakeTime;

  xLastWakeTime = xTaskGetTickCount();

  while (1) {
    vTaskDelayUntil(&xLastWakeTime, M2T(1));

    wind_volts_last = analogReadVoltage(DECK_GPIO_MISO);
    flowvolt.volt = wind_volts_last;
    airspeed_last = 25.8666354823914f*wind_volts_last*wind_volts_last*wind_volts_last*wind_volts_last-1.664910993036515e2f*wind_volts_last*wind_volts_last*wind_volts_last+4.030483719450837e2f*wind_volts_last*wind_volts_last-4.325309182694595e2f*wind_volts_last+1.730907713055474e2f;
  }
}

static const DeckDriver airflow_deck = {
  .vid = 0xBC,
  .pid = 0x02,
  .name = "bcAirflowDeck",
  .usedGpio = DECK_USING_PA6,

  .init = airflowDeckInit,
  .test = airflowDeckTest,
};

DECK_DRIVER(airflow_deck);

PARAM_GROUP_START(deck)

PARAM_ADD_CORE(PARAM_UINT8 | PARAM_RONLY, airflowDeck, &isInit)
PARAM_GROUP_STOP(deck)

LOG_GROUP_START(airflow)
LOG_ADD(LOG_FLOAT, v_wind_ext, &wind_volts_last)
LOG_ADD(LOG_FLOAT, airspeed_ext, &airspeed_last)
LOG_GROUP_STOP(airflow)
```

The drivers for the current sensor is shown as follow.

```
/**
Current Sensor Driver
*/

#define DEBUG_MODULE "CURRENT"

#include "FreeRTOS.h"
#include "task.h"
#include "deck.h"
#include "system.h"
#include "debug.h"
#include "log.h"
#include "param.h"
#include "currentdeck.h"

static float reading_last = 0;
static float current_last = 0;
static float current = 0;

static bool isInit;

void currentDeckInit(DeckInfo* info)
{
  if (isInit)
    return;

  xTaskCreate(currentDeckTask, CURRENTDECK_TASK_NAME, CURRENTDECK_TASK_STACKSIZE, NULL,
      CURRENTDECK_TASK_PRI, NULL);

  isInit = true;

  DEBUG_PRINT("Current deck initialization is done.\n");
}

bool currentDeckTest(void)
{
  bool testStatus;
  testStatus = true;

  DEBUG_PRINT("Current deck test is done.\n");

  if (!isInit)
    return false;

  return testStatus;
}

void currentDeckTask(void* arg)
{
  systemWaitStart();
  TickType_t xLastWakeTime;

  xLastWakeTime = xTaskGetTickCount();

  while (1) {
    vTaskDelayUntil(&xLastWakeTime, M2T(1));

    reading_last = analogReadVoltage(DECK_GPIO_SCK);
    current_last = 36.7f*reading_last/3.0f-18.3f;

    current = 0.975f*current + 0.025f*current_last;
  }
}

static const DeckDriver current_deck = {
  .vid = 0xBC,
  .pid = 0x09,
  .name = "bcCurrentDeck",
  .usedGpio = DECK_USING_PA5,

  .init = currentDeckInit,
  .test = currentDeckTest,
};

DECK_DRIVER(current_deck);

PARAM_GROUP_START(deck)

PARAM_ADD_CORE(PARAM_UINT8 | PARAM_RONLY, bcCurrentDeck, &isInit)
PARAM_GROUP_STOP(deck)

LOG_GROUP_START(current)
LOG_ADD(LOG_FLOAT, v_raw, &reading_last)
LOG_ADD(LOG_FLOAT, i_raw, &current_last)
LOG_ADD(LOG_FLOAT, current, &current)
LOG_GROUP_STOP(current)
```

B

Previous Design of Adaptive Controller

B.1. Airflow-sensing based adaptive feedforward velocity controller

Prior to the design of the adaptive position and velocity controller presented in 2, the adaptive controller was designed as only adaptive part in the feedforward block of the velocity controller, but with more specified intervals and corresponding K_{FF_x} gains. The overall diagram of the flight control system and the diagram of the adaptive feedforward velocity controller are shown in B.1 and B.2.

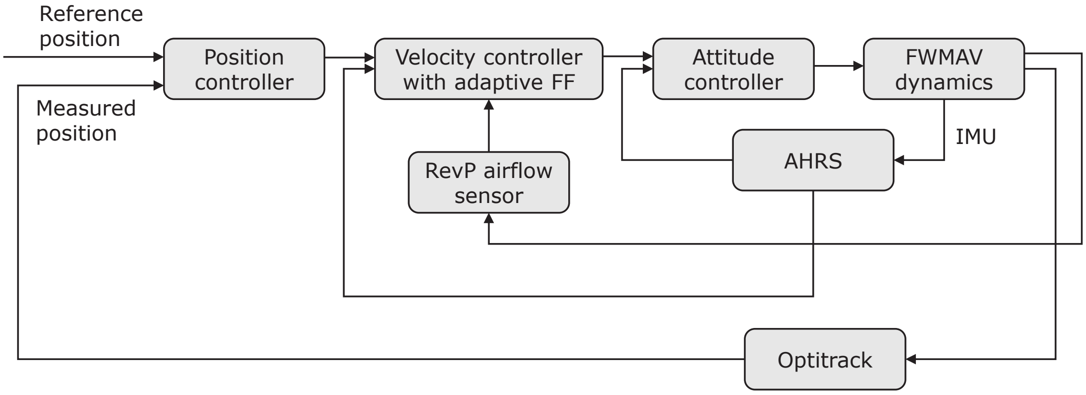


Figure B.1: The diagram of the flight control system with the only adaptive part in the feedforward block of the velocity controller.

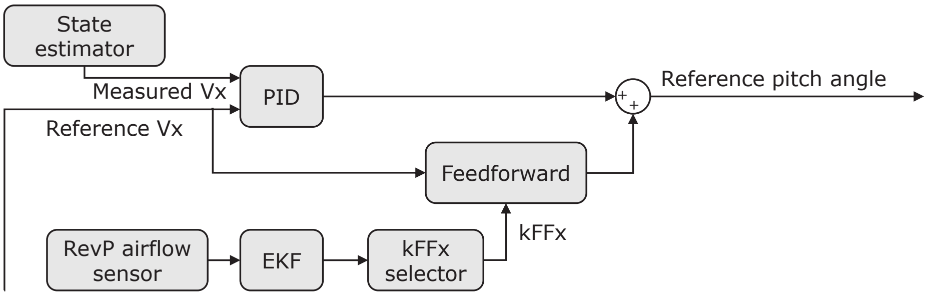


Figure B.2: The diagram of the adaptive feedforward velocity controller.

B.2. K_{FF_x} selector

The values of K_{FF_x} corresponding to different intervals of wind speed and filtered airflow sensor reading has been shown in B.1.

Wind speed [m/s]	Filtered airflow sensor reading [V]	K_{FF_x} [-]	No. of intervals
[0, 0.780)	[0, 1.740)	10.0 (default)	1
[0.780, 1.087)	[1.740, 1.830)	17.0	2
[1.087, 1.449)	[1.830, 1.905)	18.5	3
[1.449, 1.817)	[1.905, 1.960)	19.0	4
[1.817, 2.257)	[1.960, 2.010)	21.5	5
[2.257, ∞)	[2.010, ∞)	23.0	6

Table B.1: The values of K_{FF_x} with the corresponding ranges of wind speeds and filtered airflow sensor readings.

The K_{FF_x} works as Algorithm 1.

Algorithm 1 K_{FF_x} selector

```
while $t = t_i$ do
    initialize $s_i \leftarrow 0$, $S_i \leftarrow 0$ and $len \leftarrow 10$
    gain $s_{i-1}$ and $S_{i-1}$ from the previous time step
    read the filtered airflow sensor reading from EKF and assign it to $V_{air}$
    check which filtered airflow sensor reading interval the value of $V_{air}$ is in
    assign the corresponding number of intervals to $s_i$, as $s_i = 1, 2, ..., 6$
end while
if $s_i = s_{i-1}$ then
    $S_i = S_{i-1} + 1$
else if $s_i = 1$ then
    $S = 0, K_{FF_x} = 10.0$
else
    $S_i = 0$
end if
while $S_i > 0$ do
    $S_i = min(S_i, len)$
end while
if $S_i = len$ then
    update $K_{FF_x}$ with the $K_{FF_x}$ value corresponding to $s_i$ in B.1
else
    keep the value of $K_{FF_x}$ as the previous time step
end if
```

B.3. Performance

This adaptive feedforward velocity controller is validated with several hovering flights under the gust alternating between $0.5m/s$ and $2.4m/s$ at different frequencies. The RMSE values are shown in B.2.

Wind speed	$0.5\ m/s$ and $2.4\ m/s$		
	RMS errors		
f_{gust} [Hz]	X_G [m]	Z_G [m]	θ [deg]
0.25	0.3435	0.0833	8.8880
0.33	0.3184	0.0620	5.5373
0.50	0.1947	0.0313	1.6016
0.75	0.1235	0.0416	3.1116

Table B.2: The RMS errors of the position in X_G and Z_G axis, and pitch angle θ from the in-gust hovering flights with the adaptive feedforward velocity controller under the gust alternating between $0.5m/s$ and $2.4m/s$.