



THE HONG KONG
POLYTECHNIC UNIVERSITY

香港理工大學

Pao Yue-kong Library

包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

**MODELLING AND CONTROLLING OF AN
AUTONOMOUS TAIL-SITTER VERTICAL TAKE-OFF
AND LANDING (VTOL) UNMANNED AERIAL
VEHICLES (UAVs)**

WEIFENG ZHOU

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University
Department of Mechanical Engineering

**Modelling and Controlling of an Autonomous Tail-sitter
Vertical Take-off and Landing (VTOL) Unmanned Aerial
Vehicles (UAVs)**

Weifeng Zhou

A thesis submitted in partial fulfillment of
the requirements for the degree
of
Doctor of Philosophy

Oct 2020

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

_____ (Signed)

Zhou Weifeng (Name)

Abstract

Tail-sitter vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV) combines the advantage of simple mechanism, easy takeoff and landing, high-speed cruises and long flight endurance. These advantages have attracted the attention of researchers, but it is still required to overcome the difficulties of flight control for utilizing this kind of vehicle. In this study, model predictive controllers (MPCs) were proposed to control the position of a quadrotor tail-sitter VTOL UAV. The vehicle was manufactured and its dynamic and kinematic was described in the inertial frame of north-east-down (NED). The propulsion system was modeled by setting up a thrust experiment. The aerodynamic effect of the wing was modelled by applying the momentum theorem and the component breakdown method. Then a successive linearization MPC (SLMPC) controller was designed for hovering control based on a plant model, an estimated disturbance model and an unmeasured disturbance model, followed by the determination of the objective function and the constraints on variables. The SLMPC controller was tested and tuned under a software-in-loop (SIL) condition until a stable and satisfactory performance. Then it was installed onto the vehicle and indoor flight experiments of disturbance rejection were conducted in the Vicon motion capture environment. The result has shown that the proposed SLMPC control method can perform a precise and stable position holding under non-uniform windy conditions compared to the traditional linear MPC (LMPC) controller.

A system identification method is taken to model the vehicle under the cruise stage. Grey box models were derived from the dynamic and kinematic equations and the outdoor flight experiment was designed. An identification section and a validation section were selected from the collected flight data, followed by feeding to a low-pass filter. A least square regression method is taken to fit the grey box model to the data of the identification section and the problem is solved by using a trust-region algorithm. By obtaining and validating the models in the longitudinal and lateral direction, MPC controllers have been set up and tested in the SIL environment. A controller switching mechanism is then developed to complete the large envelope control of the forward and backward transition. During the controller switch, a warm-up mechanism can help to suppress the unwanted control chattering. They have been tested the SIL condition and the improvement in transition performance is presented in the result.

Publications Arising from The Thesis

[1] B. Li, **W. Zhou**, J. Sun, C.Y. Wen, and C.K. Chen, "Model Predictive Control for Path Tracking of a VTOL Tailsitter UAV in an HIL Simulation Environment, " in AIAA Modeling and Simulation Technologies Conference, Kissimmee, Florida, USA, 2018.

[2] B. Li, **W. Zhou**, J. Sun, C.-Y. Wen, and C.-K. Chen, "Development of Model Predictive Controller for a Tail-Sitter VTOL UAV in Hover Flight," *Sensors*, vol. 18, p. 2859, 08/30 2018.

[3] **W. Zhou**, B. Li, J. Sun, C.-Y. Wen, and C.-K. Chen, "Position control of a tail-sitter UAV using successive linearization based model predictive control," *Control Engineering Practice*, vol. 91, p. 104125, Jan. 2019.

[4] B. Li, J. Sun, **W. Zhou**, C. -Y. Wen, K. H. Low and C. -K. Chen, "Transition Optimization for a VTOL Tail-Sitter UAV," in *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2534-2545, Oct. 2020.

[5] **W. Zhou**, S. Chen, C. W. Chang, C. Y. Wen, C. K. Chen, and B. Li, "System Identification and Control for a Tail-sitter Unmanned Aerial Vehicle in the Cruise Flight," *IEEE Access*, 2020.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Chih-yung Wen for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of journal papers. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my supervisor, I would like to thank Prof. Chih-keng Chen and Dr. Lu Peng for widen my research from various perspectives. Their insightful comments and encouragement always do a lot of helps.

My sincere thanks goes to Dr. Boyang Li and Dr. Jingxuan Sun, who led me into this research area and assisted me in overcoming countless of difficulties. Also I thank Mr. Ching-wei Chang, Mr. Shengyang Chen, Mr. Yurong Feng and Miss. Qianyu Zhou, without their precious support I would not be possible to conduct and finish this research.

I wholeheartedly thank my fellow teammates in the GH034 group for all the fun we have had in the last four years. In particular, I am grateful to the staff at the Department of Mechanical Engineering and the Research Office.

Last but not the least, I would like to thank my parents, my sisters, my little nephew and my friends for supporting me spiritually throughout my Ph.D study.

Table of Contents

1. INTRODUCTION	1
1.1 BACKGROUND	1
1.2 OBJECTIVE AND METHODOLOGY	5
1.3 OUTLINE OF THE THESIS	6
2. LITERATURE REVIEW	8
2.1 MODERN CONTROL METHODS.....	8
2.2 MODEL PREDICTIVE CONTROL	9
2.3 SYSTEM IDENTIFICATION.....	13
3. THEORY OF THE MPC ALGORITHM	16
3.1 MPC MODELS	17
3.1.1 <i>Estimated Disturbance Model</i>	17
3.1.2 <i>Prediction Model</i>	18
3.1.3 <i>Unknown Disturbance Model</i>	20
3.2 OPTIMIZATION AND STABILITY PROBLEM	21
3.2.1 <i>Objective Function</i>	21
3.2.2 <i>Constraints</i>	24
3.2.3 <i>Stability</i>	25
4. HOVER FLIGHT CONTROL	27
4.1 SYSTEM CONFIGURATION AND MODELLING.....	27
4.1.1 <i>Dynamic Modelling</i>	28
4.1.2 <i>Propulsion System Modelling</i>	29
4.1.3 <i>Aerodynamic Modelling</i>	31
4.1.4 <i>Allocation model</i>	34
4.1.5 <i>Software-in-Loop Simulation</i>	36
4.2 LINEAR MPC CONTROLLERS.....	37
4.2.1 <i>Plant Model</i>	37
4.2.2 <i>Optimization and Stability Problem</i>	40
4.2.3 <i>Compensation on Thrust and Angle Command</i>	41
4.2.4 <i>Flight Tests and Results</i>	42
4.3 SUCCESSIVE LINEARIZATION MPC CONTROLLERS.....	49

4.3.1 Plant Model.....	49
4.3.2 Discretization.....	52
4.3.3 Allocation Mapping with Control Surface.....	54
4.3.4 Effectiveness of Estimated Disturbance Model.....	56
4.3.5 Effectiveness of Unknown Disturbance Model.....	57
4.3.6 Time-Varying Weights.....	58
4.3.7 Velocity as Reference.....	60
4.3.8 Simulation Results and Comparison.....	62
4.3.9 Flight Tests and Results.....	67
5. CRUISE FLIGHT CONTROL	71
5.1 SYSTEM IDENTIFICATION SETUP.....	71
5.1.1 Grey-box Model of Longitudinal Dynamic.....	72
5.1.2 Grey-box Model of Lateral Dynamic.....	74
5.1.3 Least Square Regression with Regularization.....	74
5.1.4 Trust Region Algorithm.....	76
5.2 EXPERIMENTAL SETUP.....	78
5.3 SYSTEM IDENTIFICATION AND RESULTS.....	81
5.4 LINEAR MPC CONTROLLERS.....	83
6. TRANSITION CONTROL.....	87
6.1 OPERATION RANGE OF MPC CONTROLLERS.....	87
6.2 CONTROLLER SWITCH MECHANISM AND RESULTS.....	89
7. CONCLUSION AND FUTURE WORK.....	93
7.1 CONCLUSION.....	93
7.2 FUTURE WORK.....	94

Nomenclature

A	=	area id propeller disk, m^2
A	=	state matrix
B	=	input matrix
\bar{c}	=	mean aerodynamic chord, m
C	=	output matrix
C_L, C_D, C_M	=	lift, drag and moment coefficient
D	=	drag force, N
d	=	unknown disturbance
e_y	=	output error
e_u	=	input error
F, M	=	force and moment, N and Nm
F_{aero}, M_{aero}	=	aerodynamic force and moment in body frame, N and Nm
F_P, M_P	=	propulsion force and moment in body frame, N and Nm
F_d, M_d	=	disturbance force and moment in body frame, N and Nm
F_g	=	weight in inertia frame, N
f_T	=	thrust of a motor and propeller
g	=	gravity, m/s^2
G	=	weight, N
H	=	control horizon
I	=	inertia matrix, $kg \cdot m^2$
J	=	cost function
l_x, l_y	=	moment arm from a motor to center of gravity along x-axis and y-axis, m
L	=	lift force, N
m	=	mass, kg
n_x	=	number of state
n_y	=	number of output

n_u	=	number of input
n_v	=	number of estimated disturbance
P	=	prediction horizon
\mathbf{p}	=	position in inertia frame, $[x, y, z]$
\mathbf{Q}	=	weight matrix of output
r	=	radius of the flow tube, m
R	=	radius of propeller disk, m
\mathbf{R}	=	rotation matrix
\mathbf{R}_u	=	weight matrix of input
$\mathbf{R}_{\Delta u}$	=	weight matrix of input increment
s	=	distance between the point of interest to the center of the propeller, m
S	=	surface area of the wing, m^2
\mathcal{S}	=	scale factor
T	=	thrust, N
T_s	=	sampling time, s
T_c	=	time constant
\tilde{T}	=	compensated thrust command
\mathbf{T}	=	matrix of angular transformation
\mathbf{u}	=	control input or manipulated input
$\hat{\mathbf{u}}$	=	predicted input
$\bar{\mathbf{u}}$	=	the nominal input
$\Delta \mathbf{u}$	=	input increment
\mathbf{v}	=	velocity in body frame, $[u, v, w]$
v_0	=	trim point air speed, m/s
\mathbf{V}_{induce}^B	=	induce velocity due to propeller slipstream in body frame, m/s
v_{free}	=	freestream velocity, m/s
\mathbf{V}_{wind}^I	=	wind speed in inertia frame, m/s
\mathbf{V}_{gs}^I	=	ground speed in inertia frame, m/s
\mathbf{V}_{as}^B	=	airspeed, m/s
\mathbf{v}	=	estimated disturbance

\mathbf{x}	=	states
$\hat{\mathbf{x}}$	=	predicted states
$\bar{\mathbf{x}}$	=	nominal states
$\overline{\Delta\mathbf{x}}$	=	nominal states increments
\mathbf{x}_d	=	incrementation of unknown disturbance
\mathbf{y}	=	measured outputs
$\bar{\mathbf{y}}$	=	nominal outputs
$\hat{\mathbf{y}}$	=	predicted outputs
\mathbf{z}	=	Quadratic Programming (QP) decision
α	=	angle of attack, degree
β	=	sideslip angle, degree
γ	=	flight patch angle, degree
δ_T	=	change of throttle
δ_e	=	change of elevator
δ_a	=	change of aileron
Δ	=	deflection of control surfaces
η	=	throttle of motor
ϕ, θ, ψ	=	Euler angle of roll, pitch and yaw, degree
$\tilde{\phi}, \tilde{\theta}$	=	compensated roll and pitch command
λ	=	ridge regression coefficient
τ	=	moment, Nm
ω	=	rotation speed of propeller, rpm
$\boldsymbol{\omega}$	=	angular velocity in body frame, $[p \ q \ r]$
$\boldsymbol{\Omega}$	=	time derivative of Euler angle in inertia frame, $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]$

Abbreviations

AOA:	Angle of Attack
CTOL:	Conventional Takeoff and Landing
DAQ:	Data acquisition
EOM:	Equations of Motion
FCU:	Flight control unit
FMU:	Flight management unit
GCS:	Ground Control System
HIL:	Hardware-in-Loop
LTI:	Linear Time Invariant
IMU:	Inertia measurement unit
LP:	Linear programming
MAV:	Micro Aerial Vehicle
MAC:	Mean aerodynamic chord
MIMO:	Multi-Input-Multi-Output
MPC:	Model Predictive Control
MV:	Manipulated variable
MSE:	Mean square error
NED:	North-Earth-Down
OV:	Output variable
PID:	Proportion Integration Differentiation
PPM:	Pulse position modulation
PWA:	Piecewise Affine
PWM:	Pulse-width modulation
QP:	Quadratic Programming
RHC:	Receding horizon control
ROS:	Robot operating system

RMSE:	Root mean square error
SIL:	Software-in-Loop
SLMPC:	Successive Linearization Model Predictive Control
VTOL:	Vertical Takeoff and Landing
UAV:	Unmanned Aerial Vehicle
UDP:	User datagram protocol

List of Figures

Figure 1-1 Hiller X-18 Tilt wing cargo transport [10]	2
Figure 1-2 Russian Tiltrotor UAV RHV-35 [11]	2
Figure 1-3 MLB V-Bat [12]	2
Figure 1-4 CW-30 from JOUAV [13]	3
Figure 1-5 WingtraOne mapping drone [14]	3
Figure 1-6 Alfa Pixhawk 2 Tailsitter VTOL [15].....	3
Figure 1-7 Bell APT 70 [16].....	4
Figure 2-1 MPC Analogy [40].....	10
Figure 3-1 Receding Horizon Idea [73].....	16
Figure 3-2 An Illustration of an MPC controller	17
Figure 4-1 The testbed tail-sitter: PolyU Plus	27
Figure 4-2 Schematic of coordinate systems	28
Figure 4-3 Experimental setups of propulsion system modelling	30
Figure 4-4 The experiment data and fitted curve of the motor-propeller system.....	30
Figure 4-5 The schematic diagram of aerodynamic modelling.....	31
Figure 4-6 The component breakdown method.....	33
Figure 4-7 The relationship between thrust and moment to the rotation speed of a propeller.....	35
Figure 4-8 The fitted relationship between the throttle to the rotation speed of a motor.	35
Figure 4-9 Schematic diagram of the simulation environment	36
Figure 4-10 Block diagram of Two-loop Linear MPC controller	37
Figure 4-11 Simulation result of LMPC controller following a circular path with (blue) and without (red) thrust compensation	42
Figure 4-12 Communication network setup	43
Figure 4-13 Experimental setup.....	44
Figure 4-14 Experiment result of LMPC position controller following a circular path.....	45
Figure 4-15 Wind speed measurement	46
Figure 4-16 Measurement of unsteady wind speed of the artificial wind field.....	46
Figure 4-17 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using a PID controller.	47
Figure 4-18 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using a linear MPC controller.	48

Figure 4-19 Block diagram of the cascaded control structure of the SLMPC controller used in the PolyU Plus tail-sitter vehicle.	49
Figure 4-20 Simulation result of SLMPC Hover with (blue) and without (red) control surface .54	
Figure 4-21 Simulation result of step move and position hold by SLMPC under 2m/s wind with control surface.....	55
Figure 4-22 Simulation result of failure to hold the position of SLMPC under 2m/s wind without control surfaces	55
Figure 4-23 Simulation result of SLMPC controller following a step signal in the z-direction with (blue) and without (red) disturbance model.....	57
Figure 4-24 Simulation result of the effectiveness of the elimination of unknown disturbance under 2m/s wind condition.....	58
Figure 4-25 Simulation results of step-command following with fixed weights of $Q = 2.0$. and $Q = 5.5$ and a varying weight.....	59
Figure 4-26 Simulation results of the step command following with (red) and without (blue) velocity as reference.	61
Figure 4-27 Position of a trajectory following simulation of a circular path.	63
Figure 4-28 The corresponding error of Figure 4-27.....	63
Figure 4-29 The Euler angles of trajectory following simulation of a circular path.	64
Figure 4-30 Simulation of position control with disturbance rejection when the wind of 1m/s, 2m/s and 3m/s magnitudes coming from 0° to north (negative XI -direction in Fig. 1).	65
Figure 4-31 Root mean square errors of controllers with disturbance rejection when the wind speed ranges from 0.5 m/s to 3.3 m/s and comes from 0° to north (negative XI -direction in Fig. 1).....	66
Figure 4-32 Root mean square errors of controllers with disturbance rejection when the wind of 3 m/s coming from 0° to 180°	66
Figure 4-33 Experimental results of SLMPC position controller following a circular path	67
Figure 4-34 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using the SLMPC controller	68
Figure 4-35 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using the SLMPC controller for 4m/s wind resistance.....	69
Figure 5-1 (a) The PolyU Plus tail-sitter UAV in cruise. (b) the coordinate system of the body and wind axis.	72

Figure 5-2 (a) Schematic diagram of the signal generator module through the training line of the remote control transmitter. (b) The setup of the signal generator and the remote control transmitter.78

Figure 5-3 (a) The flight log data of the remote control input signal of a doublet waveform. (b) The flight log data of frequency sweep of a sin wave.79

Figure 5-4 (a) The flight sections for the system identification training set (red) and validation set (black) within the entire flight path (blue) in the 3D view. (b) The flight sections in three separate axes view.....81

Figure 5-5 (a) The longitudinal flight data (blue) and the reconstructed data (red) for the learning data set. (b) The validation data set according to the grey-box longitudinal model.....82

Figure 5-6 (a) The lateral flight data (blue) and the reconstructed data (red) for the learning data set. (b) The validation data set according to the grey-box lateral model.....83

Figure 5-7 A schematic diagram of application of the model predictive control to longitudinal and lateral attitude controllers.84

Figure 5-8 Flight simulation results of the tail-sitter UAV controlled by the longitudinal and lateral MPC controllers, which track altitude, airspeed, and roll angle.....85

Figure 6-1 Block diagram of the control mechanism. The left (red) part handles the hover stage and the right (blue) handles the cruise period.....87

Figure 6-2 Simulation results of controlling the UAV in large pitch down conditions of -50° (red), -60° (blue) and -65° (green).....88

Figure 6-3 Simulation results of cruise LMPC control start from the initial condition of -60° pitch and 5 m/s forward speed.89

Figure 6-4 Simulation results of the whole flight process including takeoff, forward transition, cruise, backward transition and landing.....90

Figure 6-5 Detail illustration of the simulation result of pitch angle movement with and without controller warmup.....90

Figure 6-6 Normalized control command to actuators and control output from controllers during transition.92

1. Introduction

1.1 Background

Unmanned Aerial Vehicles (UAVs) are popular in the consumer market and they are catching the more and more attention from commercial markets from different industrial sectors, such as electricity, oil and gas, bridge and tunnel, and construction. The application of autonomous unmanned UAVs has the potential to significantly improve the efficiency for many industries and services. UAVs are usually cheaper in price, smaller in size, lighter in weight and able to operate in dangerous area. Different configurations of UAVs were developed to fit their designed tasks. Conventional take-off and landing (CTOL) aircraft have a good capability of carrying heavy cargo, long-distance flight and high cruise speed, whereas the runway is required. Vertical Take-Off and Landing (VTOL) aircraft can take-off, hover and land vertically. A tail-sitter VTOL vehicle combines the advantages of easy take-off and landing as well as long flight distance and high cruising speed. However, this design does not fit the manned aircraft because it is difficult for a pilot to get in and out of the cockpit while the aircraft is vertically sitting.

Among VTOL aircraft, the tilt-wing, tilt-rotor and tail sitter designs of airframe configurations can achieve forward flight and vertical flight concluded in [6]. The Tilt-rotor design, which the wings are fixed but the mounted engines can rotate. It combines the vertical lift capability of a helicopter with the speed and range of a conventional fixed-wing aircraft. The tiltrotors have the advantage of a significantly higher speed than a helicopter. For example, the Bell-Boeing V-22 Osprey [7] which was first flying in 1989 and introduced in 2007. A drawback of the tilt-rotor aircraft is the complexity in the mechanical system which could seriously decrease the reliability. Figure 1-2 shows a Russian prototype of a Quad tilt-rotor configuration UAV, containing both the fixed wings and rotating rotors. It is complicate in mechanics but able to go through the transition easily.

The Tilt-wing design VTOL was achieved by mounting the engines on the wings and rotating with the wings according to needs. The slipstream from the rotor strikes the wing on its smallest dimension in vertical flight, the tilt-wing can apply more of its engine power to lifting the aircraft compared to tilt-rotor design. The main drawback is the difficulty of control during hover since the wing tilted vertically represents a larger surface area that can be affected by crosswinds [8]. Figure 1-1 shows the Hiller X-18 Tilt wing cargo transport, which was designed for in-house study during early 1955 and gained interest from the U.S. Air Force for cargo transport. The prototype was completed in 1958 and after a total of 20 flights, the X-18 was disassembled for scrap in 1964.

The concept of a tail-sitting aircraft was introduced by Nikola Tesla in 1928 [9]. Figure 1-3 shows a ducted-fan tail-sitter UAV. The tail of this configuration serves as the landing gear in the vertical motion as well as a rudder in forward flight. The tail-sitters are much less complicated in mechanisms, which reduces operating weight and risk of mechanical failure. However, the transition phase raises high demands on control system design due to the nonlinear dynamics of the aircraft in large angles of attack region.

Apart from the three aforementioned configurations, the Quad-plane type also gained popularity from the commercial market nowadays. This configuration is a summation of a quadrotor vehicle and a conventional aircraft. Figure 1-4 is a CW-30 is a twin-boom V-tail hybrid VTOL fixed-wing UAV platform, specifically designed for small/medium-areas flight missions. During vertical take-off and landing, the four electric motors will be used. In the cruise phase, the gasoline engine working in conjunction with a pusher type propeller will provide the thrust. By using two sets of propulsion systems, the quad-plane configuration achieves overall long endurance, high speed, large payload, high-efficiency, stable structure and high reliability, allowing missions for a large area.



Figure 1-1 Hiller X-18 Tilt wing cargo transport [10]



Figure 1-2 Russian Tiltrotor UAV RHV-35 [11]



Figure 1-3 MLB V-Bat [12]



Figure 1-4 CW-30 from JOUAV [13]

The tail-sitter VTOL vehicle combines the advantages of a quadrotor and a fix-wing aircraft, which can conduct missions without the need for a runway or a catapult while reserving the ability of long flight distance and high cruise speed. With these characteristics, tail-sitter is one of the most suitable configurations for civil applications in a high-density urban city like Hong Kong, which also has many offshore islands with inhabitants. These characteristics make tail-sitter vehicles attractive.

The novel configuration of tail-sitter UAVs with unique practical functionalities has drawn the commercial parties attention. Such as the Wingtra-One® aircraft made in Zurich Switzerland or the Alfa Pixhawk 2 Tailsitter VTOL Mapping Drone® from Shenzhen China.



Figure 1-5 WingtraOne mapping drone [14]



Figure 1-6 Alfa Pixhawk 2 Tailsitter VTOL [15]



Figure 1-7 Bell APT 70 [16]

Meanwhile, many researchers are interested in this platform. Such as Ren Suzuki et al [17] built a teleoperation tail-sitter UAV that was designed to hover indoor or very close to places that are complicated and dangerous. With the help of an onboard microcomputer, GPS, Attitude Heading Reference System (AHRS) and a set of cameras and ultrasonic sensors, the vehicle can perform indoor exploration as well as target tracking. It can also move along toward a target in an indoor environment while automatically avoid obstacles. R. Naldi et al [18, 19] designed a ducted-fan tail-sitter with several rigidly attached under-actuated modules. The modular system strongly improves the number of degrees of freedom than the standard ducted-fan configuration. With the right number and orientation of modules, it can achieve six degrees of freedom control on hovering. Parany Sinha et al. [20] designed a modular extensible tail-sitter aerial platform, which can add or subtract rotors with tail on the platform wings. Then forming different shapes of the platform to achieve different missions. Equipment and sensors can also be added as modules. This vehicle used the Paparazzi open-source software [21] and its Lisa avionics system was embedded. The experimental result has shown that this vehicle can perform all stages of hovering, transition and forward flight. Xinhua Wang et al [22] designed an agile tail-sitter aircraft with four rotors on the tail controlling motions and a main co-axial counter-rotating propeller providing thrust. The result indicated that this vehicle has achieved better agile maneuverability than conventional tail-sitter aircraft in both forward and backward transition even though it is more complex in mechanics than others.

The aforementioned literature has shown that the control stability of a tail-sitter aircraft over a large flight envelope is critical and challenging. Compare to the other flight stages, a stable and robust hovering control will be the priority since the hover flight is the most basic component of the whole flight. What's more, the tail-sitter vehicles are easily affected by the wind during hovering due to their

significant wing area, which causes extra challenges in control system development. It is concluded that a control system that integrates both vehicle models and environment models is needed.

1.2 Objective and Methodology

This project is focused on developing a control system based on model predictive control algorithm that is capable of controlling a tail-sitter VTOL UAV throughout the hover, transition and cruise flight stages.

Traditional PID controller, one of the most widely adopted control methods in preictal, is not designed with models and its performance relies on parameter tuning. The model predictive control (MPC) method is a model-based multi-input-multi-output (MIMO) control method using a receding horizon strategy. The disturbance on a tail-sitter vehicle in hover and cruise is highly related to its aerodynamic forces and moments, which should be estimated continuously and compensated. Under this situation, the control of a tail-sitter is proposed to be developed using the MPC method in this study. The aerodynamic model and environment model are implemented in the control system, and thus the disturbance forces can be estimated while optimizing the control commands. The MPC method has the advantages of imposing constraints and weights in the optimization process. The constraints can limit the control variables according to physical system characteristics while weightings can provide different priorities on tracking variables.

In this study, a VTOL tail-sitter UAV will be built for control system implantation, system identification and performance evaluation. The model of the vehicle in hover stage will be discussed, including the dynamic model, the propulsion system model and the aerodynamic model, followed by the development of the simulation environment. The nonlinear thrust and moment effect of the propulsion system is modeled by experimenting and summarized into the relationship of fourth-order functions. An aerodynamic database will be searched for lift coefficient, drag coefficient as well as moment coefficient to ensure the accuracy of aerodynamic prediction. Discrete linear time-invariant (LTI) state-space models of the vehicle will then be set up and being implemented to the MPC controller design. The vehicle model is further linearized based on the small-disturbance theory and integrated into the control system. The MPC control algorithm based on the linear UAV dynamic model will be developed and software-in-loop (SIL) simulation will be conducted to examine the developed control system.

After the controllers were set up, an iterative simulation test will be conducted to obtain a set of parameters that ensure the vehicle's stability as well as the performance. When the simulation shows

reliable and robust results, flight tests of disturbance rejection and trajectory tracking on the vehicle will be carried out. Further, the successive linearization MPC (SLMPC) method is applied and developed for the tail-sitter vehicle hover control so that the prediction model can be adjusted at run time to compensate for the nonlinear aerodynamic characteristics. By updating the state-space model of the controller at every time instant, the nominal states can be well controlled and followed.

Rather than using theoretical derivation, the model of the vehicle in cruise stage will be obtained by system identification based on real flight data. The MPC control will control the vehicle in simulation by using the identified model. A controller switching mechanism will be designed to switch between the hover MPC and the cruise MPC for transition purpose.

1.3 Outline of the Thesis

Tail-sitter vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV) combines the advantage of a simple mechanism, easy takeoff and landing, high-speed cruises, and long flight endurance. These advantages have attracted the attention of researchers, but it is still difficulties in flight control for utilizing this kind of vehicle. In this study, model predictive controllers (MPC) were proposed to control a quadrotor tail-sitter VTOL UAV under both hover and cruise conditions.

The introduction is given in chapter 1. The background of the UAV system and the necessity for VTOL capability are introduced. The different configurations of tail-sitter UAVs from the literature are reviewed and compared. The chapter ends with the research objectives and an outline.

In chapter 2, the thesis studied the conceptual design and the overall layout of the tail-sitter UAV. The performance requirements of the system are determined based on the analysis of a typical cargo delivery mission in Hong Kong. The history, development, and evolvement of the model predictive control method are studied. Followed by the revision of the modelling method of system identification.

In chapter 3, the theory of the model predictive controller is described in detail. The basic state-space model is augmented with estimated disturbance terms and unknown disturbance terms. The prediction and the recursive process is analyzed. Followed by the development of objective function and constraints.

The UAV platform was designed and manufactured in chapter 4. The design is based on the principles of optimal aerodynamic efficiency and control reliability. The aerodynamic and dynamic modelling for the UAV was built up. The momentum theory was used to model the propeller and the slipstream. The MPC controllers for hover are introduced. With the model and the control theory, the controller was then built and tested in software-in-loop, hardware-in-loop, and indoor flight as well.

An advanced successive-linearization MPC (SLMPC) was then developed and tested. Its performance was compared to a PID and linear MPC controller and show a significant improvement.

Chapter 5 demonstrated the system identification process to obtain the model of the vehicle during the cruise stage and control it. Grey box models were first derived from linearizing the longitudinal and lateral dynamics of the aircraft. Specially designed control inputs were sent to the flying aircraft through the training line during an outdoor experiment. The grey box models were fitted to the collected data by the least square regression method and a trust region algorithm was used to solve the optimization problem. The result was validated by putting the model into a linear MPC controller and achieve a precise cruise control.

Chapter 6 describes the transition mechanism for both forward-transition and back-transition. The control of the transition period the major challenge due to its high-nonlinear aerodynamic characteristics in high angle-of-attack of the vehicle. The traditional method usually results in large altitude changes, which makes the transition unpredictable and inefficient. The proposed transition mechanism takes the advantages of the SLMPC of a very large control envelope to achieve closed-loop control during the whole transition period.

The thesis is concluded in chapter 7, with the final performance of the current UAV system. The future work is proposed for further improvements for the system towards large-scale applications.

2. Literature Review

In this chapter, the development of modern control methods is reviewed and introduced in the first place. Followed by introducing and explaining the concept of model predictive control (MPC). Some other control methods that evolved from MPC are also presented as well as their application in research. In the meantime, the research studies that have a disturbance model augmented to MPC control are also reviewed, since it can make full use of the advantages of MPC and significantly improve the performance. The setup of the MPC control requires a model of the system and as a result, a modelling method is necessary. In the hovering stage, the model is set up based on a quadrotor UAV. The effect course by the wing was treated as disturbance and corresponding modification has been added to the model. In the cruise stage, a system identification method is taken to obtain a model, such that the MPC control on this stage can be facilitated. As a result, the last section in this chapter is a revision of the system identification method.

2.1 Modern Control Methods

Key milestones of the development process of some existing control methods are worthwhile to be mentioned. The classical Control method involves iterative single-loop design that was costly in time and manpower. Systems were often designed by discretizing the flight envelope at specific operating points, developing the control law at these points, and guaranteeing the robustness. This one-loop-at-a-time design approach was aided by tools like Root locus analysis, Bode plots or Nyquist plots, etc., that enabled us to visualize how the system dynamics were being modified. However, the design procedure became increasingly difficult as more loops were added and did not guarantee stability when the dynamics involve multi-variables [23].

In the 1970s and 1980s, the question of robust performance was raised, and new control system design and analysis methods emerged [24]. Modern control aims to deal with multi-input-multi-output (MIMO) systems, with much better stability and robustness. In the modern method, control gains can be computed simultaneously so that all loops are closed at the same time, which means it can design control systems in a much quickly and directly manner. Modern control has made a significant impact on the aircraft industry after then. Bryson [25, 26] pioneered in applying it to aircraft control. Boeing has implemented control systems designed using modern techniques, for instance, in the Boeing 767 autopilot [27]. Honeywell has promoted modern robust design [28]. Linear quadratic methods were used by General Dynamics in the control system of the F-16 fighter [29].

Adaptive Control research was initiated in the early 1950s. At that time, the interest in this group of systems was primarily driven by the design of autopilots for high-performance aircraft. For over 50 years, adaptive systems have remained in the mainstream of controls and dynamics research. Because adaptive control has its clearly defined goal to enable control of dynamical systems to operate in the presence of unknown parameters. The original concept of a Model Reference Adaptive System (MRAC) was proposed by Whitaker et al. in [30, 31]. Soon after its introduction, the first proof of MRAC closed-loop stability using Lyapunov theory was given in 1965 by Butchart and Shackcloth [32] and also in 1966 by Parks [33]. In the years that followed, adaptive control theory for a broad class of dynamical uncertain systems was developed and well documented in several now-classical textbooks [34-37].

There are some fundamental problems while designing a control system, like pole placement, regulator problem, tracker problem, model following problem and robustness problem, etc. Many higher-level control methods were developed to better solve these problems and to adapt to different kinds of objectives or specific tasks. This was the start of the theory of Robust Control, which took shape in the 1980s and 1990s and is still active today since the traditional method was sometimes found to lack robustness [38], prompting research to improve them. Robust control can be thought of as an online policy that capable of regulating systems whose dynamics may contain uncertainties. The popularity of this technique is primarily based on its guaranteed properties, such as closed-loop stability and robustness to parametric uncertainties [24]. Robust control is designed to operate under the worst-case condition assumption, thus it may use excessive actions to regulate the process.

2.2 Model Predictive Control

Optimal Control emerged in the 1980s [24]. It deals with the problem of finding a control law for a given system, such that a certain optimality criterion can be achieved by minimizing the cost function. For the propose of optimizing future control actions, Model Predictive Control (MPC) model predicts future plant outputs, based on past and current values. The main advantage of MPC is that it allows the current timeslot to be optimized while keeping future timeslots into account. This is achieved by optimizing a finite time horizon, but only implementing the current timeslot [39].

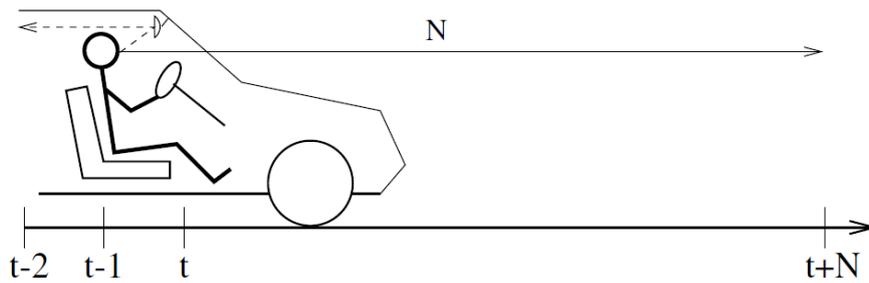


Figure 2-1 MPC Analogy [40]

Notice that the MPC strategy is very like the control strategy used in driving a car illustrated in Figure 2-1. The driver knows the desired reference trajectory for a finite control horizon and by considering the car characteristics decide which control actions (accelerator, brakes, steering) to take to follow the desired trajectory. Only the first control actions are taken at each instant, and the procedure is repeated for the next control decision in a receding horizon fashion. When using classical control schemes, such as PIDs, the control actions are taken based on past errors. The PID way of driving a car would be equivalent to driving the car just using the mirror since PID and LQR controllers do not have this predictive ability, as shown in Figure 2-1. MPC is a relatively mature technique for a linear and a rather slow system, like a process in an industry, chemical plants, etc. However, it is getting higher and higher attention in the research area of application in nonlinear processes and processes with frequent changes in operating conditions.

Unlike linear MPC which has to run in real-time, explicit MPC is one of a variation, which uses offline computations to determine regions and the corresponding control law constant [41]. Stability and robustness are confirmed region by region, which follows the predicted trajectory and the size of the region is controlled by some ‘soften’ constraints. Explicit MPC is capable to deal with the nonlinear process in short responding time and it is an easy-to-implement control law. Multiple MPC scheme allows the switching between different MPC controllers according to a predefined criterion such that each controller can optimally function at its operation range and resulting in a larger control envelop. A similar effect can also be achieved by an adaptive MPC which can update or successively linearizing its state-space model at every time instant, shifting the nominal states according to the real state. Thus, the control envelop is enlarged and a degree of nonlinearity is adapted. The nonlinear MPC scheme does not require a linear model which makes its usage much wider than others. But it brings difficulties in solving the continuous objective function, which is integration rather than a summation.

MPC has gained the attention from researchers in the UAV field and they have developed some MPC strategies and applied them to different needs, like trajectory tracking of a unique UAV

configuration, like a tail-rotor tri-copter UAV. K. Alexis et al [42] proposed a novel control strategy based on Piecewise Affine (PWA) dynamic modelling approach cooperated with a Switching Model Predictive Control (SMPC) [43] method, which successfully controlled a quadrotor UAV over a large flight envelope, such as position hold hovering, aggressive attitude control or trajectory tracking in a wind-gust condition. Yuyi Liu et al [44] developed an onboard trajectory planner based on the MPC method, which locally optimally describes the waypoint for the MAV. Meanwhile, the MAV was handled by a nonlinear backstepping approach of its position and attitude control. Moses Bangura et al [45] present a hierarchical control paradigm that MPC control strategy was used in high-level onboard trajectory tracking. At its mid-level attitude control, a Lyapunov based high-gain was implemented. PID was used at the low-level motor control. A new linear model based on the system was developed by the method of dynamic reduction and dynamic extension.

Daniel Mellinger et al [46] address a quadrotor that was able to maneuver in an indoor environment. The method of differential flatness was applied to the dynamic modelling process. As the state and input can be represented by flat outputs and their derivatives. Fewer variables were needed. At the same time, the trajectory generation was designed based on the minimum snap method, in which the optimal solution can be obtained by solving a quadric programming (QP) problem.

Cunjia Liu et al [47] designed an explicit MPC controller for tracking problems which was defined by the Bezier curve. With the help of differential flatness, the optimization problem was parameterized and converted into standard multi-parametric quadric programming (mp-QP) problem, which can be solved in off-line condition and obtaining the piecewise affine (PWA) functions. Meanwhile, the reduction of the number of variables and constraints successfully reduced the complexity of the explicit solution.

Anna Prach et al [48] has used the MPC algorithm as an attitude control of a tri-copter with tilt-rotor design. This controller handles not only the attitude but also the angular rate and the vertical acceleration of position. Thus, the control input in the inner loop secures the thrust and tilt-angle of the rear motor. This paper shows the possibility of implementing MPC onto a much faster inner loop of a not common model of vehicle, which illustrated the adaptivity and plasticity of the MPC scheme.

M. Abdolhosseini et al [49] used an efficient MPC scheme which usage a model reduction technique to downsize the state-space model and achieving lower computation requirement. Firstly, it applied the world frame instead of the body frame to describe the kinematic of the vehicle which made all terms were strongly related to the thrust. Secondly it usage a strategy to give the thrust command the highest priority to keep the high than the control of others. Also, he took $\sin\phi$ and $\sin\theta$ as control

input instead of just ϕ and θ , which means the dynamic model did not need to be linearized as much as a purely linear model. Although this paper did not give very good results, it still illuminated my path to deal with nonlinearity in the dynamic model.

Mina Kamel et al [50] compared the performance of linear MPC to a non-linear MPC while using a micro aerial vehicle (MAV) on hovering, step response and aggressive trajectory tracking under external disturbances. In the linear MPC controller, not only a feed-forward thrust compensator was designed to improve the vertical performance but also the desired trajectory acceleration was calculated along with the trajectory itself and fed into the output reference while given with weight. Two first-order transfer functions of roll and pitch angle were used to link them into the position control. The final roll and pitch command was again compensated by multiplying the rotation matrix around the z-axis such that the yaw effect was back into consideration.

Some researchers are focusing on improving the performance of the controller by modelling the disturbance and considering this effect onto the controller design. Kostas Alexis et al [51] has developed a linear Robust Explicit MPC controller as a position control for a tail-rotor tri-rotor UAV. It was aimed to ensure minimum deviation even in the worst-case disturbance while satisfying the physical constraints. With strictly defined boundaries of variables and the foreseeing by the feedback prediction term, robustness and minimum peak performance were improved. With all the defined boundaries, a step forward to explicit MPC was taken and the piecewise affine function was calculated offline to allow faster real-time execution. A set of experimental results of tracking a collision-free path in windy conditions have been shown at the end.

Wei Dong et al [52] developed a disturbance observer to achieve high performance in trajectory following for a quad-rotor UAV. The model for the rotor input delay was considered and identified by a test bench. The external disturbance was added into the dynamic model of the vehicle as well as the model mismatch effect term. The disturbance observer acted as a compensator and shows improvement in wind rejection, payload difference and rotor failure condition.

Futao Zhao et al [53] has designed a simple disturbance predictor with the prediction of self-regulating disturbance (first-order transfer signal) or non-self-regulating disturbance (ramp signal). A simplified prediction control scheme which has much lower computational effort due to only one control move being generated every step rather than taking a receding horizon approach. The simple disturbance predictor can have its model parameters updated online and being able to solve the steady-state offset problem which most model-based control schemes will face.

A variety of strategies have been developed to improve the MPC control. A control strategy using a piecewise affine (PWA) dynamic modelling approach with a switching MPC method was proposed in [54]. It was successfully used to control a quadrotor UAV over a large flight envelope, which included aggressive attitudes and trajectory tracking in gusting wind. An efficient MPC scheme was developed in [49]. It uses a model reduction technique to downsize the state-space model to lower the computational requirement. An adaptive MPC scheme was used to control a traditional quadrotor vehicle in [55] and a single degree of freedom flapping-wing UAV in [56]. The linear MPC method has successfully controlled the quadrotor UAV in [45, 57], it does not necessarily mean that it can control a tail-sitter vehicle, which has the strong nonlinear aerodynamic effect of the wings.

Disturbances can be considered and added to the MPC controller to achieve better performance. Meanwhile, disturbance modelling and rejection ability are important to a tail-sitter vehicle control as it usually faces prevailing wind or gust wind conditions. A linear robust explicit MPC controller was developed in [51] as a position control for a tri-rotor UAV to ensure minimum deviation even for the worst-case disturbance. With strictly defined boundaries for variables and the use of a prediction term, robustness and minimum peak performance were improved. A disturbance observer was designed in [58] to conduct an agile trajectory following for a quadrotor UAV. The model for the rotor input delay was included while the external disturbance and model mismatch terms were added to the dynamic model. In [59], a nonlinear disturbance observer estimates the external force/torque from the wind turbulences and it is cooperating with a nonlinear MPC controller for small-scale helicopters. An active disturbance rejection control is used in [60] coupled with an extended state observer. The external and internal disturbance caused by unmodelled dynamic and parameter uncertainties is lumped as an external state and compensated in the control input. These technics can be applied to the controller development of a tail-sitter vehicle as its wingspan will bring predictable and non-predictable disturbance to the system.

2.3 System Identification

Compared with the general flight controller based on proportional-integral-derivative (PID), a model-based control method can bring a number of advantages to UAV control. A model-based controller with a high-fidelity model can usually outperform a gain-scheduled PID controller in most areas, and it requires lower tuning skills [61]. Wind tunnel experiments and system identification are the two major methods for modeling a UAV system. Wind tunnel experiments can find the parameters that describe the aerodynamic forces and moments in a given flow condition [62]. This method, while

widely used, has limitations when applied to a UAV. For example, it is relatively costly and time-consuming; additionally, the steady and one-direction flow inside the wind tunnel is far too optimal compared to the outdoor environment [63].

System identification can be performed on the specific input and output of flight data to estimate the mathematical model or the parameters within if a physical prototype is available. This technique comprises model structure determination, experimental design, data postprocessing, parameter estimation, and model validation. Compared with wind tunnel experiments, system identification is cheap and timesaving because most of the required data, such as translational and rotational acceleration, attitude, velocity, and airspeed, can be measured by standard UAV avionics and fed to the system identification after data processing. Theoretically, this technique can model any motion that the UAV has experienced during flight as long as the flight mode has been excited. Meanwhile, compared to wind tunnel tests, the results from this technique have a closer resemblance to real flight. Nevertheless, unlike the controlled environment in the wind tunnel, there exists wind disturbance in the outdoor environment, and its effectiveness deteriorates for a slow-flying UAV. However, the disturbance is mostly random and low frequency. Only the low-frequency mode of the dynamic will be influenced, and the system identification process has to adapt to this circumstance. Specifically, designed excitations are needed to stimulate the higher frequency modes. In the experimental design, according to the model structure determination and parameter estimation, it is important to apply appropriate input waveforms to the maneuver.

Apart from the measured input and output signals from the system in the time or frequency domain, system identification requires a model structure. The black-box method gives an intuitive model only and lets the system identification process fill the inner content to achieve the best fit without knowing any background dynamic of the identified system. Grey box modeling is a system identification technique that estimates unknown parameters by assuming an intuitive form of the model in advance [64]. The development of a physical-based model requires intimate knowledge of the natural laws of dynamics and kinematics of the UAV undergoing free or force motion [65]. A grey box model is developed according to the physics and dynamics of the system, and as a result, some of the parameters can be predetermined. The values should be physically reasonable with acceptable accuracy. For this reason, the identified parameters are typically used to reconstruct and compare with a new set of flight data that is not used in the identification process. If the validation result shows a good overlap with the new set of flight data, the parameter values can be taken. Otherwise, the estimation process is repeated with a different estimation technique. It is common to use a different shape of the input waveform to reperform this process [66-68].

Researchers have applied system identification for UAV modeling in many studies. Dorobantu et al. [69] described a practical system identification procedure for small, low-cost, fixed-wing unmanned aircraft. They used a linear model obtained from the generic nonlinear equation of motion and adopted it as a baseline model for flight experimental design. The parameters of the linear model were identified by fitting the model to the data. The baseline model was compared with the identified model to demonstrate the effectiveness of system identification. Chumalee and Whidborne applied the racetrack maneuver in one flight to obtain a model with sufficient accuracy [70]. In this work, several unknown constant parameters were augmented into the model, and the hardware-in-the-loop simulation was undertaken before the flight. Scheper et al. presented a grey-box model coupled with the vehicle dynamics in both the longitudinal and lateral directions and applied it to system identification [71]. The result showed that the coupled model with identified parameters can reconstruct a response with a good fit to the aircraft's response. Gandhi et al. demonstrated a linear regression as a baseline model and a Gaussian process as a black-box model to represent the dynamic of the UAV [72]. This work focused on utilizing system identification to learn a black-box model, and the result demonstrated that the Gaussian Process model can fit the experimental data with high accuracy. The above studies have conducted system identification on UAVs and demonstrated their results by comparing the reconstructed data to real flight data. Most of them have shown good accuracy. However, none of them have directly put the identified model into a model-based control for validation.

3. Theory of the MPC Algorithm

MPC control can be considered as a receding horizon control (RHC) where the finite-time optimal control law is computed by solving the optimization problems repeatedly on-line. At present, there is no other technique to design controllers for a general large linear multivariable system with input and output constraints while with a stability guarantee [73]. The idea of a receding horizon strategy is shown in Figure 3-1. At each sampling time, starting at the current state, an optimal control problem is solved over a finite prediction horizon (P). A series of the optimal manipulated input signal (\mathbf{u}) will be computed over the control horizon (H) and only the first u is applied to the process for the sampling interval of $[t, t + 1]$. At the next time step $t + 1$, the horizon shifted.

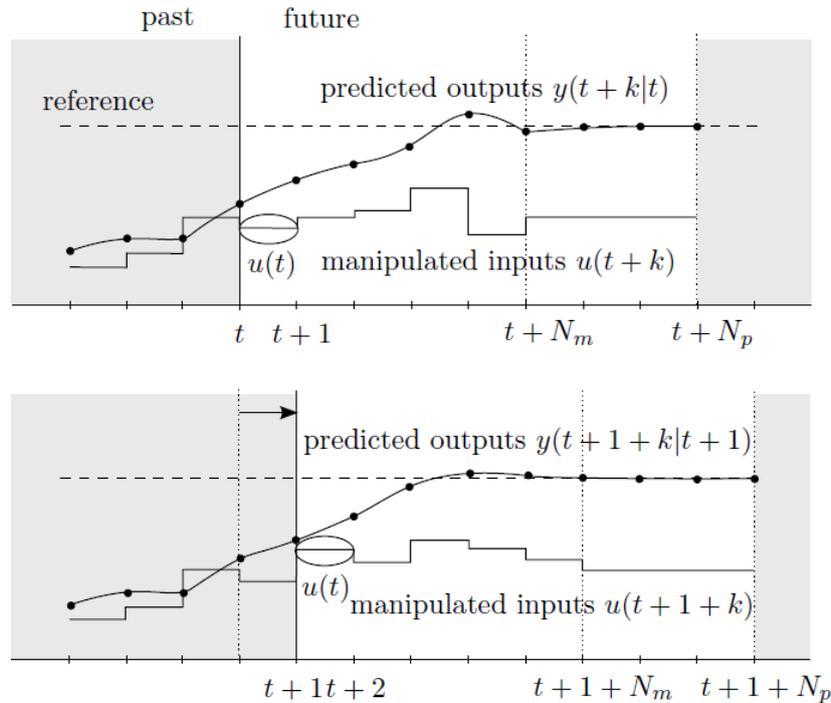


Figure 3-1 Receding Horizon Idea [73]

The schematic diagram shown in Figure 3-2 demonstrates the structure of an MPC controller, in which $\hat{\mathbf{x}}$ is the predicted state, \mathbf{x}_{ref} is the reference state and $\hat{\mathbf{u}}$ is the predicted input. The predicted input sequence is the solution of an optimization problem, which involves minimizing the quadratic cost function over a finite prediction horizon within the constraint set. The unknown disturbance affects the plant and is mitigated by a feedback integration strategy. The estimated disturbance is fed into the plant along with the prediction model to allow the controller to consider its effect and reduce the error.

In this chapter, the models will be illustrated first. Then the details of the optimization problem are introduced in the next section, including the objective function and constraints.

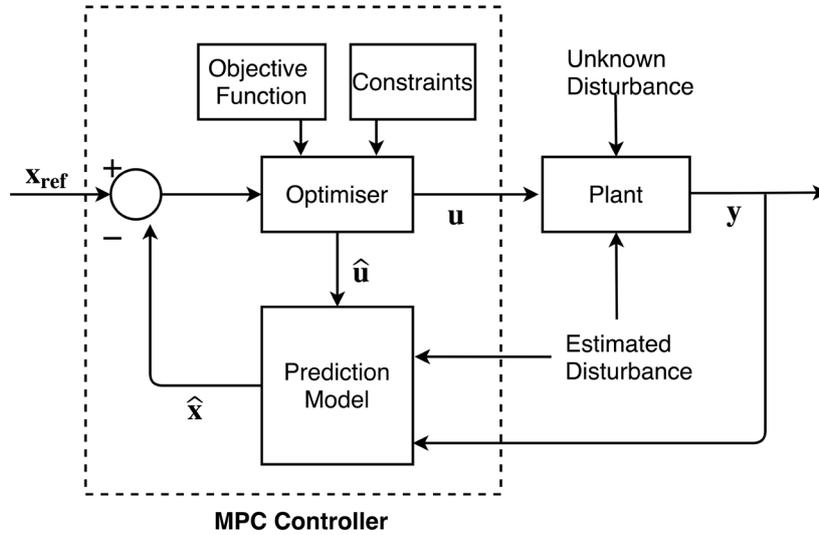


Figure 3-2 An Illustration of an MPC controller

3.1 MPC Models

The MPC method predicts future outputs and uses them to solve the optimization problem. The controller states estimation and plant outputs prediction are achieved by using the plant models, the estimated disturbance terms and the unknown disturbance terms. Each of them will be discussed in the following.

3.1.1 Estimated Disturbance Model

Normal feedback control strategy only takes corrective action until the effect of disturbance has become apparent, whereas feedforward control can anticipate and approximately canceled out the coming disturbance by suitable control actions [74]. Apply the feedforward strategy to remove most of the estimated disturbance and let the feedback control to remove the rest, for example, unknown disturbance and white noise. The effect of the disturbances on outputs is predicted, while the optimizer solves the problem with this effect. In the model, different wind speeds and wind directions will cause different aerodynamic forces, which will affect the vehicle's position. An experiment was conducted to measure the wind speed coming from a single wind direction. The experimental results of the mean wind speed and the variance of unsteady wind are presented in chapter 4.2.4 and used to estimate the disturbance. To describe the disturbance in a general form, let $\mathbf{v} \in \mathbb{R}^{n_v \times 1}$ becomes the input of estimated disturbance. The plant model can be written as

$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{A}_x \mathbf{x}(k) + \mathbf{B}_u \mathbf{u}(k) + \mathbf{B}_v \mathbf{v}(k) \\ \mathbf{y}(k) &= \mathbf{C}_x \mathbf{x}(k), \end{aligned} \tag{Eq. 3-1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x \times 1}$ is the state, $\mathbf{y} \in \mathbb{R}^{n_y \times 1}$ is the control output, and $\mathbf{u} \in \mathbb{R}^{n_u \times 1}$ is the control input. $\mathbf{A}_x \in \mathbb{R}^{n_x \times n_x}$ is the system matrix, $\mathbf{B}_u \in \mathbb{R}^{n_x \times n_u}$ is the control matrix and $\mathbf{B}_v \in \mathbb{R}^{n_x \times n_v}$ is the disturbance matrix. n_x, n_y, n_u, n_v are the number of state, output, input and estimated disturbance respectively.

3.1.2 Prediction Model

The prediction model is used by MPC controllers to predict the effect on future outputs caused by adjusting the control inputs. The prediction model can be built to investigate the system behavior in between the prediction horizon of that specific time step. Since the unknown disturbance cannot be predicted, the unknown disturbance term $\mathbf{d}(k)$ will be excluded.

The state was written in full length in the first place and propagated one step at a time to clarify its pattern.

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}_x \mathbf{x}(k) + \mathbf{B}_u [\mathbf{u}(k-1) + \Delta \mathbf{u}(k)] + \mathbf{B}_v \mathbf{v}(k) \\
\mathbf{x}(k+2) &= \mathbf{A}_x [\mathbf{A}_x \mathbf{x}(k) + \mathbf{B}_u [\mathbf{u}(k-1) + \Delta \mathbf{u}(k)] + \mathbf{B}_v \mathbf{v}(k)] \\
&\quad + \mathbf{B}_u [\mathbf{u}(k-1) + \Delta \mathbf{u}(k) + \Delta \mathbf{u}(k+1)] + \mathbf{B}_v \mathbf{v}(k+1) \\
\mathbf{x}(k+3) &= \mathbf{A}_x \{ \mathbf{A}_x [\mathbf{A}_x \mathbf{x}(k) + \mathbf{B}_u [\mathbf{u}(k-1) + \Delta \mathbf{u}(k)] + \mathbf{B}_v \mathbf{v}(k)] \\
&\quad + \mathbf{B}_u [\mathbf{u}(k-1) + \Delta \mathbf{u}(k) + \Delta \mathbf{u}(k+1)] + \mathbf{B}_v \mathbf{v}(k+1) \} \\
&\quad + \mathbf{B}_u [\mathbf{u}(k-1) + \Delta \mathbf{u}(k) + \Delta \mathbf{u}(k+1) + \Delta \mathbf{u}(k+2)] + \mathbf{B}_v \mathbf{v}(k+2) \\
&\quad \vdots \\
&\quad \vdots
\end{aligned} \tag{Eq. 3-2}$$

Then rearrange the terms and distinguish the constant and the variables.

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}_x \mathbf{x}(k) + \mathbf{B}_u \mathbf{u}(k-1) + \mathbf{B}_u \Delta \mathbf{u}(k) + \mathbf{B}_v \mathbf{v}(k) \\
\mathbf{x}(k+2) &= \mathbf{A}_x^2 \mathbf{x}(k) + (\mathbf{A}_x \mathbf{B}_u + \mathbf{B}_u) \mathbf{u}(k-1) + (\mathbf{A}_x \mathbf{B}_u + \mathbf{B}_u) \Delta \mathbf{u}(k) + \mathbf{A}_x \mathbf{B}_v \mathbf{v}(k) \\
&\quad + \mathbf{B}_u \Delta \mathbf{u}(k+1) + \mathbf{B}_v \mathbf{v}(k+1) \\
\mathbf{x}(k+3) &= \mathbf{A}_x^3 \mathbf{x}(k) + (\mathbf{A}_x^2 \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u + \mathbf{B}_u) \mathbf{u}(k-1) \\
&\quad + (\mathbf{A}_x^2 \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u + \mathbf{B}_u) \Delta \mathbf{u}(k) + \mathbf{A}_x^2 \mathbf{B}_v \mathbf{v}(k) + (\mathbf{A}_x \mathbf{B}_u \\
&\quad + \mathbf{B}_u) \Delta \mathbf{u}(k+1) + \mathbf{A}_x \mathbf{B}_v \mathbf{v}(k+1) + \mathbf{B}_u \Delta \mathbf{u}(k+2) + \mathbf{B}_v \mathbf{v}(k+2) \\
&\quad \vdots
\end{aligned} \tag{Eq. 3-3}$$

⋮

The predicted input $\hat{\mathbf{u}}$ can be expressed as the summation of the last control input $\mathbf{u}(k-1)$ and the input increment $\Delta\mathbf{u}$ as:

$$\begin{bmatrix} \hat{\mathbf{u}}(k|k) \\ \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \hat{\mathbf{u}}(k+H-1|k) \\ \vdots \\ \hat{\mathbf{u}}(k+P-1|k) \end{bmatrix} = \begin{bmatrix} I_1 \\ \vdots \\ I_P \end{bmatrix} \mathbf{u}(k-1) + \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ I & I & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & I \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I & I & \dots & \dots & I \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}(k|k) \\ \Delta\mathbf{u}(k+1|k) \\ \vdots \\ \Delta\mathbf{u}(k+H-1|k) \\ \vdots \\ \Delta\mathbf{u}(k+P-1|k) \end{bmatrix}, \quad \text{Eq. 3-4}$$

where $\Delta\mathbf{u}(k+i|k)$ is the input increment at the k^{th} time step and the i^{th} prediction horizon. The predicted input matrix \mathbf{U} can be simplified into:

$$\mathbf{U} = \mathbf{I}_{P \times 1} \mathbf{u}(k-1) + \mathbf{L}_I \Delta\mathbf{U}. \quad \text{Eq. 3-5}$$

By propagating one step at a time, as shown above, the predicted state $\hat{\mathbf{x}}(i|k)$ of the i^{th} prediction instant at the k time step. can be shown as:

$$\hat{\mathbf{x}}(k+i|k) = \mathbf{A}_x^i \mathbf{x}(k) + \sum_{j=0}^{i-1} \mathbf{A}_x^{i-1-j} \left[\mathbf{B}_u \mathbf{u}(k+j) + \mathbf{B}_v \mathbf{v}(k+j) \right], \quad \text{Eq. 3-6}$$

where i is prediction instant, and $i = [k, k+P], i \in \mathbb{Z}; j = [k, k+H], j \in \mathbb{Z}$. P is the number of prediction horizons and H is the number of control horizons, which $1 \leq H \leq P$.

The predicted output matrix \mathbf{Y} can be formulated by the predicted state matrix \mathbf{X}

$$\mathbf{y} = \mathbf{C}_x \mathbf{X} \quad \text{Eq. 3-7}$$

where $\mathbf{X} = \begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hat{\mathbf{x}}(k+2|k) \\ \hat{\mathbf{x}}(k+3|k) \\ \vdots \\ \hat{\mathbf{x}}(k+H|k) \\ \vdots \\ \hat{\mathbf{x}}(k+P|k) \end{bmatrix}$, Therefore, it can be summarized as

$$\mathbf{y} = \mathbf{S}^x \mathbf{x}(k) + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^{\Delta u} \Delta\mathbf{u} + \mathbf{S}^v \mathbf{v} \quad \text{Eq. 3-8}$$

where

$$\begin{aligned}
\mathbf{y} &= \begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \hat{\mathbf{y}}(k+2|k) \\ \hat{\mathbf{y}}(k+3|k) \\ \vdots \\ \hat{\mathbf{y}}(k+H|k) \\ \vdots \\ \hat{\mathbf{y}}(k+P|k) \end{bmatrix}, \quad \Delta \mathbf{u} = \begin{bmatrix} \Delta \mathbf{u}(k|k) \\ \Delta \mathbf{u}(k+1|k) \\ \Delta \mathbf{u}(k+2|k) \\ \vdots \\ \Delta \mathbf{u}(k+H-1|k) \\ \vdots \\ \Delta \mathbf{u}(k+P-1|k) \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}(k|k) \\ \mathbf{v}(k+1|k) \\ \mathbf{v}(k+2|k) \\ \vdots \\ \mathbf{v}(k+H-1|k) \\ \vdots \\ \mathbf{v}(k+P-1|k) \end{bmatrix}, \\
\mathbf{S}^x &= \mathbf{C}_x \begin{bmatrix} \mathbf{A}_x \\ \mathbf{A}_x^2 \\ \vdots \\ \mathbf{A}_x^H \\ \vdots \\ \mathbf{A}_x^P \end{bmatrix}, \quad \mathbf{S}^u = \mathbf{C}_x \begin{bmatrix} \mathbf{B}_u \\ \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u \\ \vdots \\ \sum_{i=0}^{H-1} \mathbf{A}_x^i \mathbf{B}_u \\ \vdots \\ \sum_{i=0}^{P-1} \mathbf{A}_x^i \mathbf{B}_u \end{bmatrix}, \quad \mathbf{S}^v = \mathbf{C}_x \begin{bmatrix} \mathbf{B}_v & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A} \mathbf{B}_v & \mathbf{B}_v & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{H-1} \mathbf{B}_v & \vdots & \vdots & \vdots & \mathbf{B}_v \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}^{P-1} \mathbf{B}_v & \dots & \dots & \dots & \mathbf{B}_v \end{bmatrix}, \quad \text{Eq. 3-9} \\
\mathbf{S}^{\Delta u} &= \mathbf{C}_x \begin{bmatrix} \mathbf{B}_u & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u & \mathbf{B}_u & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{H-1} \mathbf{A}_x^i \mathbf{B}_u & \vdots & \vdots & \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u & \mathbf{B}_u \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=0}^{P-1} \mathbf{A}_x^i \mathbf{B}_u & \dots & \dots & \dots & \sum_{i=0}^{P-H} \mathbf{A}_x^i \mathbf{B}_u \end{bmatrix}.
\end{aligned}$$

3.1.3 Unknown Disturbance Model

It is practically impossible to accurately model all the disturbances which are acting to the system. A feedback integration strategy is introduced to eliminate the effects caused by the unknown disturbances, model mismatch, and measurement noise. By adding extra terms to the state-space model, the MPC controller can achieve offset-free control. This is a similar method as the integration effect of a PID control. The input of unknown disturbance is expressed as follow:

$$\mathbf{d}(k) = \mathbf{y}_{ref}(k) - \mathbf{y}(k), \quad \text{Eq. 3-10}$$

where \mathbf{d} is the unmeasured disturbance, which is the error between the measured output and the reference output. In this study, the \mathbf{C}_x matrix is assumed to be an identity matrix in Eq. 3-1, which means all the states are assumed to be fully measurable. This assumption is made upon the onboard sensors we have used on the UAV, such as the GPS can measure position and speed; the IMU can measure the linear/angular acceleration. The point will be discussed in the next chapter. As a result, $\mathbf{d} \in \mathbb{R}^{n_y \times 1}$ and $n_y = n_x$, then the system model was developed as

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}_x \mathbf{x}(k) + \mathbf{B}_u \mathbf{u}(k) + \mathbf{B}_v \mathbf{v}(k) + \mathbf{x}_d(k) + \mathbf{d}(k) \\
\mathbf{x}_d(k+1) &= \mathbf{x}_d(k) + \mathbf{d}(k)
\end{aligned} \quad \text{Eq. 3-11}$$

$$\mathbf{y}(k) = \mathbf{C}_x \mathbf{x}(k) + \mathbf{x}_d(k).$$

where \mathbf{x}_d is the accumulation of unmeasured disturbance, which can be augmented to the system state.

The augmented state-space model can be summarised as follows

$$\begin{aligned} \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}_d(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_x & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_d(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_u \\ \mathbf{0} \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} \mathbf{B}_v \\ \mathbf{0} \end{bmatrix} \mathbf{v}(k) + \mathbf{d}(k) \\ \mathbf{y}(k) &= [\mathbf{C}_x \quad \mathbf{I}] \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_d(k) \end{bmatrix} \end{aligned} \quad \text{Eq. 3-12}$$

And then being simplified into

$$\begin{aligned} \mathbf{x}_A(k+1) &= \mathbf{A} \mathbf{x}_A(k) + \mathbf{B} \mathbf{u}_t(k) \\ \mathbf{y}(k) &= \mathbf{C} \mathbf{x}_A(k), \end{aligned} \quad \text{Eq. 3-13}$$

where $\mathbf{x}_A(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_d(k) \end{bmatrix}$, $\mathbf{A} = \begin{bmatrix} \mathbf{A}_x & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$, $\mathbf{u}_t = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{v}(k) \\ \mathbf{d}(k) \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} \mathbf{B}_u & \mathbf{B}_v & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$, $\mathbf{C} = [\mathbf{C}_x \quad \mathbf{I}]$.

3.2 Optimization and Stability Problem

The optimization problem includes an objective function, which can be specifically designed according to need and the set of constraints, which most of the time, according to the physical system. The designed objective function and the constraints will be integrated with the prediction model and reduced to its standard form.

3.2.1 Objective Function

The squared 2-norm form is employed more often in the objective function than the 1- or ∞ -norm for MPC design. Because solving the 1- or ∞ -norm formulation will lead to solving Linear Programming (LP) problems, which solution always lies at the intersection of constraints and changes discontinuously as the tuned parameters are varied. This brings difficulties in the formulation of the control problem. On the other hand, the 1- or ∞ -norm formulation involves many more constraints than the 2-norm formulation. Thus, in general, the 1- or ∞ -norm will lead to a larger number of regions of the explicit control law. As a result, the 2-norm form was chosen. The objective function was designed as

$$\begin{aligned} \min J(\mathbf{z}_k) &= \sum_{i=0}^{P-1} \{ [\mathbf{e}_y^T(k+i) \mathbf{Q} \mathbf{e}_y(k+i)] + [\mathbf{e}_u^T(k+i) \mathbf{R}_u \mathbf{e}_u(k+i)] + \\ &[\Delta \mathbf{u}^T(k+i) \mathbf{R}_{\Delta u} \Delta \mathbf{u}(k+i)] \}, \end{aligned} \quad \text{Eq. 3-14}$$

where \mathbf{Q} , \mathbf{R}_u and $\mathbf{R}_{\Delta u}$ are positive-semi-definite matrices with weight information in scalar of their diagonal, \mathbf{e}_y , \mathbf{e}_u and $\Delta \mathbf{u}$ represent error on output, error on input and the input increment, respectively. \mathbf{e}_y , \mathbf{e}_u and $\Delta \mathbf{u}$ can be expressed as

$$\begin{aligned} \mathbf{e}_y(k+i) &= \mathbf{S}_y^{-1}[\mathbf{r}(k+i+1|k) - \mathbf{y}(k+i+1|k)] \\ \mathbf{e}_u(k+i) &= \mathbf{S}_u^{-1}[\mathbf{u}_{target}(k+i|k) - \mathbf{u}(k+i|k)] \\ \Delta \mathbf{u}(k+i) &= \mathbf{S}_u^{-1}[\mathbf{u}(k+i|k) - \mathbf{u}(k+i-1|k)], \end{aligned} \quad \text{Eq. 3-15}$$

where, $\mathbf{r}(k+i|k)$ is the plant output reference value at the i^{th} prediction horizon step. The optimization process included the error between the current reference value \mathbf{r} and the predicted output \mathbf{y} in chapter 3.1.2 such that the controller can take the future into account. If reference values are loaded into the optimizer in advance from 1 to $P-1$ steps, the optimiser can generate a predicted input that accounts for future error. If not, every reference value is considered a constant for the horizon. \mathbf{S}_y and \mathbf{S}_u are diagonal matrix of scale factors of plant output and input in engineering units.

Apart from the setup of the objective function, the variables are constrained according to the physical system and expressed as follows:

$$\begin{aligned} \mathbf{y}_{j,min}(i) &\leq \mathbf{y}_j(k+i|k) \leq \mathbf{y}_{j,max}(i), & \text{for } i = 1:P, j = 1:n_y \\ \mathbf{u}_{j,min}(i) &\leq \mathbf{u}_j(k+i-1|k) \leq \mathbf{u}_{j,max}(i), & \text{for } i = 1:P, j = 1:n_u \\ \Delta \mathbf{u}_{j,min}(i) &\leq \Delta \mathbf{u}_j(k+i-1|k) \leq \Delta \mathbf{u}_{j,max}(i), & \text{for } i = 1:P, j = 1:n_u \end{aligned} \quad \text{Eq. 3-16}$$

where n_y and n_u are the number of output and input. $\mathbf{y}_{j,min}(i)$ and $\mathbf{y}_{j,max}(i)$ are lower and upper bounds for the j^{th} plant output at the i^{th} prediction horizon step, $\mathbf{u}_{j,min}(i)$, $\mathbf{u}_{j,max}(i)$, $\Delta \mathbf{u}_{j,min}(i)$ and $\Delta \mathbf{u}_{j,max}(i)$ are having similar meanings but of input and input increment.

When the objective function and constraints are determined, the quadratic programming (QP) problem can be derived. An open-loop optimization is repeated until the control horizon H is reached to eliminate the error between the reference state and the predicted state within the prediction horizon P . The QP decision is obtained as follows:

$$\mathbf{z}_k^T = [\mathbf{u}(k|k)^T \quad \mathbf{u}(k+1|k)^T \quad \dots \quad \mathbf{u}(k+H-1|k)^T \quad \dots \quad \mathbf{u}(k+P-1|k)^T] \quad \text{Eq. 3-17}$$

where the control input \mathbf{u} will be calculated by solving the QP problem until the $H-1$ step and stays at the same value until the $P-1$ step. Only the first term is implemented as a control effort at each time step, according to the receding horizon control algorithm.

Within the prediction horizon of a time step, the control input of the last time step $\mathbf{u}(k-1)$ is a constant, but the $\Delta\mathbf{u}$ at each prediction step varies. It will create a new series of control input $\mathbf{u}(k)$ up to $\mathbf{u}(k+P-1)$ according to Eq. 3-4. However, the control effort series $\mathbf{z}(k)$ to $\mathbf{z}(k+H-1)$ will stop varying if it reaches the control horizon. Thus, their relation can be expressed as

$$\begin{aligned} \mathbf{u}(k-1) + \Delta\mathbf{u}(k) &= \mathbf{u}(k) = \mathbf{z}(k) \\ \mathbf{u}(k-1) + \Delta\mathbf{u}(k) + \Delta\mathbf{u}(k+1) &= \mathbf{u}(k+1) = \mathbf{z}(k+1) \\ &\vdots \\ &\vdots \end{aligned} \tag{Eq. 3-18}$$

Therefore, ignoring the constant and combining the variables, the relation can be expressed as

$$\begin{bmatrix} \Delta\mathbf{u}(k) \\ \vdots \\ \Delta\mathbf{u}(k+P-1) \end{bmatrix} = \mathbf{J}_M \begin{bmatrix} \mathbf{z}(k) \\ \vdots \\ \mathbf{z}(k+H-1) \end{bmatrix}, \tag{Eq. 3-19}$$

where the vector $[\mathbf{z}(k), \dots, \mathbf{z}(k+H-1)]$ comprise the optimization variables until the control horizon N_m . \mathbf{J}_M is a matrix in size $(H-1 \times P-1)$.

Then, the objective function in Eq. 3-14 will be expanded in detail and gradually substitute the already derived terms until the standard form of an objective function is reached. The expanded form is shown as follow:

$$\begin{aligned} \min J(\mathbf{z}_k) &= \left(\begin{bmatrix} \mathbf{y}(k) \\ \vdots \\ \mathbf{y}(P) \end{bmatrix} - \begin{bmatrix} \mathbf{r}(k) \\ \vdots \\ \mathbf{r}(P) \end{bmatrix} \right)^T \mathbf{Q} \left(\begin{bmatrix} \mathbf{y}(k) \\ \vdots \\ \mathbf{y}(P) \end{bmatrix} - \begin{bmatrix} \mathbf{r}(k) \\ \vdots \\ \mathbf{r}(P) \end{bmatrix} \right) \\ &+ \left(\begin{bmatrix} \mathbf{u}(k) \\ \vdots \\ \mathbf{u}(k+P-1) \end{bmatrix} - \begin{bmatrix} \mathbf{u}_{target}(k) \\ \vdots \\ \mathbf{u}_{target}(k+P-1) \end{bmatrix} \right)^T \mathbf{R}_u \left(\begin{bmatrix} \mathbf{u}(k) \\ \vdots \\ \mathbf{u}(k+P-1) \end{bmatrix} - \begin{bmatrix} \mathbf{u}_{target}(k) \\ \vdots \\ \mathbf{u}_{target}(k+P-1) \end{bmatrix} \right) \\ &+ \left(\begin{bmatrix} \Delta\mathbf{u}(k) \\ \vdots \\ \Delta\mathbf{u}(k+P-1) \end{bmatrix} \right)^T \mathbf{R}_{\Delta\mathbf{u}} \left(\begin{bmatrix} \Delta\mathbf{u}(k) \\ \vdots \\ \Delta\mathbf{u}(k+P-1) \end{bmatrix} \right). \end{aligned} \tag{Eq. 3-20}$$

Firstly, replace the predicted output $\begin{bmatrix} \mathbf{y}(k) \\ \vdots \\ \mathbf{y}(P) \end{bmatrix}$ by the prediction matrix \mathbf{Y} according to Eq. 3-8. Then, the objective function became

$$\begin{aligned} \min J(\mathbf{z}_k) &= \\ &(\mathbf{S}^x \mathbf{x}(k) + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^v \mathbf{v} - \mathbf{r}(k))^T \mathbf{Q} (\mathbf{S}^x \mathbf{x}(k) + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^v \mathbf{v} - \mathbf{r}(k)) \end{aligned} \tag{Eq. 3-21}$$

$$+2(\mathbf{r}(k) + \mathbf{S}^v \mathbf{v} + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^x \mathbf{x}(k)) \Delta \mathbf{u} + \Delta \mathbf{u}^T (\mathbf{S}^{\Delta u T} \mathbf{Q} \mathbf{S}^{\Delta u} + \mathbf{R}_u) \Delta \mathbf{u} \\ + (\mathbf{U} - \mathbf{U}_{target})^T \mathbf{R}_u (\mathbf{U} - \mathbf{U}_{target}).$$

Let $\mathbf{c}_y = \mathbf{S}^x \mathbf{x}(k) + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^v \mathbf{v} - \mathbf{r}(k)$, and replace $\begin{bmatrix} \mathbf{u}(k) \\ \vdots \\ \mathbf{u}(k + N_p - 1) \end{bmatrix}$ by Eq. 3-5. Then the objective function became

$$\mathbf{min} J(\mathbf{z}_k) = \mathbf{c}_y^T \mathbf{Q} \mathbf{c}_y \\ + (\mathbf{I}_{P \times 1} \mathbf{u}(k-1) + \mathbf{L}_I \Delta \mathbf{u} - \mathbf{U}_{target})^T \mathbf{R}_u (\mathbf{I}_{P \times 1} \mathbf{u}(k-1) + \mathbf{L}_I \Delta \mathbf{u} - \mathbf{U}_{target}) \\ + 2(\mathbf{r}(k) + \mathbf{S}^v \mathbf{v} + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^x \mathbf{x}(k)) \Delta \mathbf{u} \\ + \Delta \mathbf{u}^T (\mathbf{S}^{\Delta u T} \mathbf{Q} \mathbf{S}^{\Delta u} + \mathbf{R}_{\Delta u}) \Delta \mathbf{u}. \quad \text{Eq. 3-22}$$

Let $\mathbf{c}_u = \mathbf{I}_{P \times 1} \mathbf{u}(k-1) - \mathbf{U}_{target}$, and substitute $\begin{bmatrix} \Delta \mathbf{u}(k) \\ \vdots \\ \Delta \mathbf{u}(k + N_p - 1) \end{bmatrix}$ by Eq. 3-19 and finally, the standard form of the objective function can be rewritten as

$$\mathbf{min} J(\mathbf{z}_k) = \mathbf{c}_y^T \mathbf{Q} \mathbf{c}_y + \mathbf{c}_u^T \mathbf{R}_u \mathbf{c}_u + \mathbf{z}^T \mathbf{K}_{\Delta u} \mathbf{z} + 2 \mathbf{K}_y \mathbf{z}, \quad \text{Eq. 3-23}$$

where

$$\mathbf{K}_{\Delta u} = (\mathbf{J}_M \mathbf{S}^{\Delta u})^T \mathbf{Q} (\mathbf{J}_M \mathbf{S}^{\Delta u}) + (\mathbf{J}_M \mathbf{L}_I)^T \mathbf{R}_u (\mathbf{J}_M \mathbf{L}_I) + \mathbf{J}_M^T \mathbf{R}_{\Delta u} \mathbf{J}_M \\ \mathbf{K}_y = \mathbf{r}(k)^T \mathbf{J}_M + (\mathbf{S}^v \mathbf{v})^T \mathbf{J}_M + [\mathbf{S}^u \mathbf{u}(k-1)]^T \mathbf{J}_M + \mathbf{U}_{target}^T \mathbf{J}_M + [\mathbf{S}^x \mathbf{x}(k)]^T \mathbf{J}_M$$

3.2.2 Constraints

The same method above can also be applied to the constrains, transforming the bounds into polyhedral constraints by variables substitution. The bound constraint was defined as

$$\begin{bmatrix} y_{min}(k) \\ y_{min}(k+P) \\ u_{min}(k) \\ u_{min}(k+P-1) \\ \Delta u_{min}(k) \\ \Delta u_{min}(k+P) \end{bmatrix} \leq \begin{bmatrix} y(k) \\ y(k+P) \\ u(k) \\ u(k+P-1) \\ \Delta u(k) \\ \Delta u(k+P-1) \end{bmatrix} \leq \begin{bmatrix} y_{max}(k) \\ y_{max}(k+P) \\ u_{max}(k) \\ u_{max}(k+P-1) \\ \Delta u_{max}(k) \\ \Delta u_{max}(k+P) \end{bmatrix}. \quad \text{Eq. 3-24}$$

The control output term $y(k)$ can be substitute with Eq. 3-8 then the output inequality can be expressed as:

$$\mathbf{S}^x \mathbf{x}(k) + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^{\Delta u} \Delta \mathbf{u} + \mathbf{S}^v \mathbf{v} \leq \mathbf{y}_{lim}(k). \quad \text{Eq. 3-25}$$

The control input term $u(k)$ can be replaced by Eq. 3-5 and the input inequality can be derived as:

$$\mathbf{I}_{P \times 1} \mathbf{u}(k-1) + \mathbf{L}_I \Delta \mathbf{u} \leq \mathbf{u}_{lim}. \quad \text{Eq. 3-26}$$

Then the input increment term $\Delta \mathbf{u}(k)$ can be replaced by Eq. 3-19 and the constraint Eq. 3-24 can be expressed as:

$$\begin{aligned} (\mathbf{S}^{\Delta u} \mathbf{J}_M) \mathbf{z} &\leq \mathbf{y}_{lim} - \mathbf{S}^v \mathbf{v} - \mathbf{S}^u \mathbf{u}(k-1) - \mathbf{S}^x \mathbf{x}(k) \\ (\mathbf{L}_I \mathbf{J}_M) \mathbf{z} &\leq \mathbf{u}_{lim} - \mathbf{I}_{P \times 1} \mathbf{u}(k-1) \\ \mathbf{J}_M \mathbf{z} &\leq \Delta \mathbf{u}_{lim} \end{aligned} \quad \text{Eq. 3-27}$$

Finally, the polyhedral constraints can be simplified as:

$$\mathbf{M}_z \mathbf{z} \leq \mathbf{M}_{lim} + \mathbf{M}_v \mathbf{v} + \mathbf{M}_u \mathbf{u}(k-1) + \mathbf{M}_x \mathbf{x}(k) \quad \text{Eq. 3-28}$$

where $\mathbf{M}_z = \begin{bmatrix} \mathbf{S}^{\Delta u} \mathbf{J}_M \\ \mathbf{L}_I \mathbf{J}_M \\ \mathbf{J}_M \end{bmatrix}$, $\mathbf{M}_{lim} = \begin{bmatrix} \mathbf{y}_{lim} \\ \mathbf{u}_{lim} \\ \Delta \mathbf{u}_{lim} \end{bmatrix}$, $\mathbf{M}_v = \begin{bmatrix} -\mathbf{S}^v \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$, $\mathbf{M}_u = \begin{bmatrix} -\mathbf{S}^u \\ -\mathbf{I} \\ \mathbf{0} \end{bmatrix}$ and $\mathbf{M}_x = \begin{bmatrix} -\mathbf{S}^x \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$.

At this point, the MPC optimization problem has been set up with an objective function as

$$\min J(\mathbf{z}_k) = \mathbf{c}_y^T \mathbf{Q} \mathbf{c}_y + \mathbf{c}_u^T \mathbf{R}_u \mathbf{c}_u + \mathbf{z}^T \mathbf{K}_{\Delta u} \mathbf{z} + 2\mathbf{K}_y \mathbf{z}$$

Subjected to. Eq. 3-29

$$\mathbf{M}_z \mathbf{z} \leq \mathbf{M}_{lim} + \mathbf{M}_v \mathbf{v} + \mathbf{M}_u \mathbf{u}(k-1) + \mathbf{M}_x \mathbf{x}(k)$$

where $J(\mathbf{z}_k)$ is a positive definite QP function. The MPC optimization problem will be solved by a KWIK algorithm [75] in Matlab Simulink.

3.2.3 Stability

From Chapter 7 of [76], It is able to achieve stability in practice by parameters tuning, including weight, cost function and horizon parameters. One of the stability modifications of the MPC algorithm was to take into account an additional terminal constraint, the objective of which is to bring the predicted state at the end of the prediction horizon to an equilibrium point.

$$\min J(\mathbf{z}_k) = \mathbf{x}_N^T \mathbf{P} \mathbf{x}_N + \sum_{i=0}^{N-1} \{ \mathbf{x}_i^T \mathbf{Q} \mathbf{x}_i + \mathbf{u}_i^T \mathbf{R} \mathbf{u}_i \}$$

$$\mathbf{x}_{i+1} = \mathbf{A} \mathbf{x}_i + \mathbf{B} \mathbf{u}_i$$

$$\mathbf{x}_i \in \mathcal{X}$$

Eq. 3-30

$$\mathbf{u}_k \in \mathcal{U}$$

$$\mathbf{x}_N \in \mathcal{X}_f$$

$$\mathbf{x}_0 \in \mathbf{x}(k)$$

where \mathbf{x}_N is the terminal state and it stays in the convex space of \mathcal{X}_f . When the cost function of an MPC control is cooperates with terminal state and terminal constraint, stability can be proved by showing that the optimal cost function is a Lyapunov function.

4. Hover Flight Control

In this chapter, the tail-sitter UAV and its model is integrated into the simulation environment. The coordinate system was defined, and the equations of motion are summarized. The details of the design and manufacturing of the vehicle are described, followed by the chosen onboard avionics and propulsion system. Then the property of the propulsion system was modelled by a thrust experiment and the aerodynamic effect was theoretically modelled.

4.1 System Configuration and Modelling

It is suggested the torque generated by propellers should be balanced during hovering when designing a tail-sitter vehicle [77]. Therefore, a quadrotor tail-sitter type is chosen in this study. Figure 4-1 shows the developed tail-sitter.



Figure 4-1 The testbed tail-sitter: PolyU Plus

This tail-sitter UAV is modified from a commercial flying wing UAV platform ‘Skywalker X-5’ with its rear motor being removed and replaced by four puller motors. This UAV is different from the quadrotor type, as it has wings and therefore it has different aerodynamic effects. It uses four motors and propellers as the actuators, instead of the rudder, elevators, and ailerons used in fixed-wing aircraft. The ‘plus’ shape is chosen because this configuration better utilizes the wing. The propeller wash

passes through a major part of the wing and increases the airspeed on this wing segment. However, it also results in an unwanted aerodynamic force and moment during the hover phase, which must be corrected by the flight controller. Two 3D-printed motor seats are attached to the leading edges of the wing on both sides using epoxy resin. A plywood board was laser cut and assembled like the other two motor seats. Carbon tubes were used to mount the motor seats on the upper and lower sides of the fuselage. A 4S 2600mAh Li-po battery was installed at a position that leads to a good center of gravity. The final prototype has a wingspan of 1.1 m, with a mean aerodynamic chord (MAC) of 0.4 m and an operating weight of 1.67 kg.

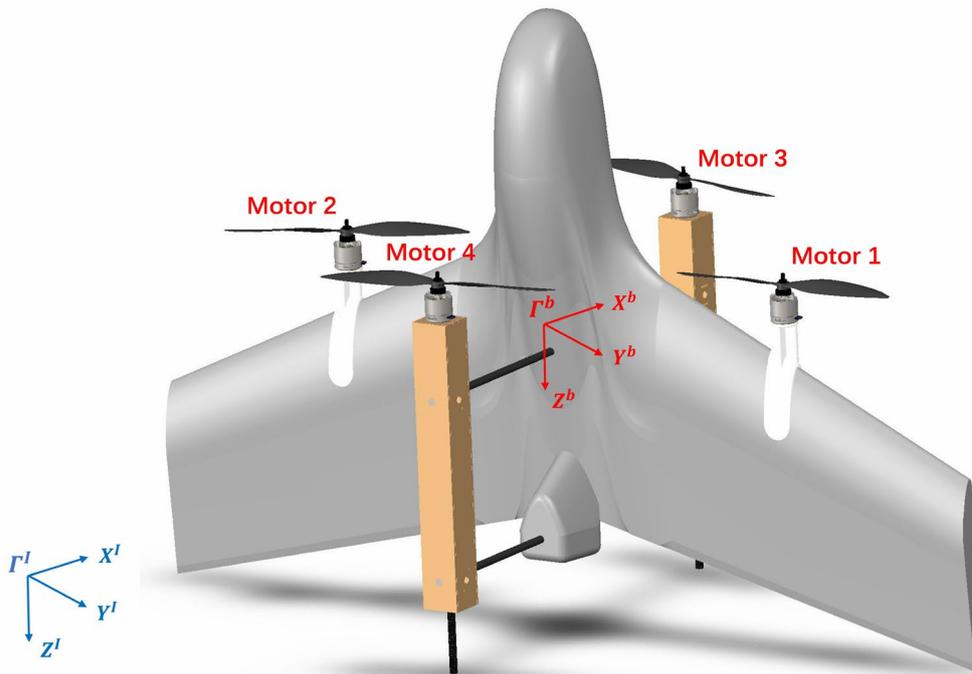


Figure 4-2 Schematic of coordinate systems

To describe the dynamic of UAVs, two sets of reference systems are required. The fixed inertial coordinate system $(\Gamma^I: X^I, Y^I, Z^I)$ points to the north, east and downward directions. The mobile body coordinate system $(\Gamma^b: X^b, Y^b, Z^b)$, which is located at the vehicle's center of mass, has an x -axis, y -axis and z -axis pointing to motor 3, motor 1 and the tail of the vehicle in Figure 4-2.

4.1.1 Dynamic Modelling

The dynamic model of the UAV can be described as

$$\dot{\mathbf{p}}^I = \mathbf{R}_b^I \mathbf{v}^b \tag{Eq. 4-1}$$

$$\boldsymbol{\Omega} = \mathbf{T}_b^I \boldsymbol{\omega}^b$$

$$m(\boldsymbol{\omega}^b \times \mathbf{v} + \dot{\mathbf{v}}) = \mathbf{F}^b$$

$$\mathbf{I} \dot{\boldsymbol{\omega}}^b + \boldsymbol{\omega}^b \times (\mathbf{I} \boldsymbol{\omega}^b) = \mathbf{M}^b,$$

where $\dot{\mathbf{p}}^I = [\dot{x} \ \dot{y} \ \dot{z}]^T \in \mathbb{R}^3$ is the derivative of position in inertia frame, and $\mathbf{v}^b = [u \ v \ w]^T \in \mathbb{R}^3$ is the velocity in Γ^I , respectively; $\mathbf{R}_b^I \in \text{SO}(3)$ is the rotation matrix that transfers Γ^b into Γ^I ; $\boldsymbol{\Omega} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \in \mathbb{R}^3$ is the time derivative of the Euler angles (roll, pitch and yaw); $\boldsymbol{\omega} = [p \ q \ r]^T \in \mathbb{R}^3$ is the angular velocity in Γ^b ; $\mathbf{T}_b^I \in \text{SO}(3)$ is the non-singular matrix that relates the angular velocities to the rates of the Euler angles; \mathbf{I} is the inertial matrix; and m is the mass of the vehicle.

\mathbf{F}_b and \mathbf{M}_b are the force and moment in Γ^b , respectively, and are expressed as follows:

$$\mathbf{F}_b = \mathbf{F}_{aero} + \mathbf{F}_P + \mathbf{R}_I^b \mathbf{F}_g + \mathbf{F}_d$$

Eq. 4-2

$$\mathbf{M}_b = \mathbf{M}_{aero} + \mathbf{M}_P + \mathbf{M}_d.$$

where \mathbf{F}_{aero} and \mathbf{M}_{aero} are the aerodynamic force and moment, respectively; \mathbf{F}_g is the gravity force in Γ^I ; \mathbf{F}_P is the thrust on the negative z-direction; \mathbf{M}_P is the moment that is created by the propulsion system; The disturbance and model error is described by \mathbf{F}_d and \mathbf{M}_d .

4.1.2 Propulsion System Modelling

The nonlinear relationship between the throttle commands and the force and moment of the propulsion system is determined by a set of experiments. Experiments have been conducted to investigate the relationship of the throttle commands to the force and moment of the propulsion system. During hover condition, the wind speed was assumed to be zero, as a result, there was no coming flow. An ATI Mini40 6-DOF Force/Torque sensor was bolted onto an aluminum bar. The selected motor and propeller were mounted on the sensor and tested. The sensor transduces the measured thrust and torque values in analog voltage and collected by the data acquisition (DAQ) system. The generated thrust and moments of the motor-propeller were recorded for 5 seconds at every 10% of throttle. The experimental setup is shown in Figure 4-3. During the experiment, several propellers and motors pair were selected and tested. According to the thrust requirement of the vehicle, the combination of Sunnysky x2212-980kv brushless motor, 30A ESC, and APC1047 propellers was selected as the propulsion system. It can create more than 3N in thrust at 50% of throttle and provide a maximum thrust/weight ratio of around 2.7.

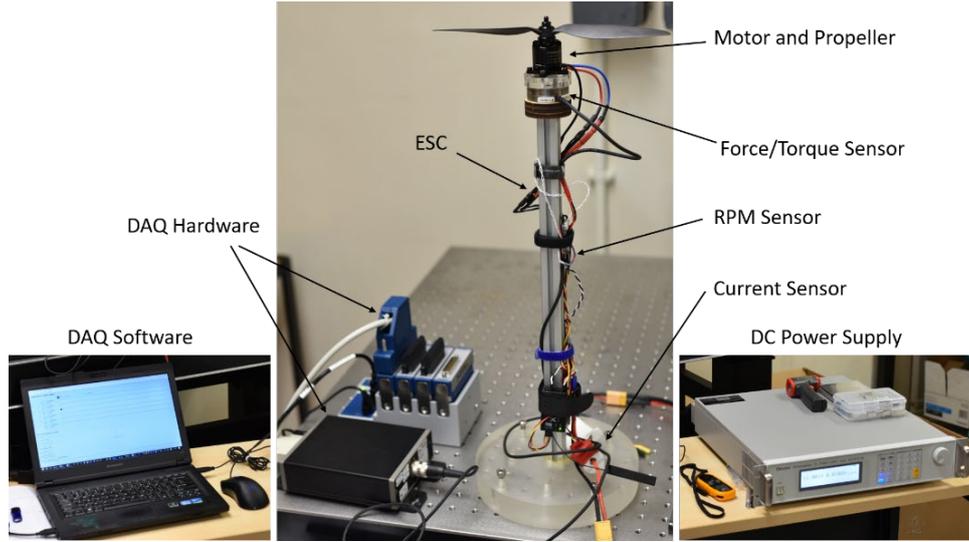


Figure 4-3 Experimental setups of propulsion system modelling

A fourth-order polynomial function was used to fit the nonlinear relationship of throttle commands to the thrust. The experimental data and the fitted polynomial for thrust and moment of the designed propulsion system are shown in Figure 4-4.

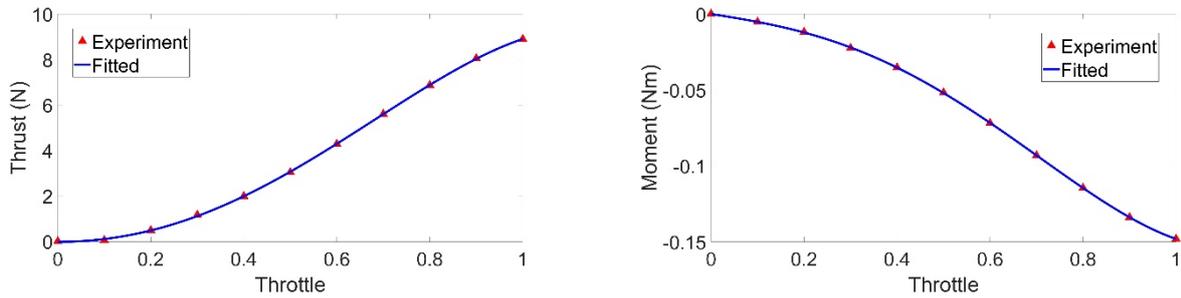


Figure 4-4 The experiment data and fitted curve of the motor-propeller system

The thrust and moment created by the propulsion system can be estimated through

$$\begin{aligned}
 \mathbf{F}_P &= \begin{bmatrix} 0 \\ 0 \\ -(f_{T1} + f_{T2} + f_{T3} + f_{T4}) \end{bmatrix} \\
 \mathbf{M}_P &= \begin{bmatrix} (f_{T2} - f_{T1})l_y \\ (f_{T3} - f_{T4})l_x \\ M_1 + M_2 - M_3 - M_4 \end{bmatrix}.
 \end{aligned}
 \tag{Eq. 4-3}$$

where f_{Ti} and M_i as $i \in [1,4]$ are the thrust and moment of each propeller, and l_y and l_x are the moment arm along the y-axis and x-axis.

4.1.3 Aerodynamic Modelling

For the current UAV configuration, a large portion of the wing of the vehicle is immersed in the induce flow generated by the propellers. The model of the aerodynamics will be integrated into the software-in-loop (SIL) simulation environment as well as feedforward disturbance measurement models of the control system. The speed of the airflow is considered uniform over the wing and it is assumed that there is no cross-coupling between the left and right wings. The side force Y which yields the aerodynamic effect on the y -axis of the vehicle is assumed zero, as its magnitude is much smaller than the lift force L and drag force D . The schematic diagram of Figure 4-5 shows the steps of estimation of the aerodynamic force and moment.

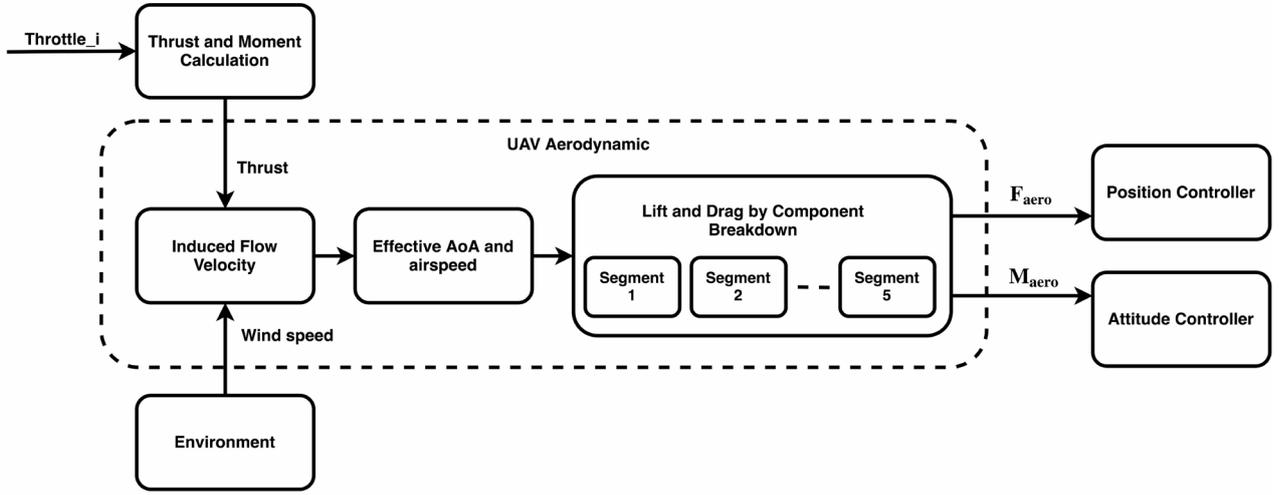


Figure 4-5 The schematic diagram of aerodynamic modelling

Induce Flow and Velocity

The velocity of the flow field was strongly affected by the slipstream of the propellers, which were changed consistently according to the throttle command. To obtain the flow condition on the wing, the freestream velocity in body coordinate, V_{as}^B , can be calculated by adding the contribution of propeller wash in the flow field, which can be written as

$$V_{as}^B = R_I^B (V_{wind}^I + V_{gs}^I) + V_{induce}^B \quad \text{Eq. 4-4}$$

The wind speed V_{wind}^I and ground speed V_{gs}^I are described in the inertia frame (NED) and R_I^b is the transformation matrix from the inertia frame to the body frame. V_{induce}^B is the propeller slipstream velocity, which is defined as

$$\mathbf{V}_{induce}^B = \begin{bmatrix} 0 \\ 0 \\ v_{induce} \end{bmatrix}. \quad \text{Eq. 4-5}$$

By using the continuity equation [78] v_{induce} can be estimated as

$$v_{induce} = \frac{v_{free} \cos \alpha + \sqrt{v_{free}^2 \cos^2 \alpha + \frac{2f_T}{\rho A}}}{2} \left[1 + \frac{s/R}{\sqrt{1+(s/R)^2}} \right]. \quad \text{Eq. 4-6}$$

where α is the angle of attack (AoA), f_T is the thrust generated by the propeller, A is the area of propeller disk, s is the distance from the center of propeller disk to the point of interest, and R is the radius of the propeller disk. v_{free} is the velocity of freestream, which can be expressed as

$$v_{free} = |\mathbf{V}_{wind}^I + \mathbf{V}_{gs}^I|. \quad \text{Eq. 4-7}$$

Effective AoA and Airspeed

The angle of attack can be defined as

$$\alpha = \tan^{-1}(V_{as,x}^b/V_{as,z}^b). \quad \text{Eq. 4-8}$$

where $V_{as,x}^b$ and $V_{as,z}^b$ are the local airspeed velocity in the direction of the x -axis and z -axis in body coordinate. Then the effective airspeed is

$$v_{eff} = |\mathbf{V}_{as}^B|. \quad \text{Eq. 4-9}$$

As mentioned, the side force and moment were ignored in the hover stage. A database of the vehicle is used for searching the lift, drag and moment coefficient at different conditions of the angle of attack and airspeed, which includes the lift coefficient C_L , drag coefficient C_D , and moment coefficient C_M at α from $-180^\circ \sim 180^\circ$ and airspeed from $0 \sim 30 \text{ m/s}$. This database was formed based on wind tunnel experiments [79].

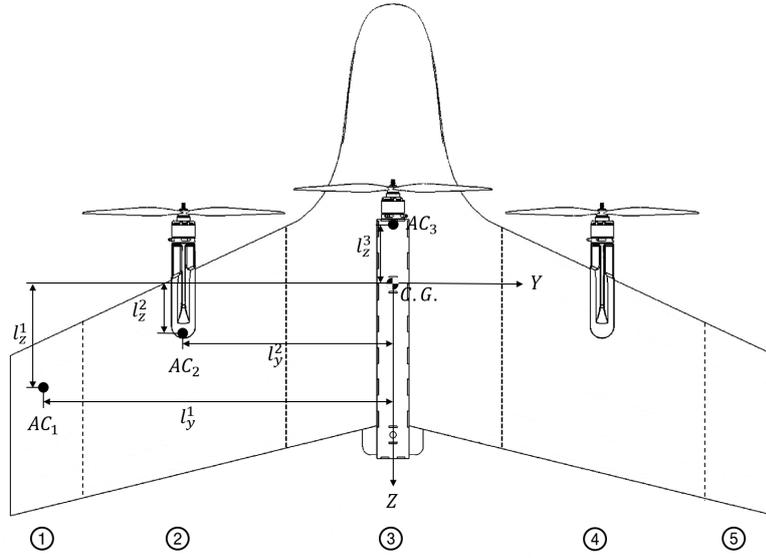


Figure 4-6 The component breakdown method

The vehicle was modeled into five parts based on the component breakdown approach [80], which is shown in Figure 4-6, to distinguish the segments with and without induced flow. The mean aerodynamic chord (MAC) and the position of the aerodynamic center (AC) of the whole wing were calculated first. By applying the Bernoulli theory and the Momentum theory of flow, r is the radius of the flow tube can be expressed as Eq. 4-10 according to [78]. Then the width of segments 2 & 4 was determined according to the slipstream radius r .

$$r = R \sqrt{\frac{s^2 + R^2}{s + \sqrt{s^2 + R^2}}} \quad \text{Eq. 4-10}$$

where s is the distance between the point of interest to the center of the propeller.

The width of the other segments can be determined accordingly. The MAC and AC of each segment were calculated as well. As a result, each segment will generate unique lift, drag, and moments on its AC according to its surface area, the condition of the angle of attack and airspeed. The aerodynamic coefficient will be searched in the database and is applied to the calculation of lift, drag and moment as

$$\begin{cases} L_i = \frac{1}{2} \rho v_{eff,i}^2 S_i C_{L,i} \\ D_i = \frac{1}{2} \rho v_{eff,i}^2 S_i C_{D,i} \\ M_i = \frac{1}{2} \rho v_{eff,i}^2 S_i \bar{c}_i C_{M,i} \end{cases} \quad \text{Eq. 4-11}$$

where i is the number of wing segment and \bar{c}_i is the mean aerodynamic chord. The transformation matrix then convert the freestream coordinate into body coordinate, which can be express as

$$\mathbf{R}_f^b = \begin{bmatrix} -\cos\alpha_i & 0 & -\sin\alpha_i \\ 0 & 0 & 0 \\ -\sin\alpha_i & 0 & \cos\alpha_i \end{bmatrix}. \quad \text{Eq. 4-12}$$

Thus, the aerodynamic force and moment can be expressed as

$$\mathbf{F}_{aero} = \begin{bmatrix} F_{aero,x} \\ F_{aero,y} \\ F_{aero,z} \end{bmatrix} = \sum_{i=1}^5 \mathbf{R}_f^b \begin{bmatrix} L_i \\ 0 \\ D_i \end{bmatrix} \quad \text{Eq. 4-13}$$

$$\mathbf{M}_{aero} = \sum_{i=1}^5 \begin{bmatrix} F_{aero,z}^i l_y^i \\ \frac{1}{2} C_{M,i} \rho v_{eff,i}^2 S_i \bar{c}_i + F_{aero,x}^i l_z^i \\ F_{aero,x}^i l_y^i \end{bmatrix}.$$

The aerodynamic force and moment are modeled and fed forwards to the controller as the estimated disturbance to minimize the model mismatch problem. This aerodynamic model is also used in the software-in-loop (SIL) simulation. In this study, the control surfaces were not used in the hovering flight. Because the four motors and propellers can provide much larger torque to control the vehicle compared to the control surfaces. The neglect of the control surface can significantly simplify the modelling process and the complexity of the controller design for hovering. During the transition, the aerodynamic effect becomes critical since the wing provides the lift rather than the propulsion system. As a result, the focus point of a transition control is completely different from a hovering control.

4.1.4 Allocation model

The compensated command thrust \tilde{T} and moments τ from the attitude controllers are fed into an actuator mapping block to converts them into the desired throttle of each motor and angle of the control surface. Lumped parameters of c_T and c_M were introduced to relate the rotation speed of the propeller to its thrust T_j and the reaction moment M_j as:

$$T_j = c_T \omega_j^2$$

$$M_j = c_M \omega_j^2 \quad \text{Eq. 4-14}$$

c_T and c_M were obtained from interpolating the relationship of ω^2 to T_j and ω^2 to M_j as shown in Figure 4-7, obtained from conducting a static thrust experiment as mentioned in the previous section.

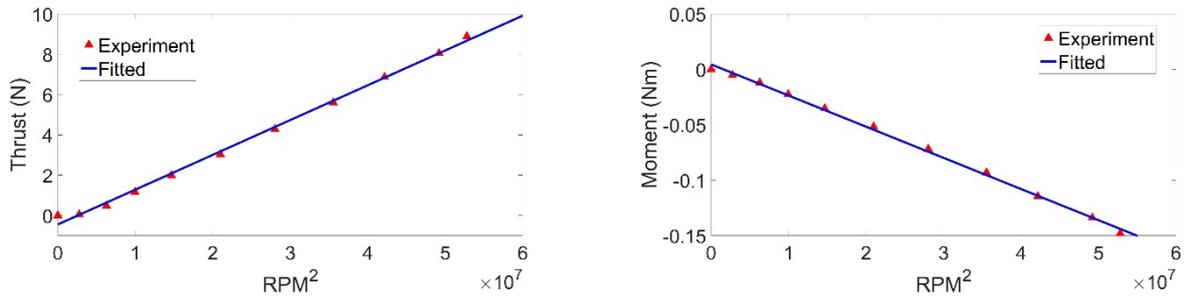


Figure 4-7 The relationship between thrust and moment to the rotation speed of a propeller

The linear relationship between the angular velocity and the desired thrust and moments can be modeled as below according to the ‘plus’ configuration of the vehicle:

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -l_y c_T & l_y c_T & 0 & 0 \\ 0 & 0 & l_x c_T & -l_x c_T \\ c_M & c_M & -c_M & -c_M \end{bmatrix}^{-1} \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad \text{Eq. 4-15}$$

If the allocation mapping receives the command signal of thrust and torques, the required rotation speed of each motor can then be calculated according to the equation above.

By the same experiment, the corresponding rotation speed to each throttle level was also discovered. The experimental result was fitted by a third-order polynomial function representing the relationship between ω^2 to throttle, which was shown in Figure 4-8. As a result, the command signal of thrust and torques were now transferred into the throttle command of each motor.

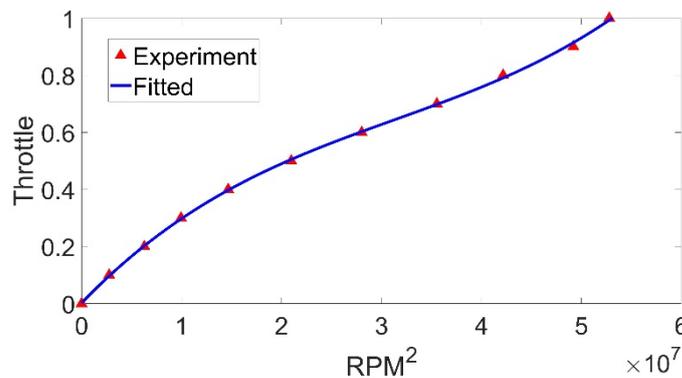


Figure 4-8 The fitted relationship between the throttle to the rotation speed of a motor.

4.1.5 Software-in-Loop Simulation

The simulation environment was built in Simulink shown in Figure 4-9. A propulsion model, a gravity model, an estimated disturbance model and an aerodynamic model were developed. Total forces and moments will be summed before acting on the vehicle, which can be considered as 6 degrees of freedom rigid body. As the control commands of throttle arrived at the propulsion system model, the value of thrust on motor 1 and 2 can be calculated and send to the aerodynamic model. With the command on the control surface angle as well as the airspeed from the environment model, the aerodynamic model can predict the aerodynamic force and moment of the vehicle.

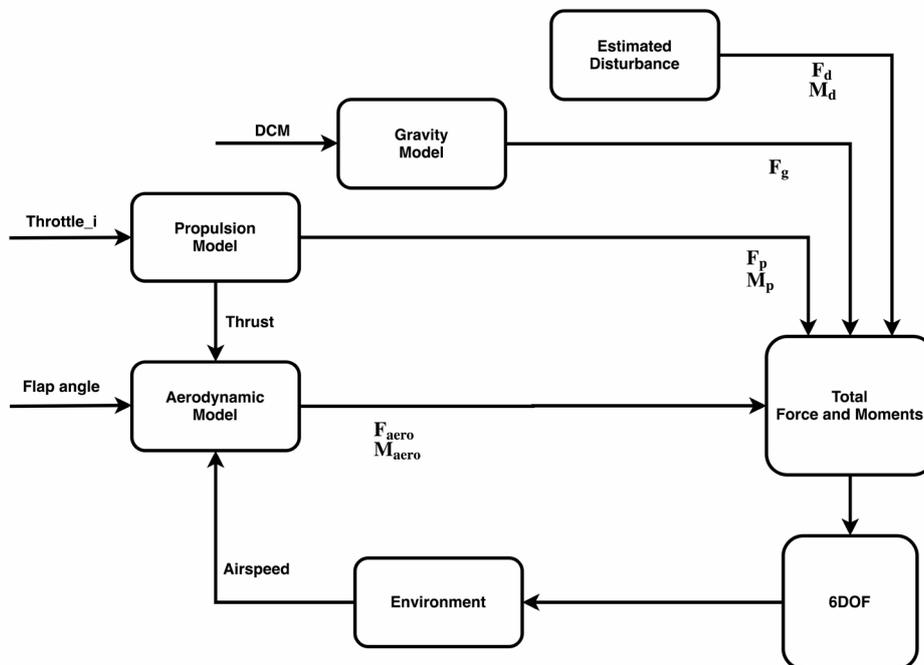


Figure 4-9 Schematic diagram of the simulation environment

4.2 Linear MPC Controllers

This chapter presents the development of linear MPC controllers for the tail-sitter vehicle. A cascade structure of position control and attitude control is illustrated first, followed by the setup of a plant model. The objective function is then decided as well as the parameters. The linear MPC controller is then tested by conducting indoor flight tests of trajectory following and disturbance rejection comparing to a traditional PID controller.

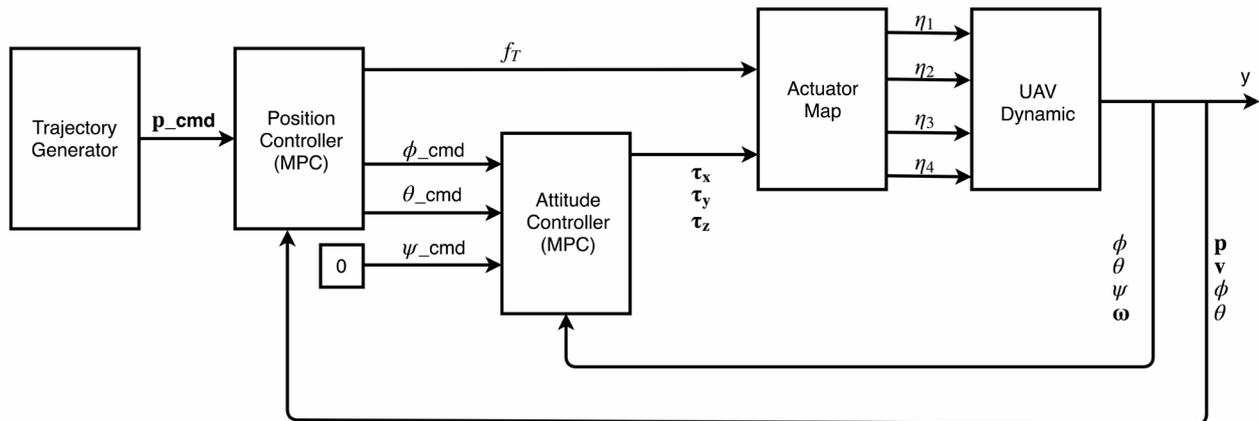


Figure 4-10 Block diagram of Two-loop Linear MPC controller

Figure 4-10 shows the block diagram of the cascade control system, in which the ‘Actuator Map’ is for converting the desired torque and moment into the required rotational speeds η of the four motors. The psi command was set as zero at all times since the yaw angle has been eliminated by the small-disturbance theory and it did not affect the position. The cascade structure has the advantage of reducing the onboard computation load. Two controllers each have fewer states is more practical for onboard flight control mission because the computation effort increases dramatically with the number of states increases of MPC controller. This strategy also allows a faster updating rate in the attitude control than the position control. Thus, there is less possible that the vehicle would crash due to the loss of attitude. With this feature, this cascade structure can be considered as an improvement to higher reliability and robustness.

4.2.1 Plant Model

The nonlinear dynamic model of the vehicle can be linearized based on the small-disturbance theory, which assumes the motion of the vehicle consists of a small deviation about a steady condition and approximates the sine function with its argument and the cosine function with unity. The product

of disturbances are also neglected. After linearization about the hovering point, the dynamic and the kinematic model can be expressed as:

$$\begin{aligned}
 \dot{x} &= u \\
 \dot{y} &= v \\
 \dot{z} &= w \\
 \dot{u} &= -g\theta \\
 \dot{v} &= g\phi \\
 \dot{w} &= g - f_T/m \\
 \dot{\phi} &= p \\
 \dot{\theta} &= q \\
 \dot{\psi} &= r \\
 \dot{p} &= \tau_x/I_x \\
 \dot{q} &= \tau_y/I_y \\
 \dot{r} &= \tau_z/I_z
 \end{aligned}
 \tag{Eq. 4-16}$$

The above linear model can be written into state-space form as

$$\begin{aligned}
 \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\
 \mathbf{y} &= \mathbf{C}\mathbf{x}.
 \end{aligned}
 \tag{Eq. 4-17}$$

The control system has the outer position control with state of

$$\mathbf{x}_p = [x \ y \ z \ u \ v \ w \ \phi \ \theta]^T$$

and control input of

$$\mathbf{u}_p = [\phi_{cmd} \ \theta_{cmd} \ f_T]^T,$$

where ϕ and θ in \mathbf{x}_p were expressed by a set of first-order transfer function [50] as

$$\begin{aligned}
 \dot{\phi} &= \frac{1}{\tau_\phi} (K_\phi \phi_{cmd} - \phi) \\
 \dot{\theta} &= \frac{1}{\tau_\theta} (K_\theta \theta_{cmd} - \theta)
 \end{aligned}
 \tag{Eq. 4-18}$$

which K_ϕ , K_θ and, τ_ϕ , τ_θ are gains and time constants of roll and pitch angle respectively. Where \mathbf{C} is an 8 by 8 identity matrix and $\mathbf{A} \in [8 \times 8]$ and $\mathbf{B} \in [8 \times 3]$ were defined as

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\tau_\phi^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\tau_\theta^{-1} \end{bmatrix}$$

Eq. 4-19

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1/m \\ K_\phi/\tau_\phi & 0 & 0 \\ 0 & K_\theta/\tau_\theta & 0 \end{bmatrix}$$

The inner attitude controller has state of

$$\mathbf{x}_a = [\phi \ \theta \ \psi \ p \ q \ r]^T$$

and control input of

$$\mathbf{u}_a = [\tau_x \ \tau_y \ \tau_z]^T$$

where \mathbf{C} is a 6 by 6 Identity matrix, $\mathbf{A} \in [6 \times 6]$ and $\mathbf{B} \in [6 \times 3]$ were defined as

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Eq. 4-20

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ I_x^{-1} & 0 & 0 \\ 0 & I_y^{-1} & 0 \\ 0 & 0 & I_z^{-1} \end{bmatrix}$$

4.2.2 Optimization and Stability Problem

The objective function applied to this controller was defined as the same as Eq. 3-14, where the weight matrix \mathbf{R}_u was given zero since there was no reference signal on input to be followed. The other parameters had to be adjusted according to their meaning to achieve a reliable and robust performance of the MPC controller.

The sampling time (or duration of each control interval) T_s is set as 0.04 seconds in the simulation, which means the control algorithm runs at 25Hz. As T_s decreases, rejection of unknown disturbance usually improves. However, as T_s becomes small, the computational requirement increases dramatically. Thus, the optimal choice is a balance of performance and computational effort [81].

The prediction horizon P should not be too large since the controller memory requirement increases as well as the size of the Quadric Programming increase. However, P cannot be too small because the constraint violations would be unforeseen. Let T as the desire prediction duration and $T \approx T_s \times P$. Usually, P must vary inversely with T_s . For example, in the simulation, P was set as 50, such that the prediction duration $T \approx 0.04 \times 50 = 2$ second ahead.

The control horizon H must fall between 1 and P . Regardless of the choice of H , when the controller operates, only the first optimized control move is used, and any others are discarded. Small H means fewer variables to compute in the QP solution at each control interval, which promotes faster computations but weaker optimization and vice versa.

Non-negative scale vectors \mathbf{S} are divided by each plant input and output to generate dimensionless signals. If any signals have significantly larger or smaller magnitudes than the others, the defining scale factor is especially important. Unwanted overshoot can be minimized by reducing the corresponding scale factor.

Adjusting the weight of each variable is critical to the performance of the controller. For output variable (OV) weight and MV weight, higher weight means higher priority with rough guidelines of [82]:

0.05 — Low priority: Large tracking error acceptable

0.2 — Below-average priority

1 — Average priority – the default.

5 — Above average priority

20 — High priority: Small tracking error desired

In most of the case, increase the OV weight until the corresponding signal oscillate. Then adjust the MV rate weight, which penalizes large MV changes in the optimization cost function [83]. Small MV rate weight means fast respond in control input. For example, simultaneously reduces the MV rate weight and increases the corresponding OV weight will produce a more robust controller. Since the controller will limit the move and put a higher priority on feedback at each control interval.

4.2.3 Compensation on Thrust and Angle Command

The acceleration in the equation of motion on z-direction includes a weight term and a thrust term. However, this weight is simply added onto the thrust command f_T since the trim operation condition around the hovering point does not consider the weight. As a result, to balance the EOM on z-direction when there is a non-zero attitude of the vehicle, the weight mg has to be compensated by dividing by the product of $\cos\phi$ and $\cos\theta$, such that the current attitude was accounted. Then assume that the weight mg has been transferred into the body frame, the same as the thrust. The final command \tilde{T} is shown as below and will be feed to the allocation mapping.

$$\tilde{T} = \left(\frac{mg}{\cos\phi\cos\theta} - f_T \right) \quad \text{Eq. 4-21}$$

The command on angles also needs to be compensated before sending it to the attitude controller. When there is an angle in the attitude, the final thrust command \tilde{T} should be larger than f_T to keep the altitude. Meanwhile, the command angle should decrease to prevent overshoot in x or y position. As a result, a simple relationship is developed as

$$\begin{aligned} \tilde{\phi} &= \frac{mg\phi_c}{\tilde{T}} \\ \tilde{\theta} &= \frac{mg\theta_c}{\tilde{T}} \end{aligned} \quad \text{Eq. 4-22}$$

where ϕ_c and θ_c and the commanded value from the control input of the position controller. $\tilde{\phi}$ and $\tilde{\theta}$ will be the final command in the roll and pitch angle that will be sent to the attitude control.

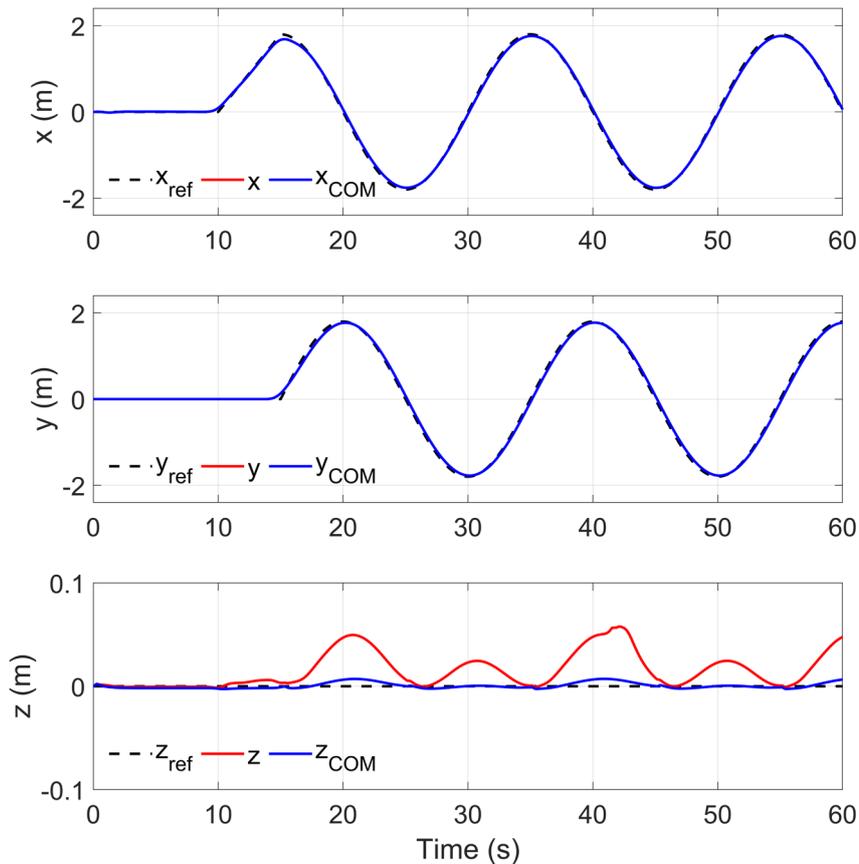


Figure 4-11 Simulation result of LMPC controller following a circular path with (blue) and without (red) thrust compensation

Figure 4-11 shows a simulation result of following a circular path by two-loop linear MPC controllers aiming to keep at the same high. It shows that without the help of a thrust compensation (red line), the high dropped 0.05m whenever the speed in the x-direction is maximum, which the pitch angle is at maximum. The thrust compensation was added to the LMPC controller (blue line). Larger compensation is set on the pitch angle since it has a much weaker effect in pitch than roll, because of the wings. The commanded pitch angle is much larger than the roll angle even the command distance on the x-position is the same as the y-position. Finally, a drop in high of only 0.007m is achieved in the blue line result, which is nearly ten times smaller than the red line result.

4.2.4 Flight Tests and Results

The flight tests were conducted to further evaluate the hover flight stability of the tail-sitter vehicle in windy conditions in the aviation laboratory of The Hong Kong Polytechnic University with the vehicle's position captures by a motion capture system. The developed LMPC controller was

compared with a PID controller. The artificial wind field is created to better control the experimental condition. The details of the experiment are presented followed by the discussion of the results. The worst-case flight tests of disturbance rejection to hold the position of the vehicle while the vehicle's wing is directly facing the wind are performed.

The flight control unit (FCU) consists of commercial autopilot hardware, Pixhawk, and open-source PX4 firmware. An Odroid XU4 companion computer equipped with Samsung Exynos5422 Cortex™-A15 (2Ghz) and Cortex™-A7 Octa-core CPUs is installed for real-time MPC computation to guarantee the computing efficiency of quadratic programming in the flight tests. A 2.4-GHz WiFi module, serving as the telemetry system, is connected to the companion computer to transmit data between the vehicle and the ground control station.

The low-level control was handled by the basic flight control unit of Pixhawk, including the attitude PID control as well as the allocation mapping. The high-level position control was handled by the Odroid XU4 companion computer due to its much stronger computation capability. The designed MPC position controller was generated into C++ code by Matlab Simulink Coder and installed onto the Odroid as a standard robot operating system (ROS) node. MAVROS is an extendable communication node of ROS which can transfer the ROS message into MAVLink protocol, which is the only protocol to communicate with Pixhawk. MAVLink protocol was used to exchange vehicle real-time states as well as the command signal between the control unit between the Odroid and the Pixhawk as shown in Figure 4-12. The others were communicated under the User Datagram Protocol (UDP). The experiment was conducted indoors using a VICON motion capture system to obtain real-time positional information for the vehicle, shown in Figure 4-13.

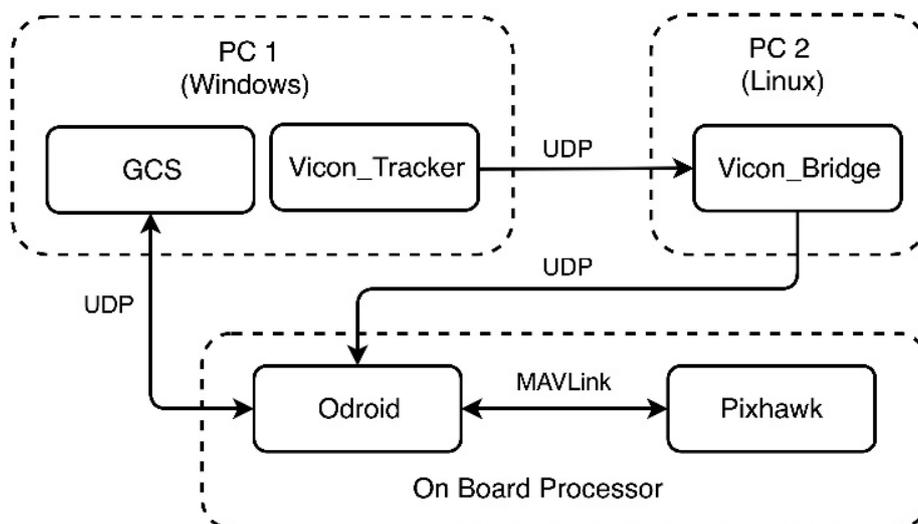


Figure 4-12 Communication network setup

VICON™ motion capture system can provide position measurement in millimeter level in the indoor environment, with updating rate up to 300Hz. The result of this system is usually used as benchmark. Several small plastic balls with infrared reflective layer have to be installed on the UAV. The VICON™ cameras emits infrared and receive reflective signal. Each small plastic ball can then be precisely located in the certain area. By combining several balls as an rigid object, the geometric center of them can be defied as the UAV's centre. By tracking all of the relevent balls and calculates the geometric center, the acceleration, velocity, position and attitude of the UAV can be tracked.

With the help of this system, multiple functions of UAV can be tested in the indoor environment first before moving to the outdoor fields.

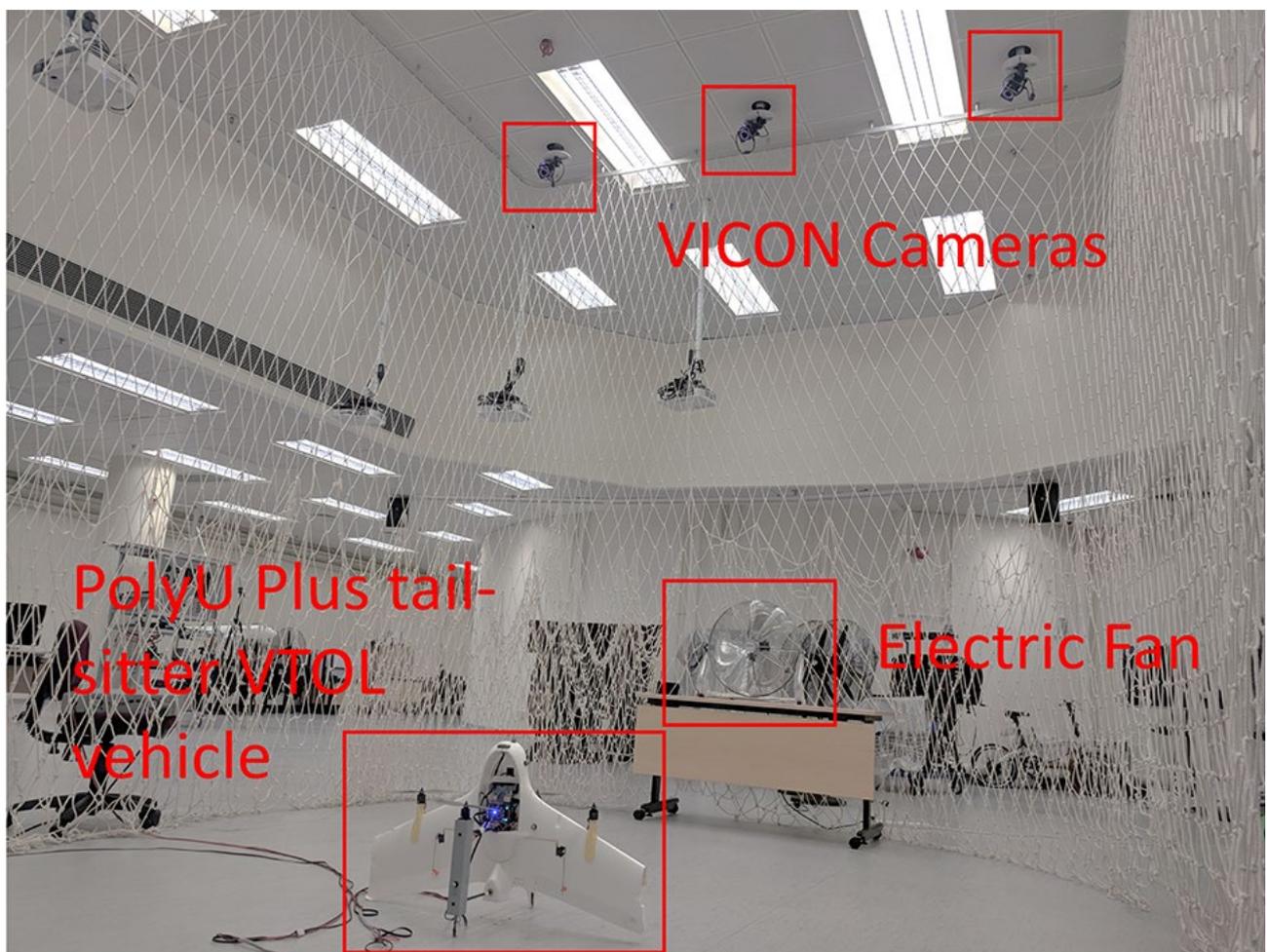


Figure 4-13 Experimental setup

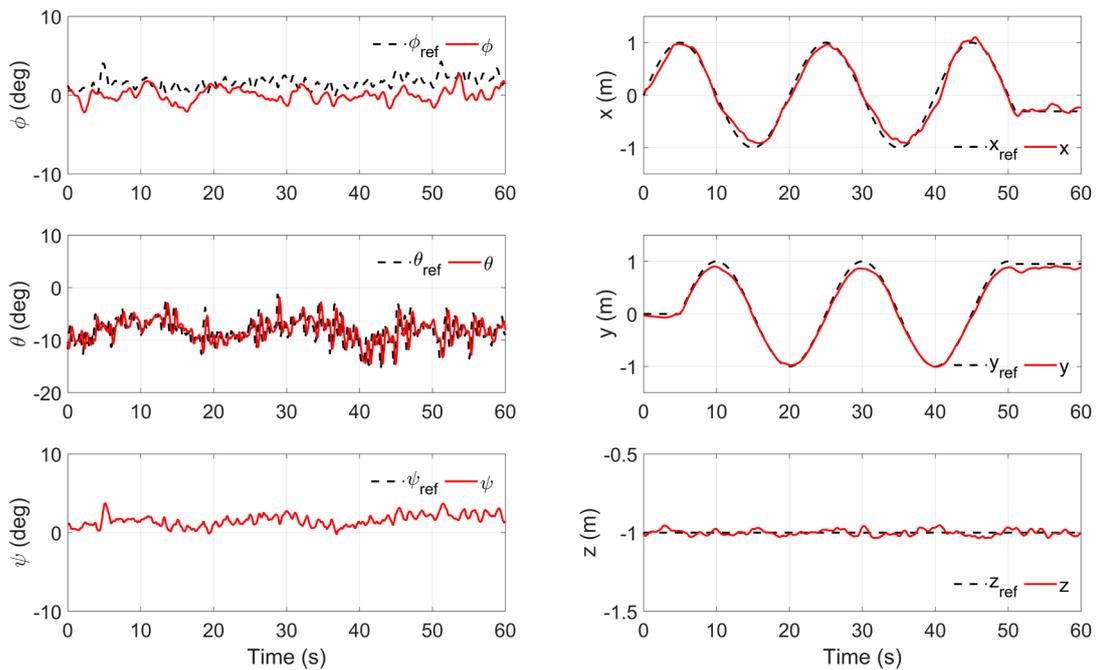


Figure 4-14 Experiment result of LMPC position controller following a circular path

Figure 4-14 shows the experimental result of the path following a circular trajectory by the linear MPC position controller. In the figure, there was no significant drop at the high even though the pitch angle has reached a maximum value of -15° . Model mismatch was encountered since the small-disturbance theory will not be satisfied at angular movement far away from zero degrees. However, with the help of the thrust and angle compensation, the linearized MPC control can work properly in a real situation.

One of the goals was to improve the hovering stability of the tail-sitter vehicle under windy conditions. As a result, artificial wind condition was generated to ensure a similar outdoor environment. Figure 4-15 shows the setup of the wind speed measurement, where the fan was approximately 3 m away from the position holding point. A Testo 480 digital meter was placed at the position holding point to measure the unsteady wind speed at a sampling frequency of 1 Hz. The result for a 3-minute measurement is demonstrated in Figure 4-16 and the mean wind speed is 1.88 m/s with a significant fluctuation of 0.58 m/s. This unsteady wind condition simulates a gusty winds outdoor environment. The wind direction and the measured mean value were fed into the estimated disturbance model.

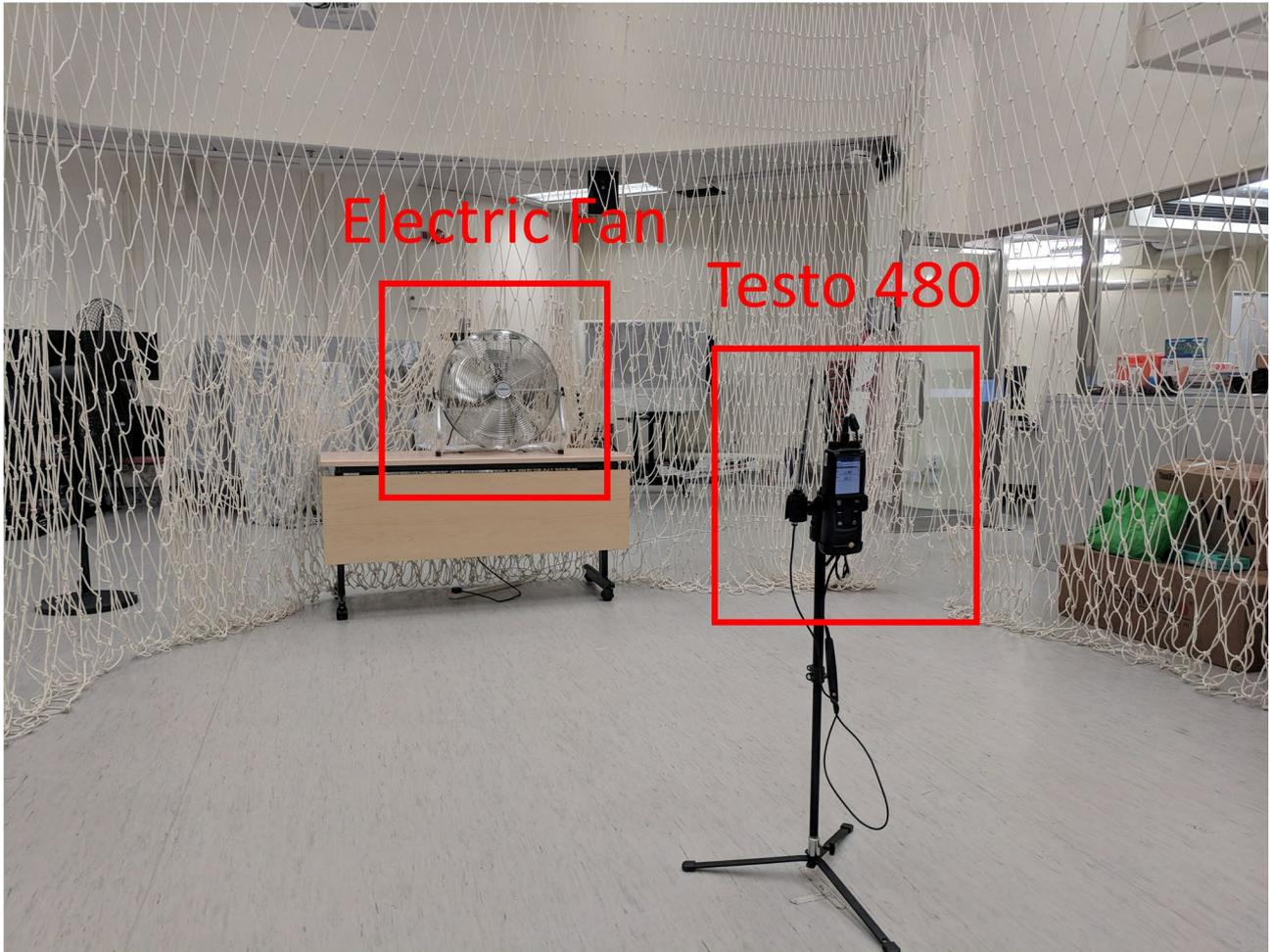


Figure 4-15 Wind speed measurement

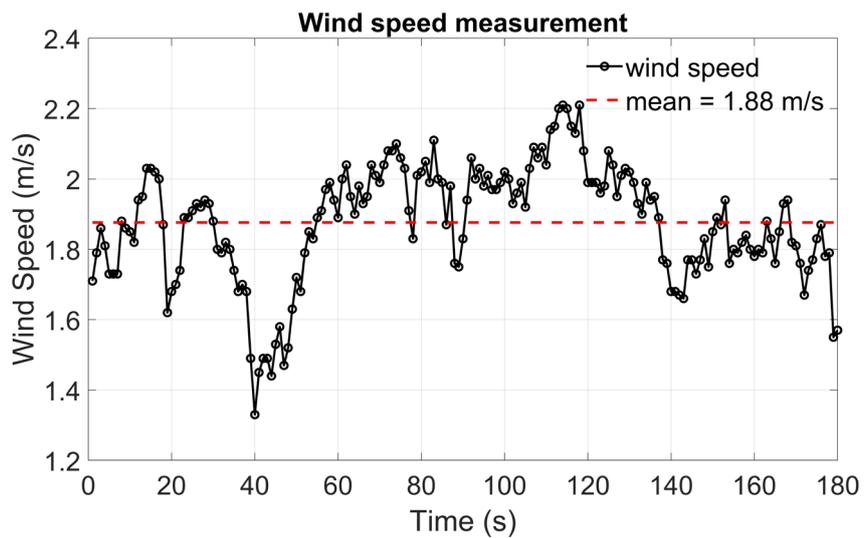


Figure 4-16 Measurement of unsteady wind speed of the artificial wind field

The designed mission was to hold the vehicle position at $[0,0,-1]m$ in the x -, y - and z -directions with wind disturbance directly facing the vehicle wing. Zero in the x - and y -directions is the origin and

-1 in the z -direction means that the vehicle is set to hover 1 m above the ground. A comparison of the position control performance in the indoor hovering tests, using a traditional PID controller, a linear MPC controller, is presented in this section. The wind speed in the environment model was set to $1.88 \pm 0.3 \text{ m/s}$ according to the experimental results. Horizontal wind in the negative x -direction was generated from the 30th second to the 50th second.

Case 1 (PID Control): A built-in PID controller in the open-source PX4 firmware has its parameters tuned via the Ziegler-Nichols method. The indoor flight test result under wind disturbance, using the tuned PID controller on the PolyU Plus tail-sitter VTOL vehicle, is shown in Figure 4-17. When the wind hits the vehicle (the shadowed area), the vehicle is blown more than 0.5 m away and gradually flies back to the setpoint in the x -direction. The integration term of the PID controller is known to affect the system when the error is large enough. Without this integration term, the vehicle would have difficulty eliminating the steady-state error, as a consistent aerodynamic force acts on the wing. Apart from the x -direction, there is a high-frequency oscillation in the roll angle (ϕ) and a large unfavourable movement of up to 30° in the yaw angle (ψ), although the performance in the y - and z -directions is acceptable. Notably, the abrupt changes in the yaw angle (ψ) and z -position during the last few seconds are caused by the manual landing process in the experiment. The corresponding results have a root mean square error of [0.2214, 0.0889, 0.1451] in the x -, y - and z -directions.

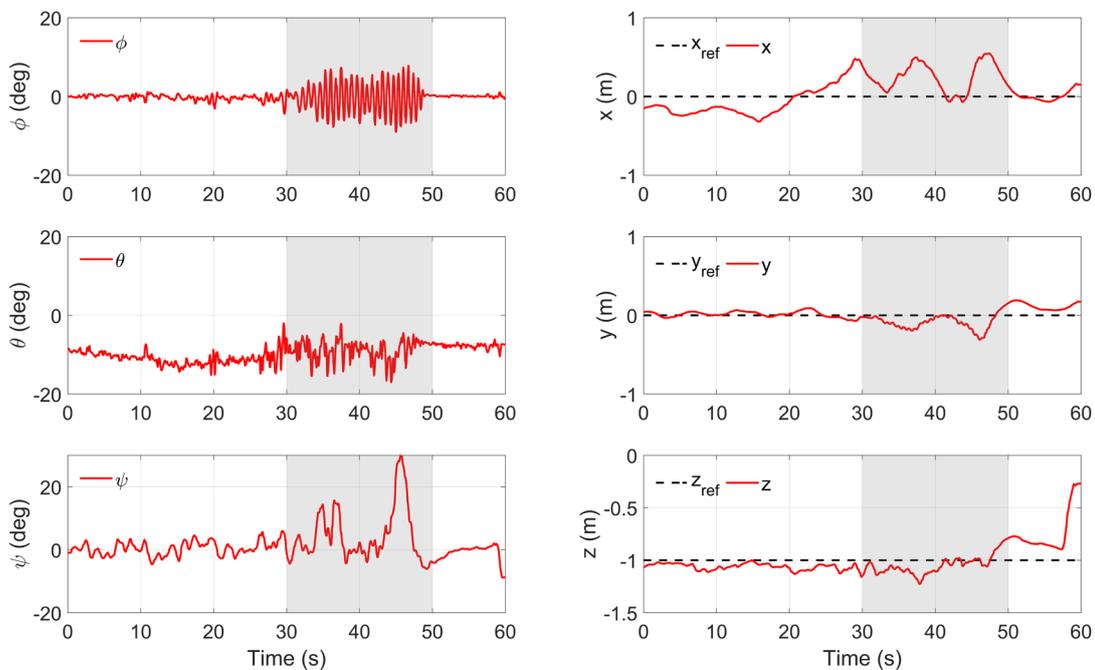


Figure 4-17 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using a PID controller.

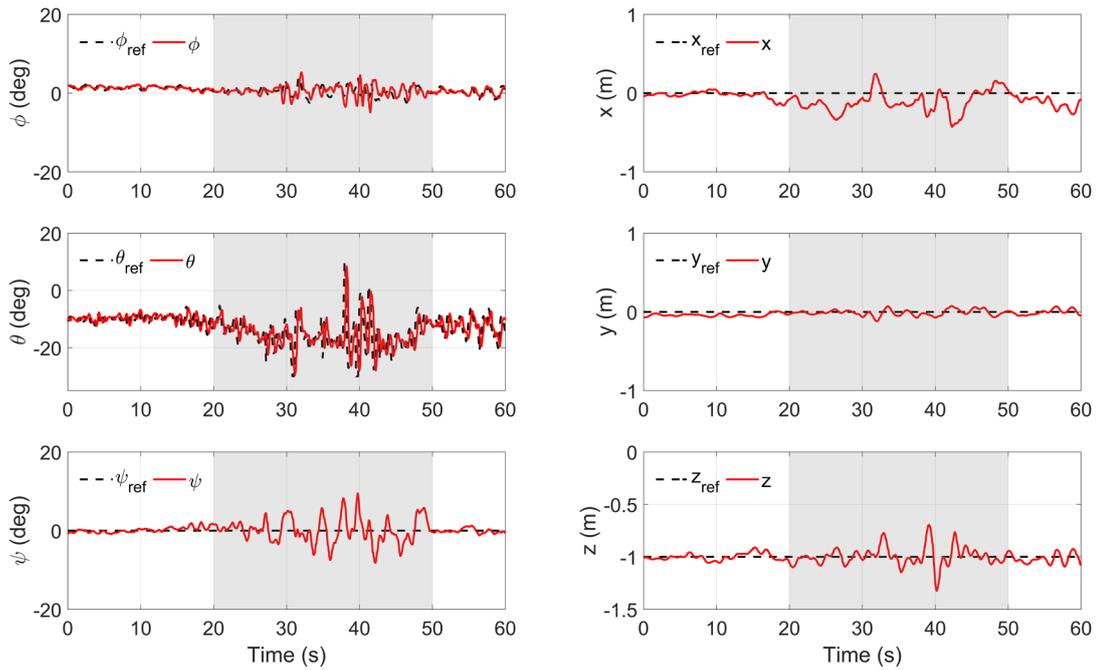


Figure 4-18 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using a linear MPC controller.

Case 2 (Linear MPC Control): The flight performance using a linear MPC position controller is presented in Figure 4-18. Its performance in the x -direction is better than that of the previous PID controller, as it has a smaller variation of approximately 0.4 m in x -direction and less peak movement. However, it does not control as effectively in the z -direction. There is a greater decrease in height and a more obvious oscillation in the z -direction. The attitude command signals also fluctuate, especially for the pitch (θ) command, fluctuating from $+10^\circ$ to -30° at approximately the 38^{th} second. As the linear MPC controller has its trim condition set around the hover point, which is 0° for all angles, any angular movement that is larger than 10° violates the predefined linear model and causes a model mismatch. However, the tendency to pitch down and then pitch up during windy conditions can be observed. The root mean square error values of $[0.1615, 0.0478, 0.0668]$ are smaller than those for the previous flight.

4.3 Successive Linearization MPC Controllers

This chapter will present the development of a successive linearization MPC (SLMPC) controller for the tail-sitter vehicle. The cascade structure and the estimated disturbance model will be illustrated first, followed by the setup of the plant model. The objective function is then developed with improvements by introducing time-varying weight and velocity as reference. The SLMPC controller is then tested by conducting indoor flight tests of trajectory following and disturbance rejection in comparison to the linear MPC controller in the previous chapter.

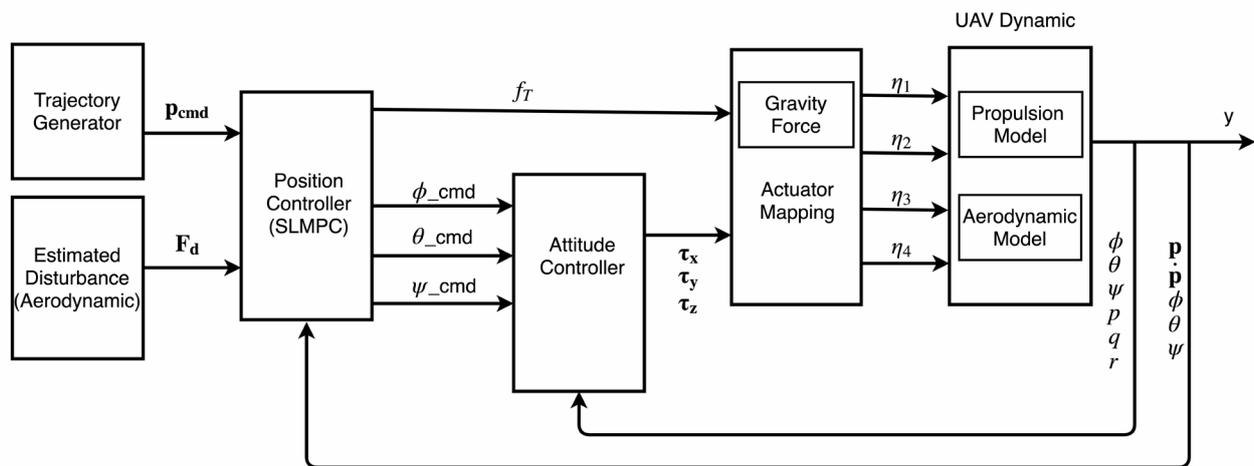


Figure 4-19 Block diagram of the cascaded control structure of the SLMPC controller used in the PolyU Plus tail-sitter vehicle.

Figure 4-19 shows the cascaded control structure, where all states are assumed to be measurable by the onboard estimator. The cascaded structure has the advantage of reducing the onboard computational load. Both controllers have relatively few states, which is more practical for onboard flight control missions, as the computational effort increases dramatically as the number of MPC controller states increases. This strategy allows for a faster updating rate in the attitude control than the position control, lowering the possibility of crashing the UAV due to attitude loss. As a result, reliability and robustness are improved.

4.3.1 Plant Model

The dynamic model of the vehicle can also be described as

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ m\dot{\mathbf{v}} &= \mathbf{R}_b^I \mathbf{F}_b \end{aligned} \tag{Eq. 4-23}$$

$$\boldsymbol{\Omega} = \mathbf{T}_b^I \boldsymbol{\omega}$$

$$\mathbf{I} \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I} \boldsymbol{\omega}) = \mathbf{M}_b$$

where $\mathbf{p} = [x \ y \ z]^T \in \mathbb{R}^3$ and $\mathbf{v} = [\dot{x} \ \dot{y} \ \dot{z}]^T \in \mathbb{R}^3$ are the position and velocity in inertia frame, $\boldsymbol{\Omega} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \in \mathbb{R}^3$ is the time derivative of Euler angles (roll, pitch and yaw), and $\boldsymbol{\omega} = [p \ q \ r]^T \in \mathbb{R}^3$ is the angular velocity in the body frame.

The full expression can be expressed as

$$\ddot{x} = -\frac{f_T}{m} [\cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi]$$

$$\ddot{y} = -\frac{f_T}{m} [\cos\phi \sin\psi \sin\theta - \cos\psi \sin\phi]$$

$$\ddot{z} = g - \frac{f_T}{m} [\cos\phi \cos\theta]$$

$$\dot{\phi} = p + r(\cos\phi \tan\theta) + q(\sin\phi \tan\theta)$$

$$\dot{\theta} = q \cos\phi - r \sin\phi$$

$$\dot{\psi} = r \frac{\cos\phi}{\cos\theta} + q \frac{\sin\phi}{\cos\theta}$$

$$\dot{p} = \frac{I_y - I_z}{I_x} r q + \frac{\tau_x}{I_x}$$

$$\dot{q} = \frac{I_z - I_x}{I_y} p r + \frac{\tau_y}{I_y}$$

$$\dot{r} = \frac{I_x - I_y}{I_z} p q + \frac{\tau_z}{I_z}$$

Eq. 4-24

The expanded dynamic model can be expressed in the state-space model by setting the outer position adaptive controller has its state defined as

$$\mathbf{x} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T$$

and control input as

$$\mathbf{u} = [f_T \ \sin\phi \ \sin\theta \ \sin\psi]^T$$

where the input signal of $\sin\phi$, $\sin\theta$ and $\sin\psi$ will be fed into an \arcsin function and transfer into ϕ_{cmd} , θ_{cmd} and ψ_{cmd} before sent to the attitude controller.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{f_T \sin \psi}{2m} & \frac{f_T \cos \phi \cos \psi}{m} & \frac{f_T \sin \phi}{2m} \\ 0 & -\frac{f_T \cos \psi}{m} & \frac{f_T \cos \phi \sin \psi}{2m} & \frac{f_T \cos \phi \sin \theta}{2m} \\ \frac{\cos \phi \cos \theta}{m} & 0 & 0 & 0 \end{bmatrix} \quad \text{Eq. 4-25}$$

Non-linearity is adopted by successively discretizing the model at every time step, as the SLMPC can consistently update parameters and nominal values at every time step. While the operation condition varies, the nominal value can be updated, and a much larger control envelope can be covered without the risk of model mismatch. At each time step, the generated command thrust (f_T) adjusts the altitude of the vehicle. Then f_T and the latest measured attitude angles (ϕ , θ and ψ) are used to update the transfer matrix (\mathbf{B}_u) and calculate a new control input. Although, the yaw angle ψ does not affect the position. However, keeping it in the plant model could allow the objective function to calculate an even smaller cost in comparison to ignoring it. It helps improving the performance in windy conditions as the uneven wind can usually cause the vehicle yaw back and forth.

The inner attitude controller has its state defined as

$$\mathbf{x}_a = [\phi \ \theta \ \psi \ p \ q \ r]^T$$

and control input defined as

$$\mathbf{u}_a = [\tau_x \ \tau_y \ \tau_z]^T$$

where \mathbf{C} is a 6 by 6 identity matrix, $\mathbf{A} \in [6 \times 6]$ and $\mathbf{B} \in [6 \times 3]$ were show in Eq. 4-26.

Same as the outer loop, at every time instant, the least measured Euler angle ϕ , θ and ψ and their derivative will be fed into the \mathbf{A} matrix and calculate new control move. After the state-space models have been updated at every time instant, discretization will be conducted before implementing it to the prediction.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & 0 & 0 & 0 & \cos\phi & -\sin\phi \\ 0 & 0 & 0 & 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \\ 0 & 0 & 0 & 0 & \frac{I_y - I_z}{2I_x} r & \frac{I_y - I_z}{2I_x} q \\ 0 & 0 & 0 & \frac{I_z - I_x}{2I_y} r & 0 & \frac{I_z - I_x}{2I_y} p \\ 0 & 0 & 0 & \frac{I_x - I_y}{2I_z} q & \frac{I_x - I_y}{2I_z} p & 0 \end{bmatrix} \quad \text{Eq. 4-26}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/I_x & 0 & 0 \\ 0 & 1/I_y & 0 \\ 0 & 0 & 1/I_z \end{bmatrix}$$

4.3.2 Discretization

For the SLMPC controller, the state-space model is updated at every sampling period and as a result, it is required to be designed in a discrete model. The discretization of a continuous system of Eq. 3-13 can be transformed into a discrete system at the sampling period T_s as:

$$\begin{aligned} \mathbf{x}((k+1)T_s) &= \mathbf{G}(T_s)\mathbf{x}(kT_s) + \mathbf{H}(T_s)\mathbf{u}_t(kT_s) \\ \mathbf{y}(kT_s) &= \mathbf{C}(T_s)\mathbf{x}(kT_s) \end{aligned} \quad \text{Eq. 4-27}$$

Since the discretization is for the state equation describing the dynamic characteristics of the system, the output equation is a static algebraic equation that should remain unchanged after discretization. As a result

$$\mathbf{C}(T_s) = \mathbf{C} \quad \text{Eq. 4-28}$$

The discretization process uses the solution formula of the state equation to ensure that the state is discretized at the time of sampling. It also ensures the continuous state equation and the discretized state equation have the same solution. The solution of Eq. 4-27 for a continuous system can be expressed as:

$$\mathbf{x}(t) = \Phi(t - t_0)\mathbf{x}(t_0) + \int_{t_0}^t \Phi(t - \tau) \mathbf{B}\mathbf{u}_t(\tau) d\tau \quad \text{Eq. 4-29}$$

Now consider only the state response between the sampling instants $t_0 = kT_s$ and $t = (k + 1)T_s$, we can have:

$$\mathbf{x}((k + 1)T_s) = \mathbf{\Phi}(T_s)\mathbf{x}(kT_s) + \int_{kT_s}^{(k+1)T_s} \mathbf{\Phi}[(k + 1)T_s - \tau] \mathbf{B}\mathbf{u}_t(\tau) d\tau \quad \text{Eq. 4-30}$$

Considering that $u(t)$ remains constant during the sampling period, such that

$$\mathbf{x}((k + 1)T_s) = \mathbf{\Phi}(T_s)\mathbf{x}(kT_s) + \int_{kT_s}^{(k+1)T_s} \mathbf{\Phi}[(k + 1)T_s - \tau] d\tau \mathbf{B}\mathbf{u}_t(kT_s) \quad \text{Eq. 4-31}$$

Let $t = (k + 1)T_s - \tau$, then

$$\mathbf{x}((k + 1)T_s) = \mathbf{\Phi}(T_s)\mathbf{x}(kT_s) + \int_0^{T_s} \mathbf{\Phi}(t) dt \mathbf{B}\mathbf{u}_t(kT_s) \quad \text{Eq. 4-32}$$

As a result, compare to $\mathbf{x}((k + 1)T_s) = \mathbf{G}(T_s)\mathbf{x}(kT_s) + \mathbf{H}(T_s)\mathbf{u}_t(kT_s)$, we have

$$\begin{cases} \mathbf{G}(T_s) = \mathbf{\Phi}(T_s) = e^{AT_s} \\ \mathbf{H}(T_s) = \int_0^{T_s} \mathbf{\Phi}(t) dt \mathbf{B} = \int_0^{T_s} e^{At_s} dt \mathbf{B} \end{cases} \quad \text{Eq. 4-33}$$

which $\mathbf{G}(T_s)$ and $\mathbf{H}(T_s)$ can be subsided back to Eq. 4-27 and the discretization process is finished.

The plant model for SLMPC can be written in terms of deviation from its nominal condition

$$\begin{aligned} \mathbf{x}(k + 1) &= \bar{\mathbf{x}} + \mathbf{A}(\mathbf{x}(k) - \bar{\mathbf{x}}) + \mathbf{B}(\mathbf{u}_t(k) - \bar{\mathbf{u}}_t) + \bar{\Delta\mathbf{x}} \\ \mathbf{y}(k) &= \bar{\mathbf{y}} + \mathbf{C}(\mathbf{x}(k) - \bar{\mathbf{x}}) + \mathbf{D}(\mathbf{u}_t(k) - \bar{\mathbf{u}}_t) \end{aligned} \quad \text{Eq. 4-34}$$

where $\bar{\mathbf{x}}$ is the nominal states, $\bar{\Delta\mathbf{x}}$ is the nominal state increment, \mathbf{u}_t is the combined plant input variable, comprising \mathbf{u} , \mathbf{v} and \mathbf{d} . $\bar{\mathbf{u}}_t$ is the nominal input, $\bar{\mathbf{y}}$ is the nominal output. The ability of consistent updating of parameters and nominal values allows the accomplishment of non-linearity by successively discretizing the model at every time step. At every timestep, the nominal values of the SLMPC are updated. It can be considered as the trim point value of a regular MPC control are changes, the control law at every timestep is aiming to fit the current trim point, rather than a fixed trim point. With the varying nominal values, the operation condition can deviate and achieved in a much larger control envelope without causing a model mismatch problem.

4.3.3 Allocation Mapping with Control Surface

According to the vehicle's configuration, there are a pair of control surface on the wing. Like the motors, the control surface pair is also an actuator. A change in its angle can create a significant effect on the aerodynamic moment on the wing. The lift drag and moment effect will differ according to the change of the angle and the airspeed across the wing. Since the large wingspan will be strongly affected by wind disturbance. The larger the control moment on pitch movement the better rejection it can be. Thus, the use of the control surface was focused on helping the pitch movement rather than others. A linear relationship was set, bridging the commanded pitch moment from the attitude controller to the control surface angle. Its value was secured by an upper and lower limit. As a result, apart from the moment created by the difference in the rotation speed of motor 3 and 4, the control surface can also provide a moment on pitch movement and help improve the corresponding performance.

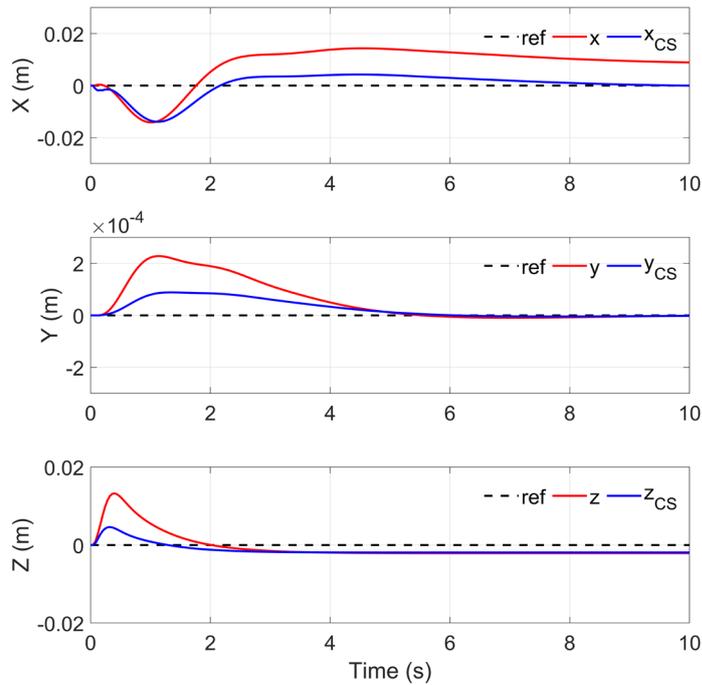


Figure 4-20 Simulation result of SLMPC Hover with (blue) and without (red) control surface

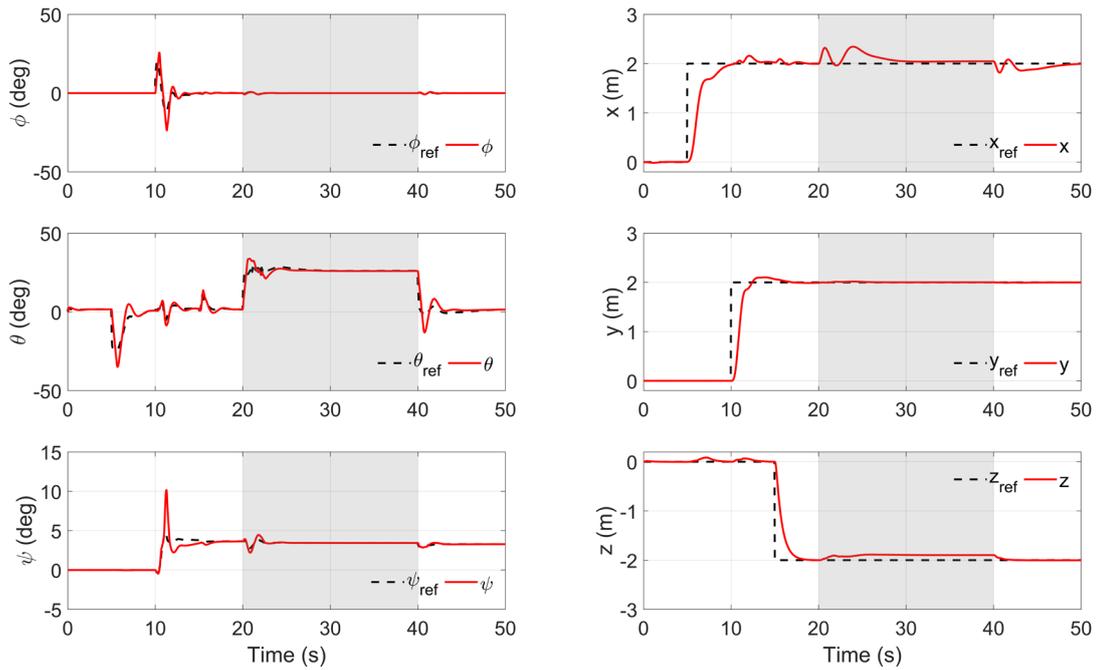


Figure 4-21 Simulation result of step move and position hold by SLMPC under 2m/s wind with control surface

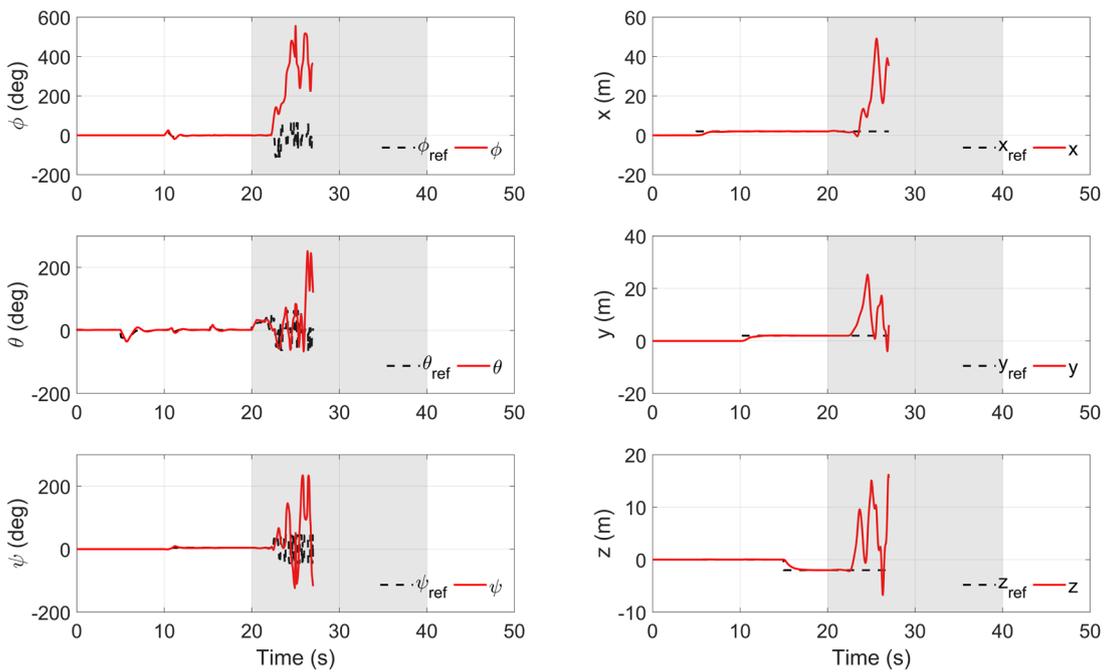


Figure 4-22 Simulation result of failure to hold the position of SLMPC under 2m/s wind without control surfaces

At the hovering stage, the control surface can help reduce the steady-state error as shown in the x-position in Figure 4-20. This offset was minimized in a trim-point of 6-degree on the control surface,

according to the red line and blue line. Although this steady-state error can also be eliminated by augmenting an error integrator in the controller, the control surface has strengthened the ability of wind resistance.

Its contribution was shown in a case of following step movement and position hold under 2m/s wind from x-direction between 20 seconds and 40 seconds with the usage of control surface was shown in Figure 4-21. As shown in the figure, the pitch angle has been commanded to its maximum value of 30 degrees and the x-position has a 0.4m deviate. The high of z-direction has slightly dropped a little bit during the windy environment. On the other side, without the help of the control surface, a maximum of 30 degrees pitch was not enough for the vehicle to resist the wind in the previous case in Figure 4-21. The vehicle has significantly lost control after 23 seconds, shown in Figure 4-22.

4.3.4 Effectiveness of Estimated Disturbance Model

Apart from being a part of the simulation model, the aerodynamic model was also used in the MPC controller design as an estimated disturbance. With the knowledge of the wind speed and direction, the aerodynamic force and moment can be estimated and being fed to a disturbance term \mathbf{v} of Eq. 3-1. For the position controller while the definition of \mathbf{x} and \mathbf{u} were keep unchanged, \mathbf{F}_d was set as \mathbf{v} and the corresponding \mathbf{B}_v can be derived as below:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \mathbf{A}\mathbf{x} + \mathbf{B}_u\mathbf{u} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/m & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & 1/m \end{bmatrix} \begin{bmatrix} F_{d,x} \\ F_{d,y} \\ F_{d,z} \end{bmatrix} \quad \text{Eq. 4-35}$$

The same method applied to the attitude controller.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{A}\mathbf{x} + \mathbf{B}_u\mathbf{u} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/I_x & 0 & 0 \\ 0 & 1/I_y & 0 \\ 0 & 0 & 1/I_z \end{bmatrix} \begin{bmatrix} M_{d,x} \\ M_{d,y} \\ M_{d,z} \end{bmatrix} \quad \text{Eq. 4-36}$$

With the disturbance model taken into account, the controller can be able to compensate for the aerodynamic effect. For example, according to the characteristics of the vehicle, an increase in thrust from propellers 1 and 2 in Figure 4-6 increases the induced velocities over wing segments 2 and 4 in Figure 4-6. The vehicle pitches down, as there is an increase in the lift force on the AC of the wing

and creates an extra moment. As a result, an increase in thrust induces a sudden unfavourable pitch movement and may cause a deviation in the position in the x -direction. Using this aerodynamic disturbance model, the controller can compensate this unwanted aerodynamic effects. With the notices of this disturbance, the controller has successfully weakened this effect as the blue line shown in Figure 4-23

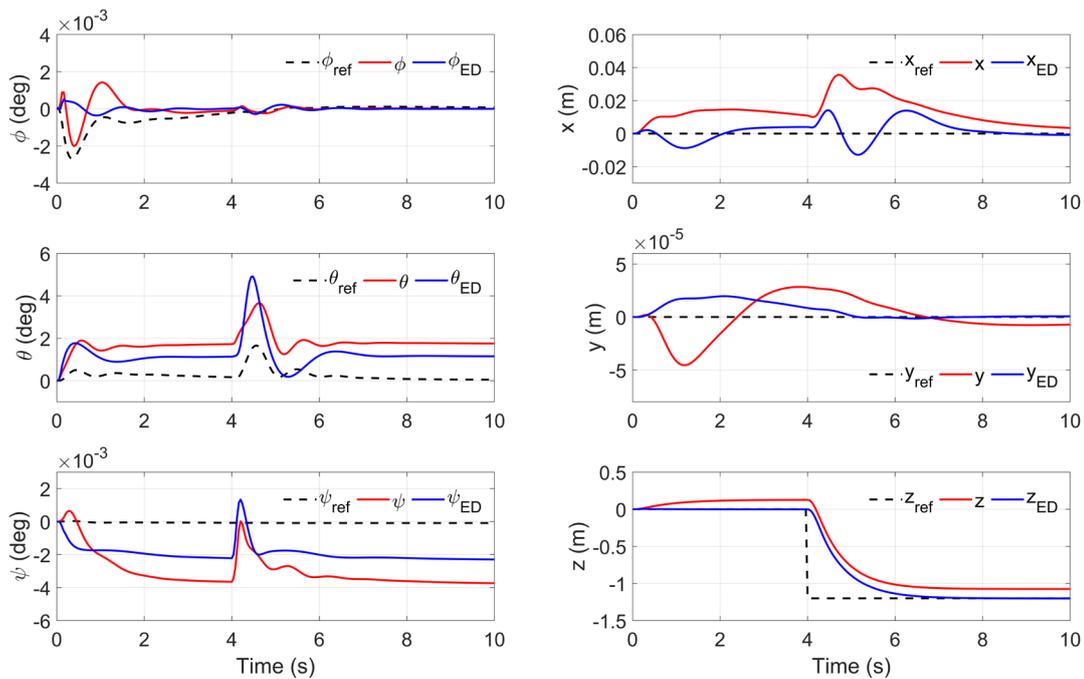


Figure 4-23 Simulation result of SLMPC controller following a step signal in the z -direction with (blue) and without (red) disturbance model

Another example is the change in yaw motion whenever there is a command on the roll. Roll motion is achieved by the difference of rotation speed of motor 1 and 2, which create a torque to roll. However, the difference in the rotation speed of motor 1 and 2 will also cause uneven lift force on each side of the wing and creating unwanted yaw motion simultaneously. The controller with the disturbance model also has successfully weakened this effect.

4.3.5 Effectiveness of Unknown Disturbance Model

As mentioned in chapter 3.1.1, the environment and aerodynamic effect that can be modelled will be eliminated by a feedforward strategy of an estimated disturbance model. The other disturbances, such as white noise, model mismatch, or unknown disturbance mentioned in chapter 3.1.3 will be

compensated by a feedback integrating strategy, which is aiming to eliminate any steady-state error and achieve offset-free control.

Compare to Figure 4-21, with the implementation of the integration function on both the position and attitude controller shown in Figure 4-24, the vehicle can reach the step commands and resist the wind in a better manner. Especially in the attitude control that the Euler angle chases the command much precise than before, resulting in no oscillation and smaller overshoot.

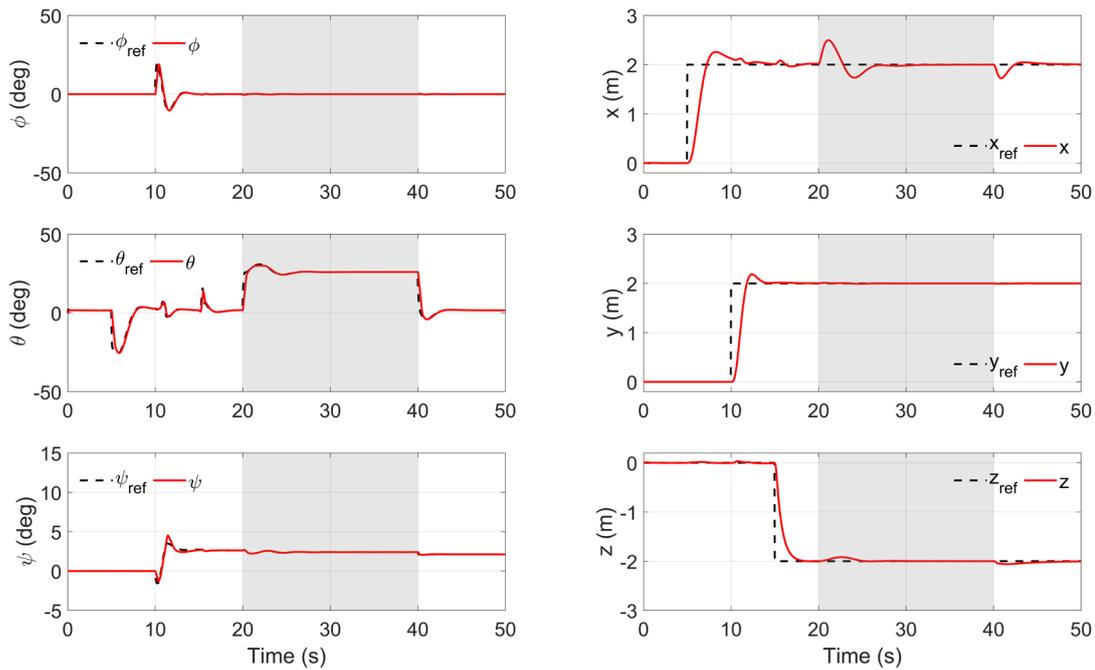


Figure 4-24 Simulation result of the effectiveness of the elimination of unknown disturbance under 2m/s wind condition

4.3.6 Time-Varying Weights

According to the objective function described in Eq. 3-14, when $e \neq 0$, the objective function drives the output error towards zero, with a priority that depends on the weight parameters, to minimize the cost. In general, a step command will create a large, sudden output error, the cost of the objective function increases significantly, and the control law will consequently use the maximum effort to eliminate it immediately. In most cases, this maximum effort causes excessive movement and often results in an unacceptable oscillatory response, especially in real-time flight tests.

To ease the system's response and reduce oscillation, a large value for input increment weight ($R_{\Delta u}$) is used, which penalizes large input increments (Δu) at every control horizon. From Eq. 3-14

the cost will increase as $R_{\Delta u}$ increases for the same Δu . Thus, in order to lower the cost, the controller will give smaller Δu as $R_{\Delta u}$ increases. Meanwhile, the output of Eq. 3-14 can also be adjusted to ease the system response by varying the output variable weight (Q) from one step to the next. In order to allow Q varies as the horizon changes, Eq. 3-14 can be written as:

$$J(z_k) = \sum_{i=0}^{P-1} \{ [e_y^T(k+i) Q_i e_y(k+i)] + [\Delta u^T(k+i) R_{\Delta u} \Delta u(k+i)] \} \quad \text{Eq. 4-37}$$

where Q_i is a diagonal matrix containing weight information of each output variables and is described as:

$$Q_i = \text{diag}[Q_i^1 \quad Q_i^2 \quad \dots \quad Q_i^j] \quad \text{Eq. 4-38}$$

for $i = 0:P - 1, \quad j = 1:n_y$

The coefficient in Q_i is linearly increased from the beginning of the horizon to its set value. The cost will no longer increase abruptly, and the first move will be smaller, resulting in an increase of robustness. Although this approach complicates the tuning process, it provides an additional opportunity to adjust the control performance.

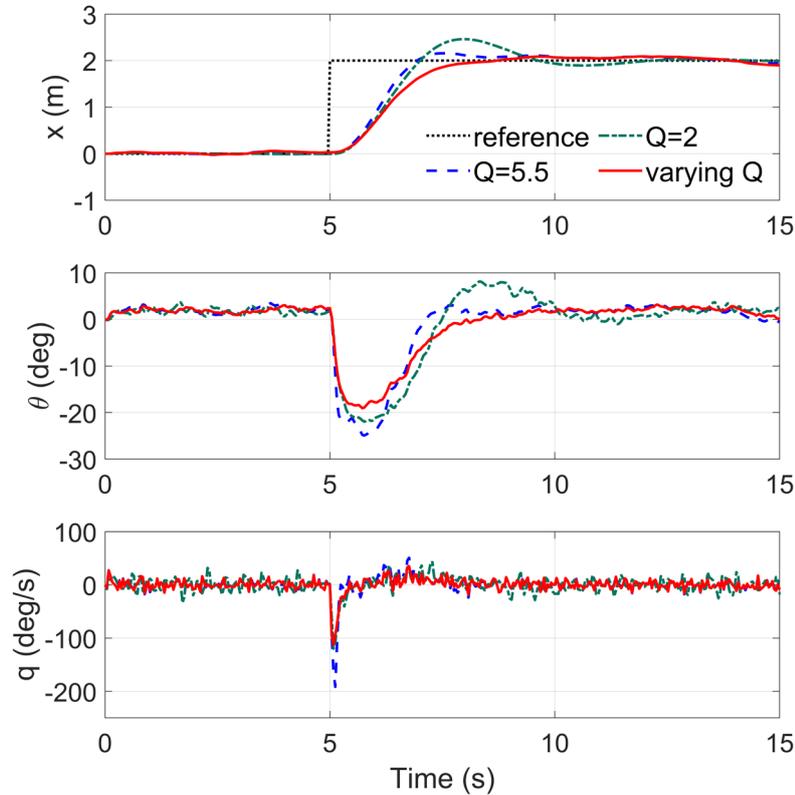


Figure 4-25 Simulation results of step-command following with fixed weights of $Q = 2.0$. and $Q = 5.5$ and a varying weight.

According to the experiences of flight tests, any large and abrupt control command is to be avoided since it will cause an excessive response of the vehicle and usually need extra effort for correction. To achieve stable and efficient control when a large and abrupt control command is encountered, the time-varying weight is a good measure to ease the response.

The simulation results of following a step command with measurement noise in the x -direction position are shown in Figure 4-25. The results of fixed weights of $Q = 2.0$ (green line) and $Q = 5.5$ (blue line) and a time-varying weight (red line) are compared. According to the objective function, a smaller fixed weight $Q = 2.0$ in the x -position will make the cost smaller, compared with the case of $Q = 5.5$. The priority of eliminating the error in the x -direction is lower when $Q = 2.0$, yielding a much larger overshoot in the x -direction and a much longer regression time. Increasing Q from 2.0 to 5.5 will reduce the overshoot in the x -direction and shorten the regression time, yet it creates large system responses in θ and q . Further, an increase in the fixed weight Q will deteriorate system responses in θ and q . The most significant improvement of the time-varying weight added, compared to the fixed weight $Q = 5.5$, is that the maximum pitch angular rate (q) at the 5th second is reduced from from $-200^\circ/s$ to $-100^\circ/s$ and the maximum pitch angle (θ) also decreases. However, the minimum time required to reach a range of 5% error in controlling the x -axis position is similar. In the real flight, it is obvious that a lower maximum angular rate requires a much smaller sudden change in torque, which is strongly related to the rotation speed, and result in lower energy consumption. Above all, the results indicate that the time-varying weight can ease the control responses and improve the robustness without compromising the performance.

4.3.7 Velocity as Reference

The position controller is trying to bring the position error value to zero at every time step. The position error is defined as follows:

$$\mathbf{p}_{error} = \mathbf{p}_{ref} - \mathbf{p}_{mea} \quad \text{Eq. 4-39}$$

where \mathbf{p}_{ref} is the reference position and \mathbf{p}_{mea} is the measured position. Currently, the controller has state contains both the position and the velocity ($\mathbf{x} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T$). Usually, only the position $[x \ y \ z]$ is given with weights in the objective function since, in most of the case, a trajectory contains only position reference. However, the time derivative of the position error \mathbf{v}_{ref} can be set as a velocity reference for $[\dot{x} \ \dot{y} \ \dot{z}]$. In addition to the position weight, \mathbf{v}_{ref} is also given with weights in the

objective function. Not only the position is considered and controlled by the optimization process, but also the velocity. The v_{ref} is assumed to be the position error divided by a time constant T_c , as follows:

$$v_{ref} = \frac{p_{error}}{T_c} \quad \text{Eq. 4-40}$$

The smaller the time constant is, the faster the response is. The weight of v_{ref} in the objective function can create an effect similar to that of the derivative term for traditional PID control. When there is a large position error, the objective function generates a large command. As it nears the target, the position error becomes small, so the vehicle starts to slow down and reduces the overshoot effect.

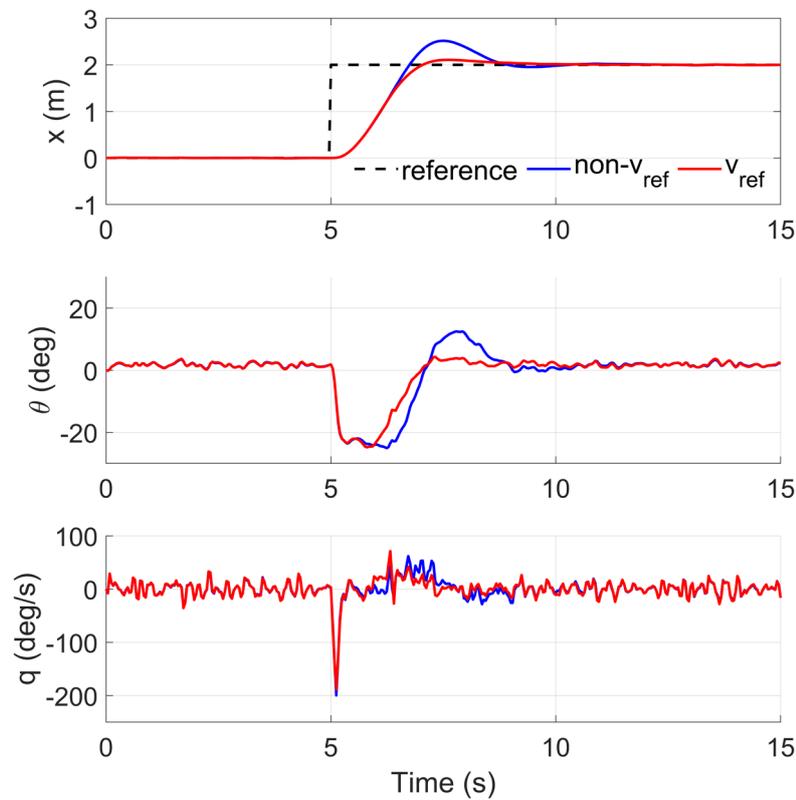


Figure 4-26 Simulation results of the step command following with (red) and without (blue) velocity as reference.

A simulation result of following a step signal by the proposed SLMPC controller is shown in Figure 4-26. For the SLMPC control itself (blue line), it tends to use maximum control effort to minimize the error between the reference and the output to achieve a smaller value of the objective function. As a result, longer duration in the maximum pitch angle ($\theta \approx -20^\circ$) is commanded and reaches the setpoint earlier. However, the subsequent pitches up motion ($\theta > 0$) slows down the x -direction velocity, but overshoot has already been caused.

With the velocity reference added (red line), the pitch angle (θ) starts to reduce at the 6th second rather than at the 6.5th second with the original controller (blue line). Compared with the case without the velocity reference added (blue line), this earlier decrease in the pitch angle can reduce the x -direction speed at an earlier stage and as a result, a much smaller overshoot is observed in the x -direction position (red line). Notably, for the case without the velocity reference added (blue line), there is a significant positive pitch movement between the 7th second and the 9th second to compensate for the overshoot of the x -direction position. Meanwhile, the reaction time of the system is not compromised because both cases reach the reference value at the same time in the x -direction position. With this measure taken, in the real-time flight tests, the vehicle will have less fluctuation in both the attitude control and position control.

4.3.8 Simulation Results and Comparison

The linear MPC controller from the previous chapter and the proposed SLMPC controller are compared by performing a trajectory following mission simulation. The detailed parameter settings are presented in Table 1 for the SLMPC controller used in the ‘PolyU Plus’ tail-sitter vehicle.

Table 1. Parameter settings for the SLMPC controller.

Parameter	Value
T_s	0.04
P	50
H	4
Q	$diag\{5.5,4,5.5,2,2,2.5\}$
$R_{\Delta u}$	$diag\{10,35,30,25\}$
$ y $	$[inf, inf, inf, 5,5,5]^T$
$ \Delta u $	$[10,0.75,0.75,0.7]^T$
S_y	I
S_u	$[15,0.8,1,1]^T$

The trajectory is a 4-meter diameter circular path in the same high to better distinguish the performance in altitude tracking. The performance on position tracking is shown in Figure 4-27. For the linear MPC controller (blue line), a steady-state error in the x -direction can be observed in Figure 4-28 due to the aerodynamic lift and drag of the wing itself has not been considered by the controller. When it starts to track the circle trajectory, significant unwanted variation in the z -direction can also be observed. According to the characteristics of this tail-sitter vehicle, the maneuverability in the x -direction is much weaker than the y -direction due to the wing. As a result, from Figure 4-29, following the circle path requires a maximum 5° in roll (ϕ) but 20° in pitch (θ). Whenever the pitch is large, the

high drops (+0.2m in the z-direction). Since the linear MPC controller is developed base on a zero-degree Euler angle trim point. A 20° in pitch is too far from the trim point, the model is no longer matched and leading to inaccurate in the control effort. The z-direction is ignored while maximum effort has been given in keeping the x-direction causing this drop in high. The high is restored only when the priority of the x-direction is lower.

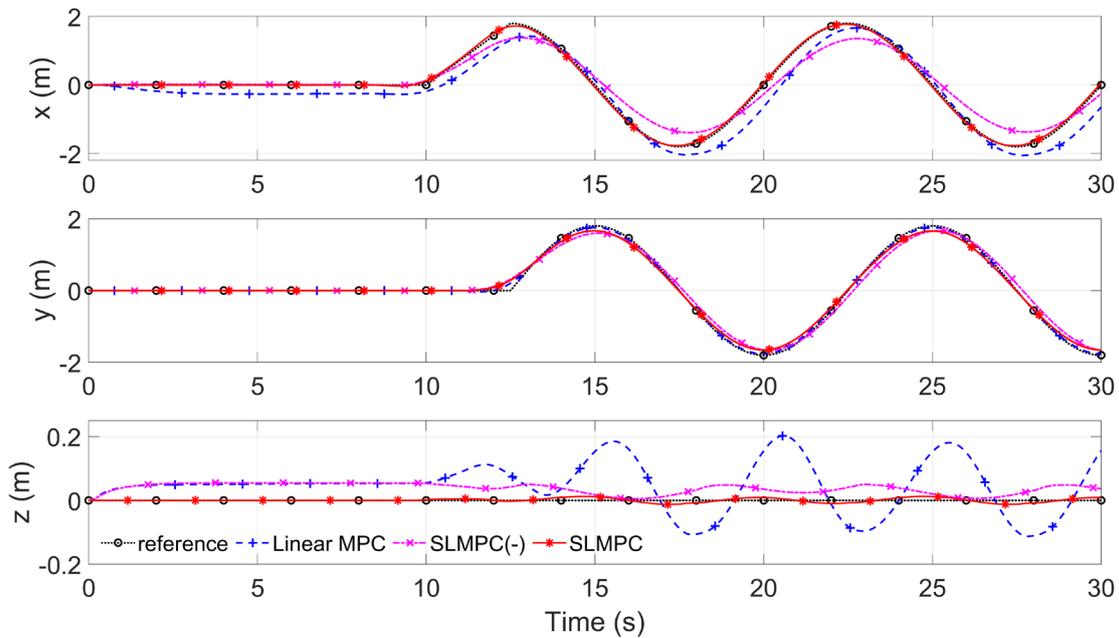


Figure 4-27 Position of a trajectory following simulation of a circular path.

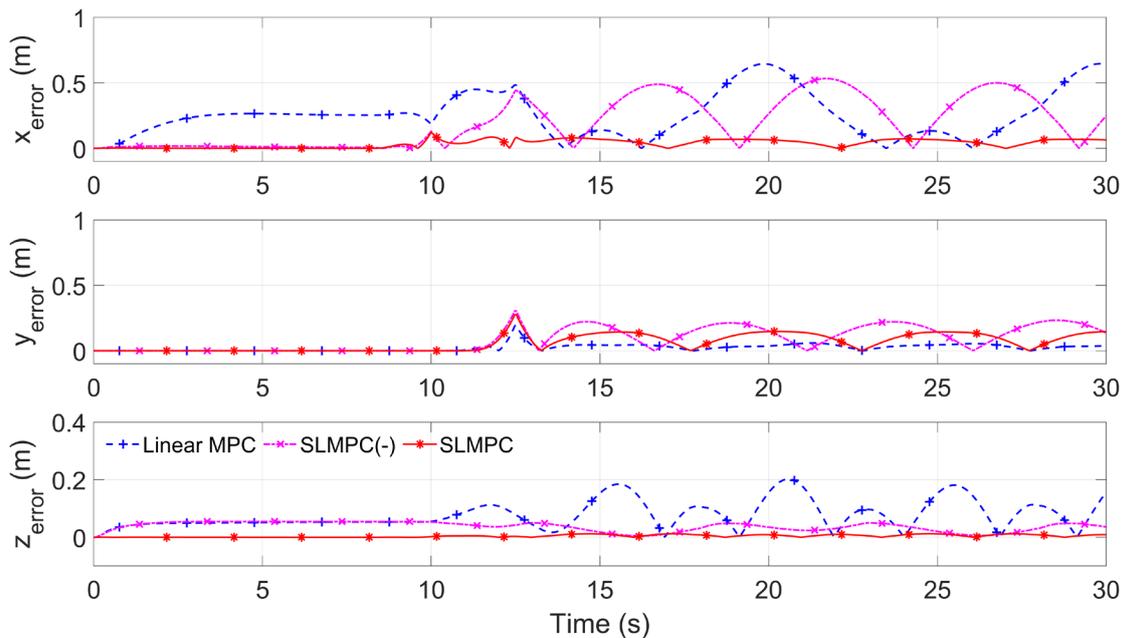


Figure 4-28 The corresponding error of Figure 4-27.

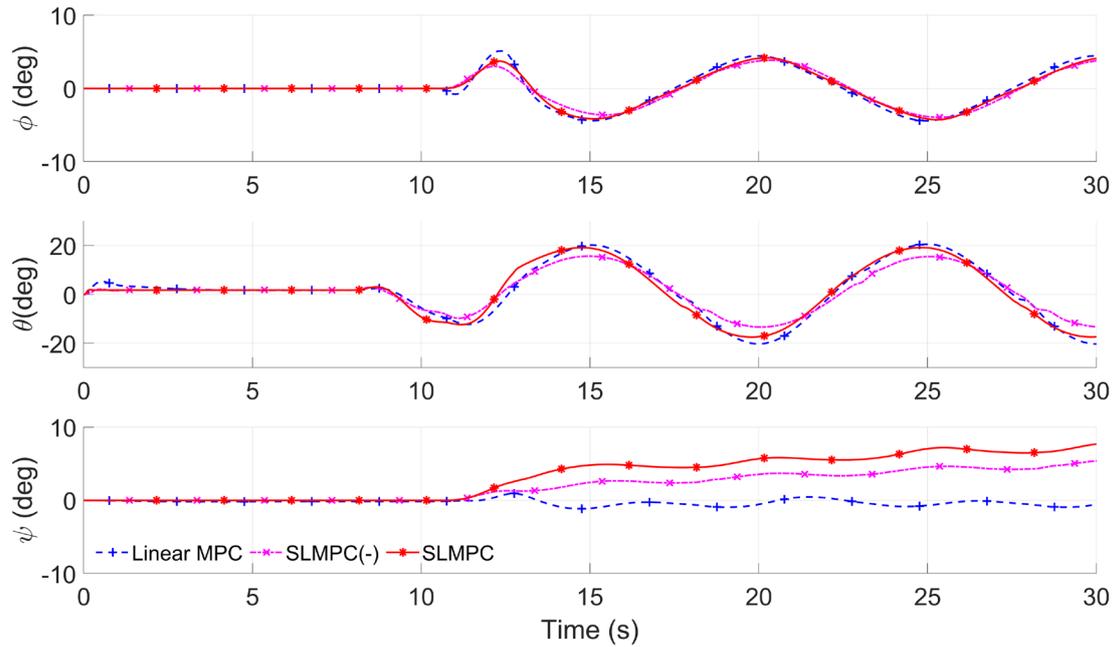


Figure 4-29 The Euler angles of trajectory following simulation of a circular path.

For the SLMPC without the augmented model (pink line), the performance in the x-direction is also imprecise, no matter how the weight has been adjusted in the objective function. However, the z-direction has a slightly better performance than the linear MPC even though the steady-state error still exists. For the proposed SLMPC with the improvement methods augmented (red line), the following of the reference command in all direction is precise and stable without any steady-state error.

From Figure 4-29, it can be observed that the yaw angle (ψ) of the linear MPC has stayed in 0° while the yaw angle of SLMPC has deviated from it. As shown in Figure 4-19, the yaw angle command (ψ_{cmd}) is generated by the MPC position controller and followed by the PID attitude controller. Although the yaw angle is rather irrelevant to the vehicle position, it is included in the control input in (10) and still affects the objective function and the position optimization. Unlike the linear MPC which commands the yaw angle to its trim point (0°), the yaw command from the SLMPC is its current yaw angle. Due to the aerodynamic characteristics of the vehicle, whenever there is a roll movement, the yaw changes. From Figure 4-2, a roll movement is achieved by the difference of rotation speeds of propellers 1 and 2, which create a torque to roll. The difference in rotation speeds of propellers 1 and 2 will cause the different slipstream velocities on the wing surface. As a result, uneven lift forces on two sides of the wing simultaneously cause an unwanted yaw motion. Real-time linearization is adapted by the SLMPC control which can immediately accept the current yaw angle on the optimization process and control the position with the lowest cost, consequently. Unlike the linear MPC control, there is no extra control effort spent on yawing the vehicle back to 0° .

Disturbance resistance controlled by the linear MPC and SLMPC with the wind of 1 m/s, 2 m/s and 3 m/s magnitudes coming from 0° to north (negative X^1 -direction in Figure 4-2) is presented in Figure 4-30. As shown, the shaded area indicates the period when the wind encounters the vehicle. The x-direction position error increases as the wind speed increases. Under the large wind speed of 3 m/s, the linear MPC cannot steer the vehicle back to reference value until the wind effect disappears. The vehicle becomes unstable when controlled by the SLMPC because it fluctuates around the reference value and converges slowly. The corresponding root mean square errors ($RMSE = RMSE_x + RMSE_y + RMSE_z$) are presented in Figure 4-31, with the wind speed ranging from 0.5 m/s to a tolerance of 3.3 m/s. The RMSE of SLMPC is smaller than that of the linear MPC and the performance of SLMPC in the extreme case of 3.3 m/s wind speed is much better than the linear MPC. The vehicle controlled by a linear MPC has failed to hold the position under a 3.3 m/s wind since it has been completely blown away in the x-direction before the cease of the wind.

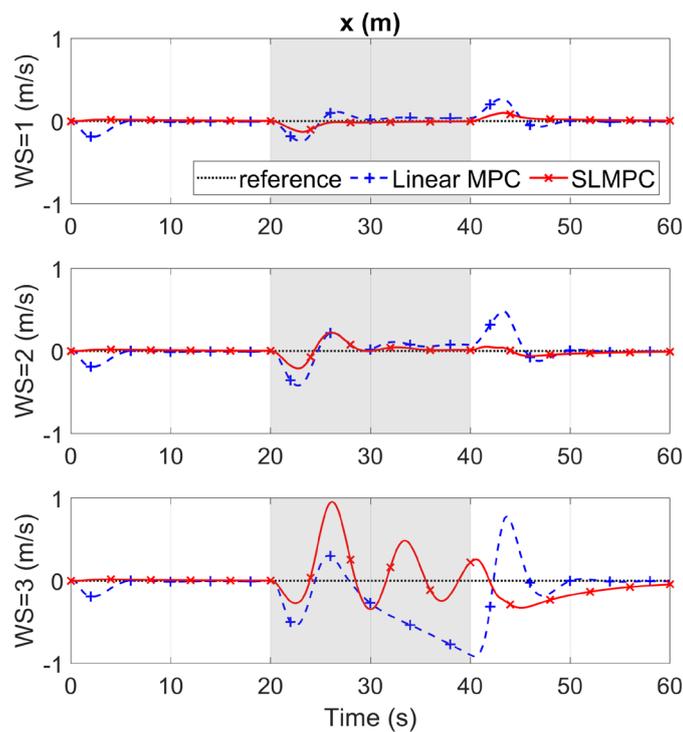


Figure 4-30 Simulation of position control with disturbance rejection when the wind of 1m/s, 2m/s and 3m/s magnitudes coming from 0° to north (negative X^1 -direction in Fig. 1).

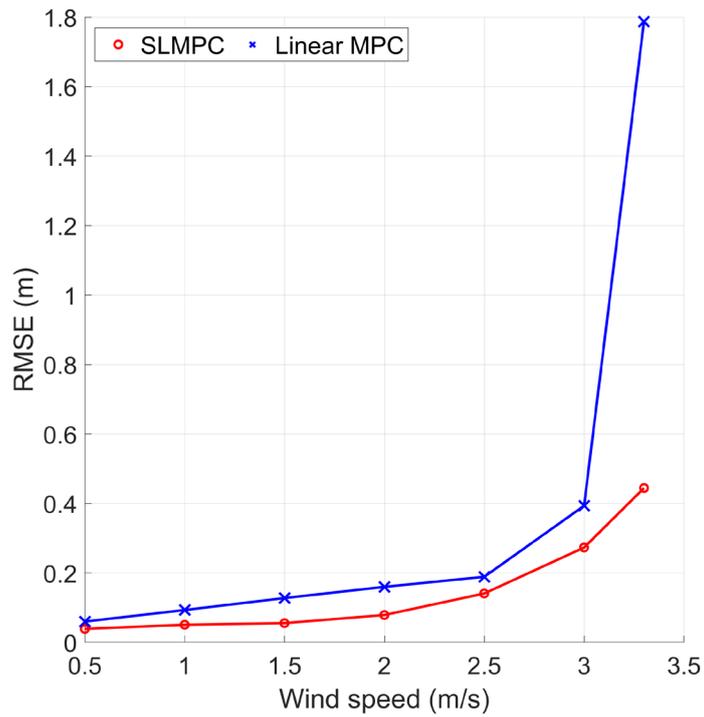


Figure 4-31 Root mean square errors of controllers with disturbance rejection when the wind speed ranges from 0.5 m/s to 3.3 m/s and comes from 0° to north (negative X^I -direction in Fig. 1).

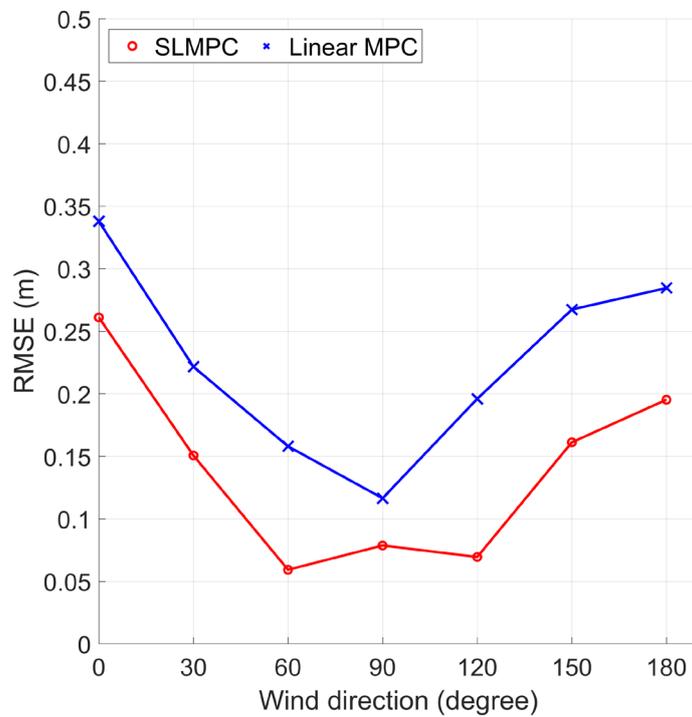


Figure 4-32 Root mean square errors of controllers with disturbance rejection when the wind of 3 m/s coming from 0° to 180°.

The effectiveness of disturbance resistance by both the controllers with the wind of 3 m/s coming from 0° to 180° (clockwise) to the north is illustrated in Figure 4-32. From the results, the variation of RMSE complies with the characteristics of the vehicle. The errors caused by the wind coming from $\pm x$ -direction (0° or 180°) is much larger than that from the y -direction (90°), since there is a large wing surface area facing the wind in the $\pm x$ -direction. Overall, the RMSE of SLMPC is smaller than that of the linear MPC.

4.3.9 Flight Tests and Results

The SLMPC controller with a build-in estimated aerodynamic disturbance model was installed as position control of the vehicle in the indoor flight test of trajectory following under the same condition as Chapter 4.2.4. In Figure 4-33, the result on altitude keeping was as good as the linear MPC controller shown in Figure 4-14. The consistence update of the nominal states and model parameters have significantly help to prevent model mismatch at large angle movement, which is -12 degree in pitch. Also, compared to the linear MPC controller Figure 4-14, the performance here is more stable and less oscillatory. The oscillation frequency and magnitude are lower, the trend of angular movement is clearer and result in a tighter trajectory tracking.

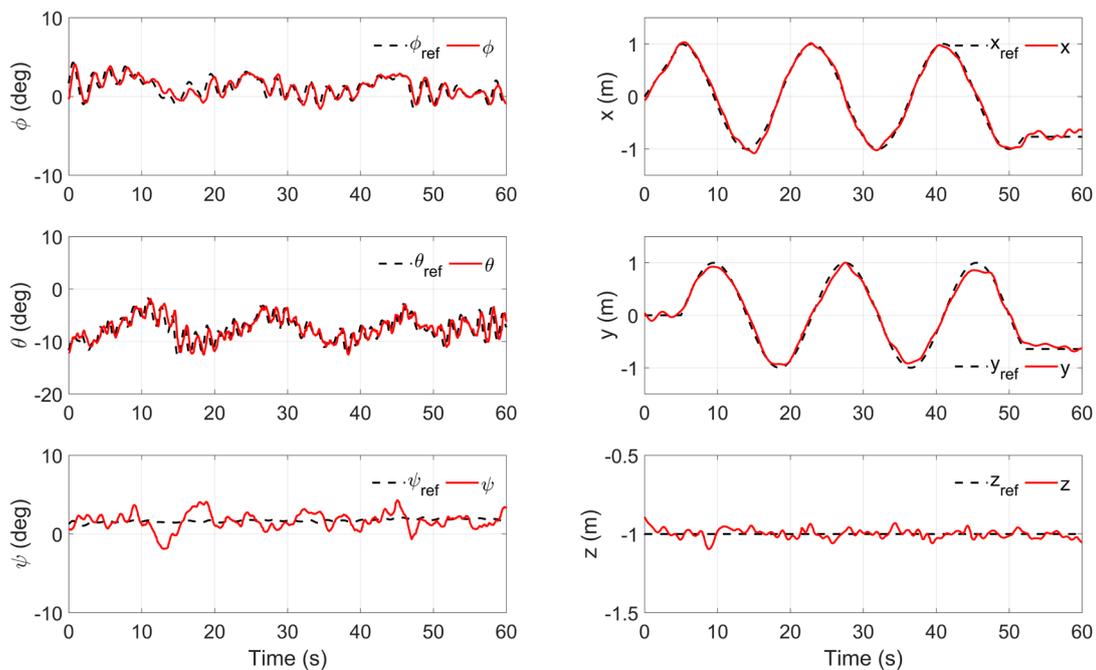


Figure 4-33 Experimental results of SLMPC position controller following a circular path

Case 3 (Proposed SLMPC Control): The flight test results for the proposed SLMPC position controller are shown in Figure 4-34. When the wind arrives, the pitch down command is clear and obvious. Compared with the linear MPC controller in Figure 4-18, the proposed controller shows a more stable and less oscillatory angular movement. The controls in the x - and z -directions are more precise. The consistent updating of the nominal states and model parameters prevents model mismatch for large angular movements. The yaw movement is mostly caused by the non-uniform wind field and the large moment arm of the wing. This effect occurs in all three cases but is much smaller in both the MPC control (less than 10°) than the PID control (up to 30°) in Figure 4-17. The root mean square error values of the position error under wind disturbance using the SLMPC controller are the smallest among all three tests, as shown in Table 2. The beauty of the SLMPC method lies in its ability to simultaneously optimize and adapt the real-time model to achieve minimum control effort and minimum model mismatch.

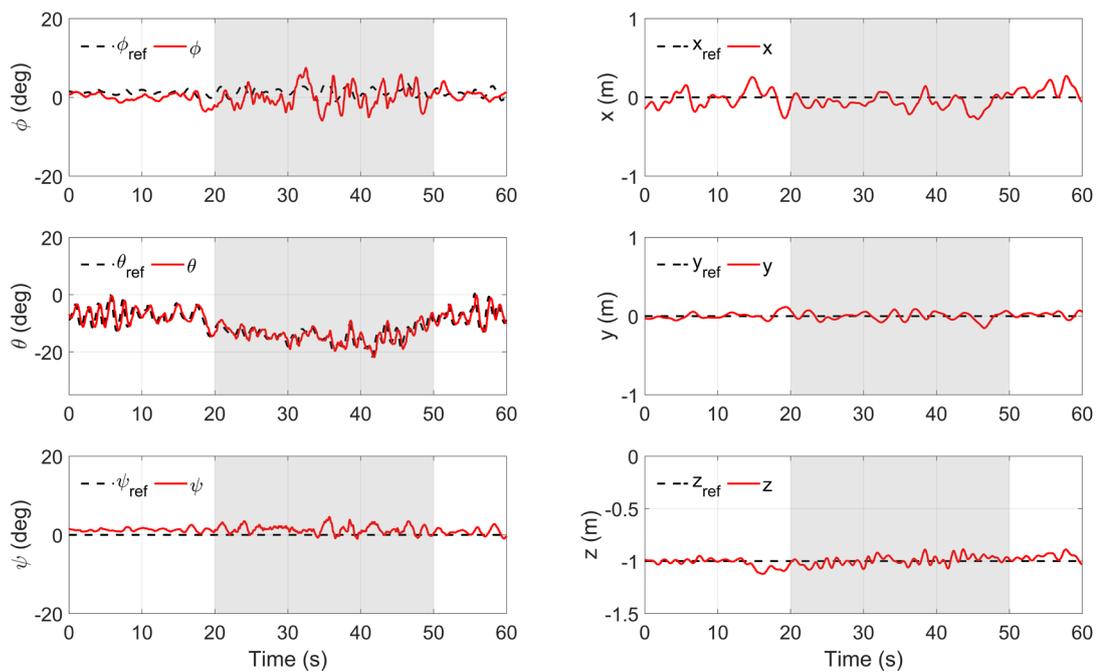


Figure 4-34 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using the SLMPC controller

Table 2. Comparison of the root mean square error values (m) of the position error under wind disturbance, using the traditional PID, linear MPC (LMPC) and successive linearization based MPC (SLMPC) controllers.

RMSE	x	y	z
PID	0.2214	0.0889	0.1451
LMPC	0.1615	0.0478	0.0668
SLMPC	0.1298	0.0459	0.0413

Figure 4-35 shows the resistance of a larger wind condition by the same AMPC position controller. While under large wind conditions of $4\text{m/s} \pm 1\text{m/s}$, the controller tried to resist, and the vehicle has already pitched down until about the 45th second but fail to continue. Before the 50th second, the vehicle's attitude started to fluctuate severely leading to an unsuccessful position hold. As a result, at the 50th second, the vehicle has been blown away by 0.5m in the x-direction. At this point, the objective function should have a large cost, since the position error has kept accumulating, even though maximum effort on pitch has been given for a while.

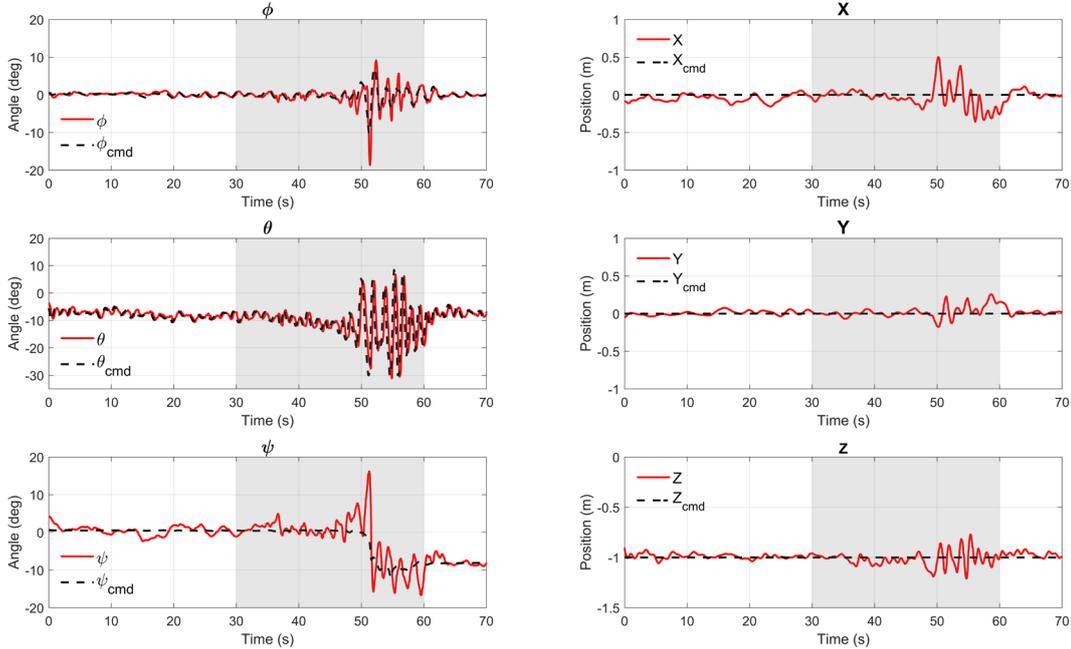


Figure 4-35 Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using the SLMPC controller for 4m/s wind resistance

Meanwhile, the yaw angle started to vibrate from the 45th second and reach the first peak right after the 50th second. Since the moment arm from the wing to the center of mass is large, it became very effective for the uneven wind to yaw the vehicle. The moment when the yaw angle reached its peak, the roll angle also reached its peak and error in x-direction suddenly decreased. Since then, the controller started to give consistent command on the yaw angle until about -10° . The root mean square value of x, y and z error was [0.1143, 0.0595, 0.0548].

What has happened was that the objective function was at a large value at the 51st second and it was also been blown to yaw at the 51st second. As it yawed to more than $+15^\circ$, the roll angle become effective in helping the x-position. Because there is no wing in the y-direction, the roll movement was always more effective in controlling position than pitch. Thus, the roll movement can affect the x-position under a certain yaw angle and the error in the x-direction can be reduced by both the effort of roll and pitch movement. Then the cost of the objective function was decreased in alongside with the decrease in x-error. As a result, to keep the vehicle's position but with a smaller cost, the control law started to give command on the yaw angle until the position was successfully held.

This is the beauty of an adaptive MPC control which can do optimization as well as real-time model adapt. Without the optimization, the command on yaw will never appear to enhance the control effort on x-direction. Without the adaptation ability, the model mismatch problem will become severe as the yaw angle was far away from zero.

5. Cruise Flight Control

In this section, system identification was implemented in practice on a novel quadrotor tail-sitter UAV in cruise stage with innovations in experimental design and regression problem development. The advantage of using a training line to transmit external control command to a flying UAV is taken, and no other researchers have attempted this approach before. A least-square problem was developed and augmented with ridge regression regularization to achieve better balance between important and less-important parameters. A new procedure with confidence in system identification is proposed to implement the identified models into an MPC control for validation. By investigating the control performance, the accuracy of the identified model is verified. The goal of this study is to enable a UAV developer who is not familiar with the system identification technique to obtain a useable model to facilitate the design and implementation of a higher-level model-based controller to the UAV platform.

Controlling the cruise stage by a traditional PID control may cause a saturation problem on the control surfaces. Because the command to the control surfaces is the overlap of the roll command and pitch command, which are separately generated, their summation may already exceed the maximum control effort. The MPC method can generate optimized actuator inputs within the predefined constraints. Its control inputs are calculated by minimizing the control objectives based on the prediction of the future response according to the UAV model. Keeping all inputs within the constraints is always considered in the objective function. It can usually achieve better performance for a wide range of operating conditions regarding the aspect of lower sensitivity to variation in system properties and shorter settling time, etc. (shown in [84, 85]).

For a UAV in cruise, the real-time situation may change rapidly and become severe. The one-meter wingspan of the studied UAV will easily catch disturbances and brings challenges to the controllability. The consideration of expected and unexpected disturbances in the controller design is important. Measurements can be taken to predict and reduce any unwanted wind disturbance [86]. Special treatment on the controller can be designed to improve the robustness and stability under a windy condition [87, 88]. In the transition period, the optimization method has been applied to the altitude variation and energy consumption in [89].

5.1 System Identification Setup

The coordinate system of the body and the wind axis are shown in Figure 5-1. Unlike a conventional fixed-wing UAV, the tail-sitter UAV has only one pair of control surfaces that act as

elevons, which control the roll and pitch movement simultaneously. All four motors are directly controlled by the thrust command. As a result, they always rotate at the same speed in the cruise flight.

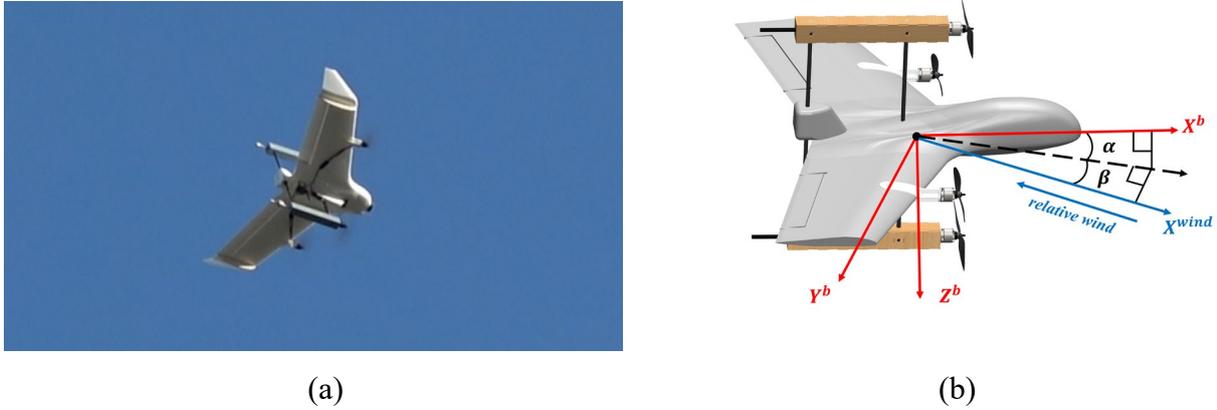


Figure 5-1 (a) The PolyU Plus tail-sitter UAV in cruise. (b) the coordinate system of the body and wind axis.

5.1.1 Grey-box Model of Longitudinal Dynamic

A grey box model for the tail-sitter aircraft dynamic can be described from the equation of motion in the body-axis coordinates. The standard nomenclature is used: $x - y - z$ velocities in body axes are (u, v, w) in m/s ; the rotation Euler angles are (ϕ, θ, ψ) in rad ; and the angular rates are (p, q, r) in rad/s . The angle of attack is α , and the sideslip angle is β . The aerodynamic forces on body axes are denoted by \mathcal{X} , \mathcal{Y} and \mathcal{Z} , while the corresponding aerodynamic moments are denoted by \mathcal{L} , \mathcal{M} , and \mathcal{N} . The aircraft inputs consist of angles of control surfaces and the throttle. The combination of the elevator command and the aileron command controls a pair of control surfaces. Their variations around a trim condition are given by the variables δ_T , δ_e , and δ_a , respectively, and are normalized between 0 and 1. The four motors are rotating in different directions, and their gyroscopic effect has been balanced out. The thrust T and weight G are assumed to act at the center of gravity.

The longitudinal equations of motion are shown in Eq. 5-1

$$\begin{cases} \mathcal{X} = m(\dot{u} + qw - rv) + G\sin(\gamma) \\ \mathcal{Z} = m(\dot{w} + pv - qu) - G\cos(\gamma)\cos(\phi) \\ \mathcal{M} = \dot{q}I_y - pr(I_z - I_x) - (r^2 - p^2)I_{xz} \end{cases} \quad \text{Eq. 5-1}$$

where m is the mass and I is the moment of inertia. Meanwhile, the thrust forces T , drag D , and lift L acting on the body can be expressed as Eq. 5-2:

$$\begin{cases} \mathcal{X} = T\cos(\alpha) - D \\ \mathcal{Z} = -T\sin(\alpha) - L \end{cases} \quad \text{Eq. 5-2}$$

To simplify the system in the longitudinal direction, the multiplication terms are ignored. Assuming small angles of attack and combining Eq. 5-1 and Eq. 5-2, the model is formed in Eq. 5-3:

$$\left\{ \begin{array}{l} m \frac{du}{dt} = T - D - G \sin \gamma \\ m u \frac{d\gamma}{dt} = -L + G \cos \gamma \\ I_y \frac{dq}{dt} = \mathcal{M} \\ \frac{d\theta}{dt} = q \\ \gamma = \theta - \alpha \end{array} \right. \quad \text{Eq. 5-3}$$

where γ is the flight path angle. For small angles of attack, the forces and moment can be expressed as Eq. 5-4 by applying Taylor expansion on relevant variables around the trim point.

$$\begin{aligned} T &= T_0 + T_{\delta_T} \Delta \delta_T \\ L &= L_0 + L_u \Delta u + L_\alpha \Delta \alpha + L_q \Delta q + L_{\delta_e} \Delta \delta_e + L_{\delta_T} \Delta \delta_T \\ D &= D_0 + D_u \Delta u + D_\alpha \Delta \alpha + D_q \Delta q + D_{\delta_e} \Delta \delta_e \\ \mathcal{M} &= \mathcal{M}_0 + \mathcal{M}_u \Delta u + \mathcal{M}_\alpha \Delta \alpha + \mathcal{M}_q \Delta q + \mathcal{M}_{\delta_e} \Delta \delta_e + \mathcal{M}_{\delta_T} \Delta \delta_T \end{aligned} \quad \text{Eq. 5-4}$$

Replacing the forces and moment in Eq. 5-3 with Eq. 5-4 and applying small-perturbation theory [90], we can have a model around the trim point condition. The model can be written in a state-space form with the state $\mathbf{x}_{lon} = \{\Delta \bar{u}, \Delta \alpha, \Delta q, \Delta \theta\}$, $\mathbf{x}_{lon} \in \mathbb{R}^{N_x}$ and input $\mathbf{u}_{lon} = \{\Delta \delta_T, \Delta \delta_e\}$, $\mathbf{u}_{lon} \in \mathbb{R}^{N_u}$.

$$\begin{aligned} \dot{\mathbf{x}}_{lon} &= \mathbf{A}_{lon} \mathbf{x}_{lon} + \mathbf{B}_{lon} \mathbf{u}_{lon} \\ \mathbf{y}_{lon} &= \mathbf{C}_{lon} \mathbf{x}_{lon} \end{aligned} \quad \text{Eq. 5-5}$$

where

$$\mathbf{A}_{lon} = \begin{bmatrix} -\frac{D_u}{m} & \frac{g}{v_0} - \frac{D_\alpha}{m v_0} & -\frac{D_q}{m v_0} & -\frac{g}{v_0} \\ \frac{L_u}{m} & \frac{L_\alpha}{m v_0} & \frac{L_q}{m v_0} + 1 & 0 \\ \frac{\mathcal{M}_u v_0}{I_y} & \frac{\mathcal{M}_\alpha}{I_y} & \frac{\mathcal{M}_q}{I_y} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \mathbf{B}_{lon} = \begin{bmatrix} -\frac{D_{\delta_e}}{m v_0} & \frac{T_{\delta_T}}{m v_0} \\ \frac{L_{\delta_e}}{m v_0} & \frac{L_{\delta_T}}{m v_0} \\ \frac{\mathcal{M}_{\delta_e}}{I_y} & \frac{\mathcal{M}_{\delta_T}}{I_y} \\ 0 & 0 \end{bmatrix}; \mathbf{C}_{lon} = \mathbf{I}_{4 \times 4};$$

$$\bar{u} = \frac{u}{v_0}; v_0 = \sqrt{u^2 + v^2 + w^2}.$$

5.1.2 Grey-box Model of Lateral Dynamic

The lateral equations of motions are shown in Eq. 5-6:

$$\begin{cases} \mathcal{Y} = m(\dot{v} + ur - wp) - G\cos\theta\sin\phi \\ \mathcal{L} = \dot{p}I_x + qr(I_z - I_y) - (\dot{r} + pq)I_{xz} \\ \mathcal{N} = \dot{r}I_z + pq(I_y - I_x) - (\dot{p} - qr)I_{xz} \end{cases} \quad \text{Eq. 5-6}$$

Unlike any other traditional aircraft, the configuration of the tail-sitter UAV shown in Fig. 1 has no rudder. As a result, the yaw angular rate r and yaw moment \mathcal{N} are not controlled. By applying Taylor expansion on the force and moment, they can be expressed as:

$$\begin{aligned} \mathcal{Y} &= \mathcal{Y}_0 + \mathcal{Y}_\beta\Delta\beta + \mathcal{Y}_p\Delta p + \mathcal{Y}_{\delta_\alpha}\Delta\delta_\alpha \\ \mathcal{L} &= \mathcal{L}_0 + \mathcal{L}_\beta\Delta\beta + \mathcal{L}_p\Delta p + \mathcal{L}_{\delta_\alpha}\Delta\delta_\alpha \end{aligned} \quad \text{Eq. 5-7}$$

The small-perturbation theory is then applied in the equilibrium condition of Eq. 5-6. By substituting Eq. 5-7 into Eq. 5-6, we can have a model around the trim point condition as shown:

$$\begin{cases} mv_0 \left(\frac{d\Delta\beta}{dt} \right) = \mathcal{Y}_\beta\Delta\beta + \mathcal{Y}_p\Delta p + \mathcal{Y}_{\delta_\alpha}\Delta\delta_\alpha + G\Delta\phi \\ I_x \frac{d\Delta p}{dt} = \mathcal{L}_\beta\Delta\beta + \mathcal{L}_p\Delta p + \mathcal{L}_{\delta_\alpha}\Delta\delta_\alpha \\ \frac{d\Delta\phi}{dt} = \Delta p \\ \Delta\beta = \Delta v/v_0 \end{cases} \quad \text{Eq. 5-8}$$

The model in Eq. 5-8 can be written in a state-space form like Eq. 5-5 with the state $\mathbf{x}_{lat} = \{\Delta\beta, \Delta p, \Delta\phi\}$, $\mathbf{x}_{lat} \in \mathbb{R}^{N_x}$ and input $\mathbf{u}_{lat} = \{\Delta\delta_\alpha\}$, $\mathbf{u}_{lat} \in \mathbb{R}^{N_u}$. The simplified state matrix \mathbf{A}_{lat} , control matrix \mathbf{B}_{lat} and output matrix \mathbf{C}_{lat} can be represented as:

$$\mathbf{A}_{lat} = \begin{bmatrix} \frac{\mathcal{Y}_\beta}{mv_0} & \frac{\mathcal{Y}_p}{mv_0} & \frac{g}{v_0} \\ \frac{\mathcal{L}_\beta}{I_x} & \frac{\mathcal{L}_p}{I_x} & 0 \\ 0 & 1 & 0 \end{bmatrix}; \mathbf{B}_{lat} = \begin{bmatrix} \frac{\mathcal{Y}_{\delta_\alpha}}{mv_0} \\ \frac{\mathcal{L}_{\delta_\alpha}}{I_x} \\ 0 \end{bmatrix}; \mathbf{C}_{lat} = \mathbf{I}_{3 \times 3} \quad \text{Eq. 5-9}$$

5.1.3 Least Square Regression with Regularization

Model parameters are obtained by fitting measured data to the predicted model response. The least-square regression is a curve-fitting algorithm for finding a parameter estimate, which consists of a vector of unknown model parameters σ .

$$\hat{\sigma} = \arg \min_{\sigma} V(\sigma) \quad \text{Eq. 5-10}$$

That is, the parameter estimates are obtained by minimizing a weighted quadratic norm of the predicted error $\varepsilon(t, \sigma)$:

$$V(\sigma) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \sigma) \quad \text{Eq. 5-11}$$

where t is the time variable and N is the number of data samples. The predicted error ε is computed as the difference between the observed output and the predicted output:

$$\varepsilon(t, \sigma) = y(t) - \hat{y}(t, \sigma) \quad \text{Eq. 5-12}$$

in which vector σ is subject to bound contains $\sigma = \{\sigma: lb \leq \sigma \leq ub\}$. lb is a lower bound, and ub is an upper bound. $\hat{y}(\ast)$ is the predicted output from the grey box model response, and y is the measured data.

The standard of an accurate model should have a small mean square error (MSE), which is the sum of systematic error (bias) and random error (variance):

$$MSE = |Bias|^2 + Variance \quad \text{Eq. 5-13}$$

where the bias can be minimized by using a higher order model because it can fit the observed data with higher accuracy. However, it will cause a higher uncertainty at the same time (i.e., the variance) due to the increase in flexibility. Alternately, systematic errors causing by the model mismatch will dominate MSE if a model is chosen with an overly low order. Thus, the minimization is a tradeoff in constraining the model. A flexible (high-order) model gives small bias and large variance, whereas a simpler (low-order) model results in a larger bias and smaller variance errors. In grey-box models as derived above, the order is fixed by the underlying ODEs and cannot be changed. If the data are not rich enough to capture the full range of dynamic behavior, it may lead to high uncertainty in the estimated values.

Regularization is the technique for specifying constraints on the flexibility of a model, thereby reducing uncertainty in the estimated parameter values [91]. First, a concept from statistics and machine learning called ridge regression is introduced in Eq. 5-14:

$$\hat{\sigma} = \arg \min_{\sigma} V(\sigma) + \lambda \|\sigma\|^2 \quad \text{Eq. 5-14}$$

where λ is known as the ridge regression coefficient. This term penalizes the parameter values with the effect of keeping the values small. The larger λ is, the higher the bias and lower the variance of σ

is. This coefficient can usually increase the resemblance of the prediction model to the validation model [92].

Then, a positive definite matrix R acting as a penalty term is added to Eq. 5-14, which represents the confidence in the prior knowledge of the parameters:

$$\hat{\sigma} = \arg \min_{\sigma} V(\sigma) + \lambda \sigma^T R \sigma \quad \text{Eq. 5-15}$$

The matrix R gives additional freedom for shaping the penalty term λ to each of the parameters in σ . In the grey-box model, some of the estimated parameters in the initial guess may be trustworthy according to the physical model. To accommodate this phenomenon, the regularization in Eq. 5-15 can be generalized into:

$$\hat{\sigma} = \arg \min_{\sigma} V(\sigma) + \lambda (\sigma - \sigma^*)^T R (\sigma - \sigma^*) \quad \text{Eq. 5-16}$$

where σ^* is the initial guesses for the unknown parameters. At this point, $\lambda * R$ represents the confidence in the initial guesses. Minimizing this cost function has the effect of estimating σ such that some value remains close to the initial guess and some are freed but small.

In the system identification process in section 5.3, the values of σ^* are decided first. For unknown parameters without initial guesses, zero will be taken. There are some states that play a more important role in the control, and as a result, the estimation accuracy of these states should be higher. Thus, the R value is higher in the state $\Delta \bar{u}$ and $\Delta \theta$ in the longitudinal direction. The same method is also applied to the lateral model. Last but not least, the λ value is obtained by trial and error.

5.1.4 Trust Region Algorithm

Trust region algorithms are a class of reliable and robust algorithms for solving optimization problems by iteration [93]. This algorithm class has the advantage of strong convergence properties.

Assume that the solution of the optimization problem is being guessed according to the initial conditions. Consequently, an approximate model can be constructed near the current guess point. The solution of this approximated model is then taken as the next iteration point. For general nonlinear functions, local approximate models can only fit the original function locally. As a result, the approximate model is only ‘trusted’ in a region near the current iteration point. The region that the approximate model can be trusted is called the trust region. The trust region is adjusted from iteration to iteration.

A trust-region algorithm contains the following steps: (i) Set up an approximate model along with a trust region from the initial conditions; (ii) Solve the local subproblem with a solution s_k , which is called a trial step; (iii) Apply a merit function to decide whether the trial step should be accepted; (iv) Update the next trust-region and choose the new iteration point; (v) Stop the iteration when the reduction of the original objective function is smaller than a user-defined value (e.g., $1 \times e^{-6}$) or when the trust-region no longer increases.

The trust region subproblem lies in each iteration, and a quadratic model is used to approximate the original objective function. Then, the optimization problem is essentially reduced to solve a sequence of trust-region subproblems. At the k^{th} iteration, the subproblem can be expressed as:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \zeta_k(d) &= g_k^T d + \frac{1}{2} d^T H_k d \\ \text{s. t. } \|d\|_2 &\leq r_k \end{aligned} \tag{Eq. 5-17}$$

where $g_k = \nabla \cdot \hat{V}(\sigma_k)$ is the Jacobian at the current iteration, $H_k = \nabla^2 \cdot \hat{V}(\sigma_k)$ is the Hessian matrix, and r_k is the positive trust-region radius. Let s_k be the solution or the local minimum of the above subproblem. An empirical threshold value ρ_k is shown below to decide whether the trial step s_k is accepted or not and to adjust the new trust-region radius r_k .

$$\rho_k = \frac{\mathcal{O}_{red_k}}{\mathcal{P}_{red_k}} \tag{Eq. 5-18}$$

for which

$$\begin{aligned} \mathcal{O}_{red_k} &= \hat{V}(\sigma_k) - \hat{V}(\sigma_k + s_k) \\ \mathcal{P}_{red_k} &= \zeta_k(0) - \zeta_k(s_k) \\ \text{s. t. } \mathcal{P}_{red_k} &> 0 \end{aligned} \tag{Eq. 5-19}$$

where \mathcal{O}_{red_k} is the actual reduction gained by the original objective function and \mathcal{P}_{red_k} is the predicted reduction expected in the local approximate model. A solution of an approximate model can be taken as the next iteration point. If the current iteration makes a satisfactory reduction in \mathcal{O}_{red_k} , which means this approximate model fits the original model well, the trust-region can be enlarged in the next iteration. Otherwise, when we achieve a limited improvement at the current iteration, radius r_k remains unchanged or even decreases in a worse case.

5.2 Experimental setup

The aircraft is instrumented with a Pixhawk 4 Mini from a Holybro® flight control unit that is small and lightweight. It has a main flight management unit (FMU) processor and a built-in inertia measurement unit (IMU), which consists of two accel/gyro sensors, a magnetometer, and a barometer. No extra sensor is installed onboard because the built-in sensors provide enough data for the system identification process.

The shape of an input signal has a major impact on the excitation of the aircraft, and it can strongly affect the accuracy of the estimation [94, 95]. In the flight experiment, the aircraft is required to fly at a trim condition and to maneuver one actuator at a time. Apart from the waveforms, the amplitudes need special care to excite certain dynamic modes [68]. The input can be a frequency sweep [69], doublet and 3-2-1-1 waveforms [96-98], etc. It has been suggested in [68] that the maneuvers should not exceed $\pm 5^\circ$ angles of attack, $\pm 20^\circ/s$ angular rates and $\pm 0.3 g$ translational accelerations. Therefore, a signal generator module was designed and built to input the desired signal shape and magnitude.

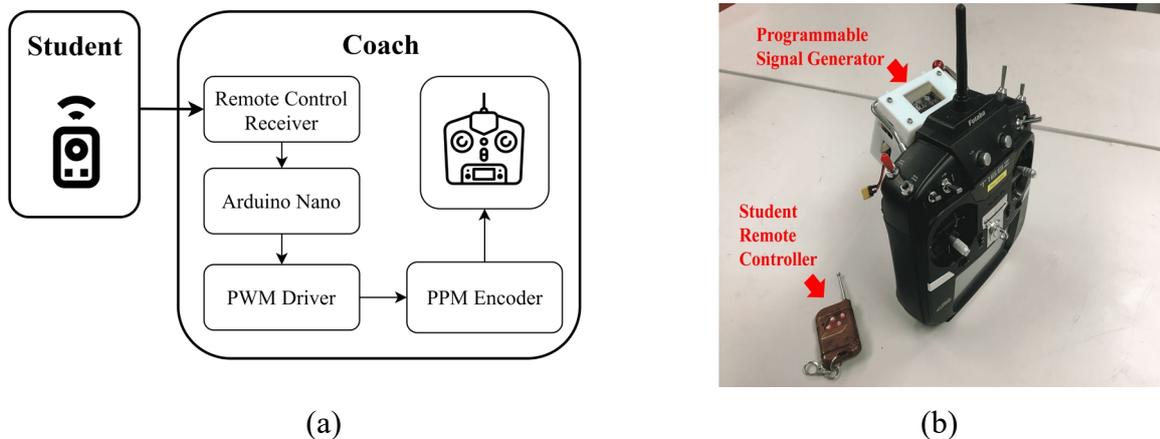
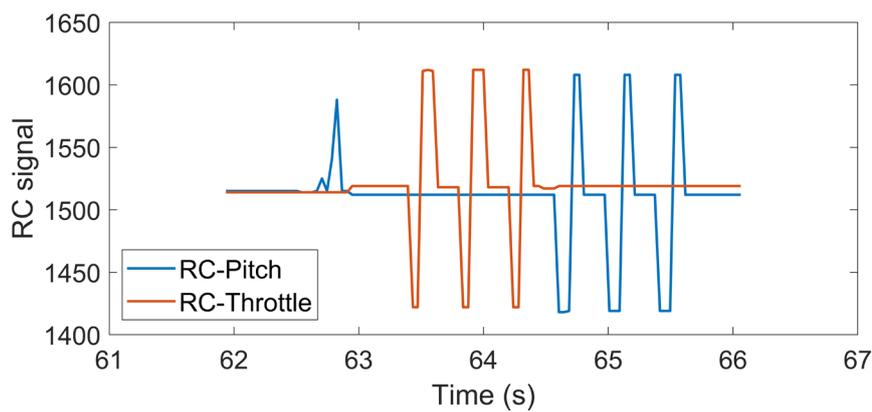


Figure 5-2 (a) Schematic diagram of the signal generator module through the training line of the remote control transmitter. (b) The setup of the signal generator and the remote control transmitter.

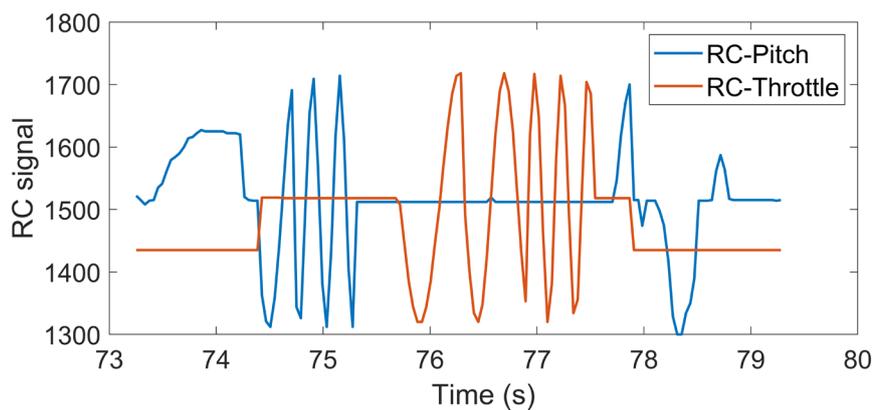
The training line of a remote control is used to send the specifically designed input to the aircraft safely. As shown in Figure 5-2 (a), two people work together, with one acting as a student and the other acting as a coach. The student holds a simple remote control with four buttons, and he can choose the desired input waveform. A receiver will transmit the signal from the student's remote control to an Arduino Nano microcontroller board, which recognizes the signal and generates the corresponding input waveform. A pulse-width modulation (PWM) driver converts the waveform signal into the PWM signal. Because the training line port takes only the pulse position modulation (PPM) signal, a PPM

encoder is used to convert the PWM signal into the PPM signal before feeding to the training line. In Figure 5-2 (b), a programmable signal generator has been built and attached to the back of the coach's remote transmitter. This position is close to the training line port, and it will not interrupt the coach.

The coach is responsible for flying the aircraft, keeping it safe, and trying to seize opportunities to implement the designed input. When the coach switches to the training line, he will temporarily give up his control, and the flight controller will receive the signal from the training line. As long as the coach notices any unusual or unsafe behavior of the aircraft, he can regain control immediately by a click of the switch to save the aircraft from a crash. Among all types of methods that can implement the series of designed input signals for the aircraft, this method can ensure safe flight during identification flights. Alternately, this method can be implemented for any other types of aircraft very easily because the onboard flight controller remains untouched and the hardware package is small and portable and has been stacked on the back of our remote control.



(a)

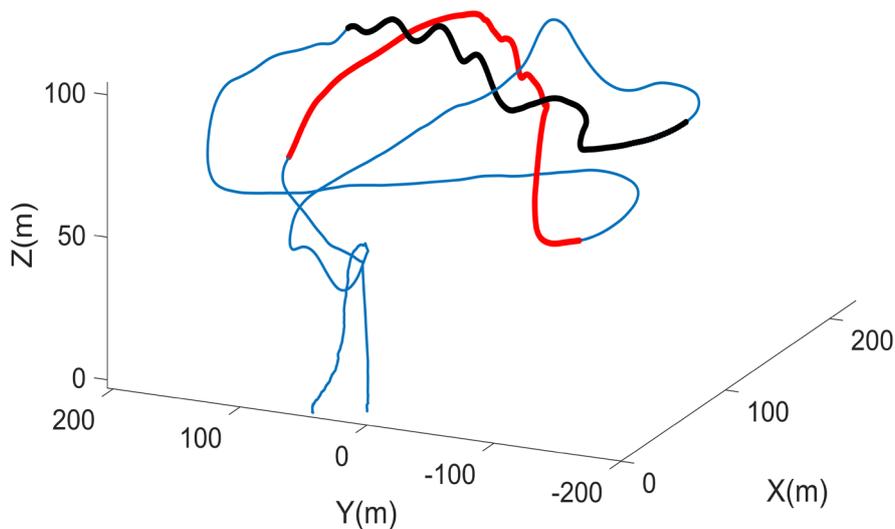


(b)

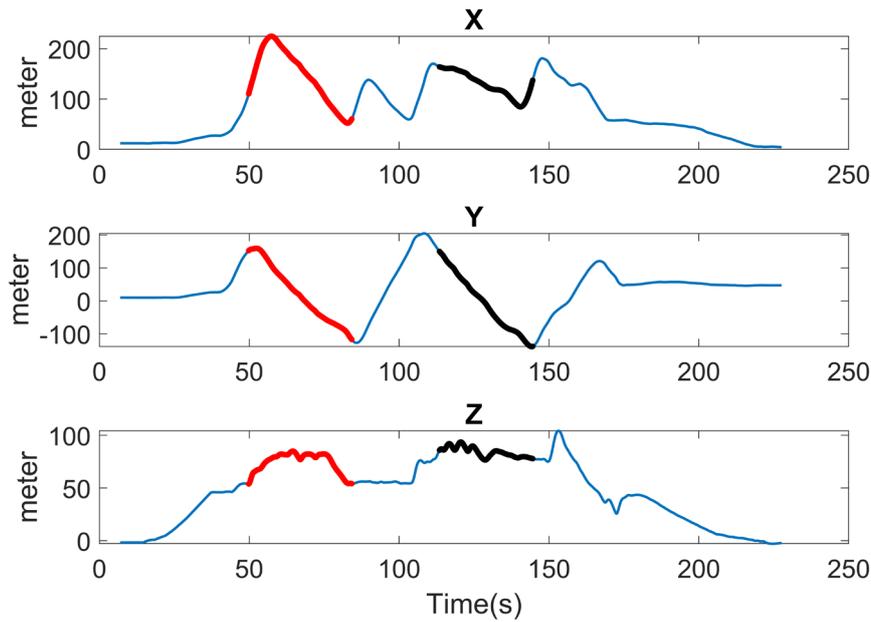
Figure 5-3 (a) The flight log data of the remote control input signal of a doublet waveform. (b) The flight log data of frequency sweep of a sin wave.

Figure 5-3 (a) is a section recorded by the flight log showing that doublet waveforms were successfully generated and sent to the flight controller. Meanwhile, the frequency sweep of a sin wave is presented in Figure 5-3 (b). Notice that the input signal apart from the designed waveforms is manual input by the pilot. The flights were performed under closed-loop control conditions. That means that the UAV will be automatically controlled by the onboard flight controller even though the inputs from the remote control in Figure 5-3 were injected. This allows a UAV to maintain flight velocity and retain stability in the presence of external disturbances. The remote control on the throttle and pitch angle will be fused and distributed into airspeed and altitude control by the flight controller. The input deflection of control surfaces designed here for the purpose of gathering system identification data will affect the UAV but not in a direct way. As a result, the control command of throttle and elevon from the log data may not have the same pattern (as shown in Figure 5-5 and Figure 5-6 later).

Figure 5-4 (a) shows the entire flight path, in which the UAV took off vertically at the origin. The transition to forward flight occurred at a height of 50 m. The red section of the flight path with a period of 31 seconds was selected for system identification purposes. There is both a straight flight path for longitudinal direction movement and a turning flight path for lateral direction maneuvers. The black section of the flight section with a period of 28 seconds for validation usage also has the characteristics mentioned above. In Figure 5-4 (b), climbing and declining for up to 30 meters in height is included in the training set (red). In the meantime, excitation in pitch and throttle during this straight flight period caused significant fluctuation in height from 60 seconds to 80 seconds. In the aspect of lateral direction, the UAV made two turns. The first turn was rapid, and the other was gentle. This motion was strong enough to excite the lateral direction dynamic to capture different modes of it.



(a)



(b)

Figure 5-4 (a) The flight sections for the system identification training set (red) and validation set (black) within the entire flight path (blue) in the 3D view. (b) The flight sections in three separate axes view.

5.3 System Identification and Results

Before conducting the system identification, a postprocessing procedure of flight data is essential so that the flight data can be better fit into the grey box model and the accuracy of the parameter estimation can be increased. The data are extracted from the flight log. The measurements are at different frequencies. For example, the airspeed sensor measures at 100 Hz, the attitude is recorded at approximately 40 Hz, and the GPS records at less than 10 Hz. To conduct system identification, the data size of each parameter has to be the same. The parameter with a small data size is interpolated to increase its data size without changing its characteristics.

The grey box models were designed as first-order models. It is undesirable for it to fit a high-frequency response. Therefore, the data were low pass filtered. The cut-off frequency for the first-order model is set at 4 Hz to 5 Hz according to [67, 70]. Because a state-space model is being used to describe the measured responses in this research, bias and scale factor errors should also be removed.

The time histories of measured and reconstructed longitudinal state variables are depicted in Fig. 6. Small errors between the measured data and the reconstructed data in the pitch angular rate q and the pitch angle θ are shown in Figure 5-5 (a). In this case, θ was set to have the highest weighting, such that its fitness is the best of all. Although the reconstructed data are not fit tightly with the flight

data in the nondimensional heading velocity \bar{u} in the body frame and the angle of attack α , their mean features are successfully captured. These deviations can be easily caused by the wind disturbance during the flight because a consistent wind or a sudden wind disturbance can strongly affect \bar{u} and α . Figure 5-5 (b) shows a comparison between the validation data and the reconstruction data from the identified model. It can be seen that q has a good resemblance, while the characteristics of the other three states are mostly captured in the validation data.

In the lateral direction, the roll angle ϕ was set to have the highest weighting. As a result, the best fit was observed between the flight data and the reconstructed data in Figure 5-6 (a). Similar to the longitudinal dynamics, the wind effect is also important in the lateral motion because a crosswind is very likely to affect β significantly. Figure 5-6 (b) demonstrates the resemblance of the validation data to the reconstructed data from the identified model. The similarity in the roll angle ϕ is acceptable.

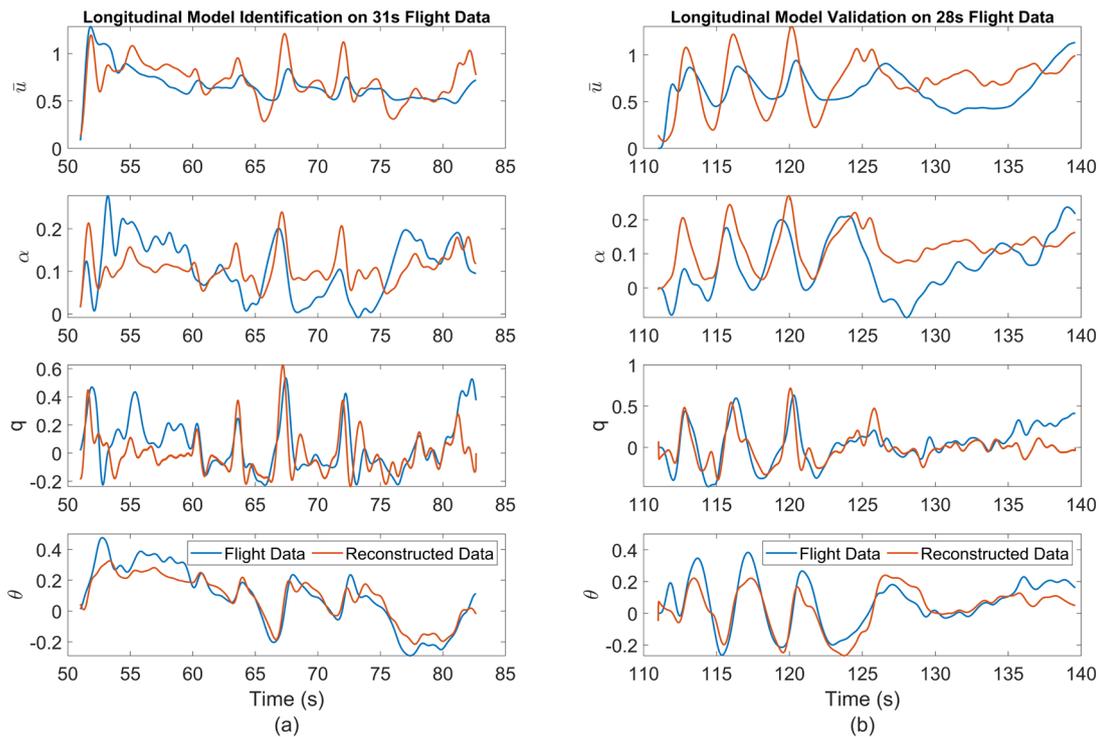


Figure 5-5 (a) The longitudinal flight data (blue) and the reconstructed data (red) for the learning data set.
 (b) The validation data set according to the grey-box longitudinal model.

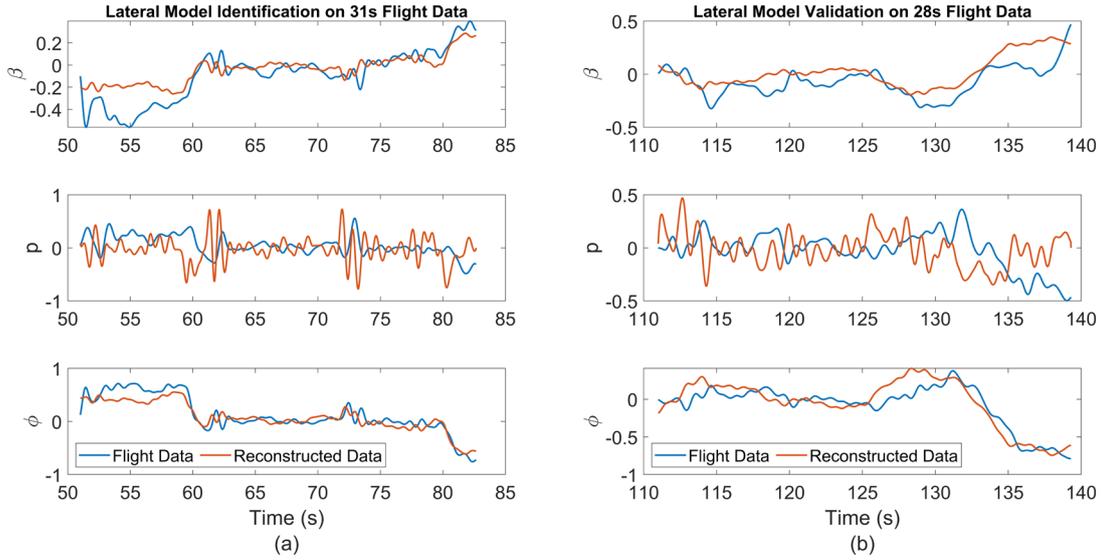


Figure 5-6 (a) The lateral flight data (blue) and the reconstructed data (red) for the learning data set. (b) The validation data set according to the grey-box lateral model.

Summarily, the state-space longitudinal and lateral models of Eq. 5-5 and Eq. 5-9 are concluded in Eq. 5-20, which will be used in model predictive control in section 5.4.

$$\mathbf{A}_{lon} = \begin{bmatrix} 0.2 & 1.751 & 2.072 & -0.65 \\ -0.109 & -0.928 & 0.159 & 0 \\ 0.344 & -0.104 & 0.353 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \mathbf{B}_{lon} = \begin{bmatrix} 1.327 & -0.312 \\ 0.889 & 0.0845 \\ -0.492 & 0.170 \\ 0 & 0 \end{bmatrix}; \quad \text{Eq. 5-20}$$

$$\mathbf{A}_{lat} = \begin{bmatrix} 0.961 & -0.669 & 0.65 \\ 0.287 & 1.131 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \mathbf{B}_{lat} = \begin{bmatrix} -0.581 \\ 0.125 \\ 0 \end{bmatrix}.$$

5.4 Linear MPC Controllers

The schematic diagram demonstrates the application of the MPC controller in the longitudinal and lateral directions in Figure 5-7.

The longitudinal MPC controller will receive the heading speed command u_{cmd} in the body frame and a pitch angle command θ_{cmd} generated by an altitude PID controller. The model in the longitudinal MPC controller is constructed according to the model in Eq. 5-5 and the identified parameters in Eq. 5-20. The same process also applies to the lateral MPC controller. After that, the longitudinal MPC controller will generate control inputs of thrust δ_T and elevator δ_e , while the lateral MPC controller will produce an aileron input δ_a . An actuator map will distribute these three control

inputs into correct throttle commands $\eta_{1\sim4}$ of the four motors and the degrees $\Delta_{1\sim2}$ of the two control surfaces. The states that the MPC controller needs will be measured from the dynamic model and fed to the controller.

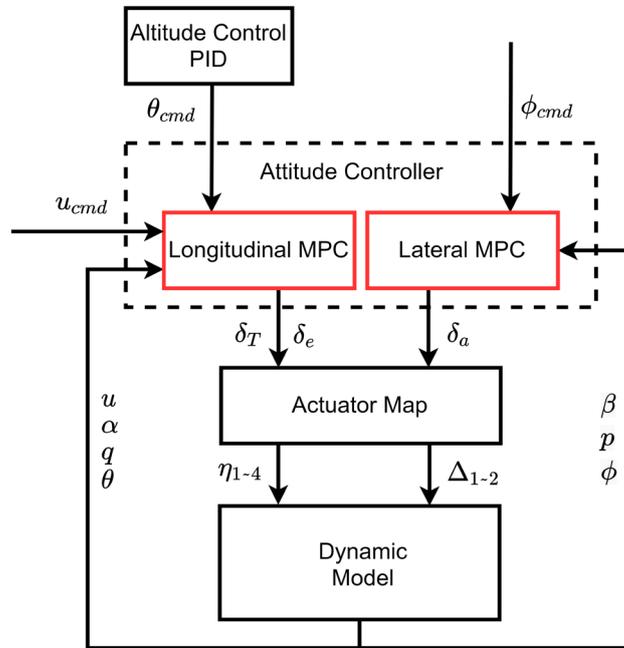


Figure 5-7 A schematic diagram of application of the model predictive control to longitudinal and lateral attitude controllers.

This simulation model includes not only the basic dynamic and characteristics of this tail-sitter UAV but also a theoretical-based aerodynamic model and an experimental-based propulsion model. In this study, we have applied this developed dynamic model to cruise control to validate the accuracy and practicability of the identified model.

To demonstrate the ability for practical usage, the PX4 fixed-wing position mode logic is imitated in the simulation. In this closed-loop logic, the pilot controls the left joystick to change the air speed and the spin angle, while the right joystick controls the climbing/descent rate and the roll angle. The airspeed command determines the heading speed u_{cmd} in Figure 5-7. The climbing/descent rate determines the pitch angle θ_{cmd} by a PID controller. The spin angle is neglected because this UAV does not have a rudder.

The simulation result is shown in Figure 5-8. At this point, there is no wind effect included in the simulation because, in this study, we are focusing on model validation rather than controller development. The altitude reference z_{ref} is given in place, between the 10th and 30th seconds. The

pitch angle reference θ_{ref} generated by the altitude PID controller is consequently given to the longitudinal MPC controller and closely followed. During this time, the airspeed has a small variation around the trim point condition of 16 m/s. After that, between the 40th and 50th seconds, an increase in heading speed reference u_{ref} is given to the longitudinal MPC controller. We can see that the UAV can increase the speed as the reference, but it cannot slow down as much as the reference. This is due to the characteristics of a fixed-wing UAV. There is no breaking mechanism onboard, and the only way to slow down is by drag force. As a result, it is reasonable for the tail-sitter UAV to have this performance in the heading speed variation. Last but not least, a roll command reference ϕ_{ref} is given to the lateral MPC controller in between the 60th and 80th seconds. The result has shown a good fit of the measured value to the reference value.

All three control inputs of the attitude controller have been tested. The results have shown that the MPC controllers can precisely control the UAV. The identified models are demonstrated to be sufficiently accurate. In the next step, outdoor flight tests of this UAV will be carried out for model validation and controller development.

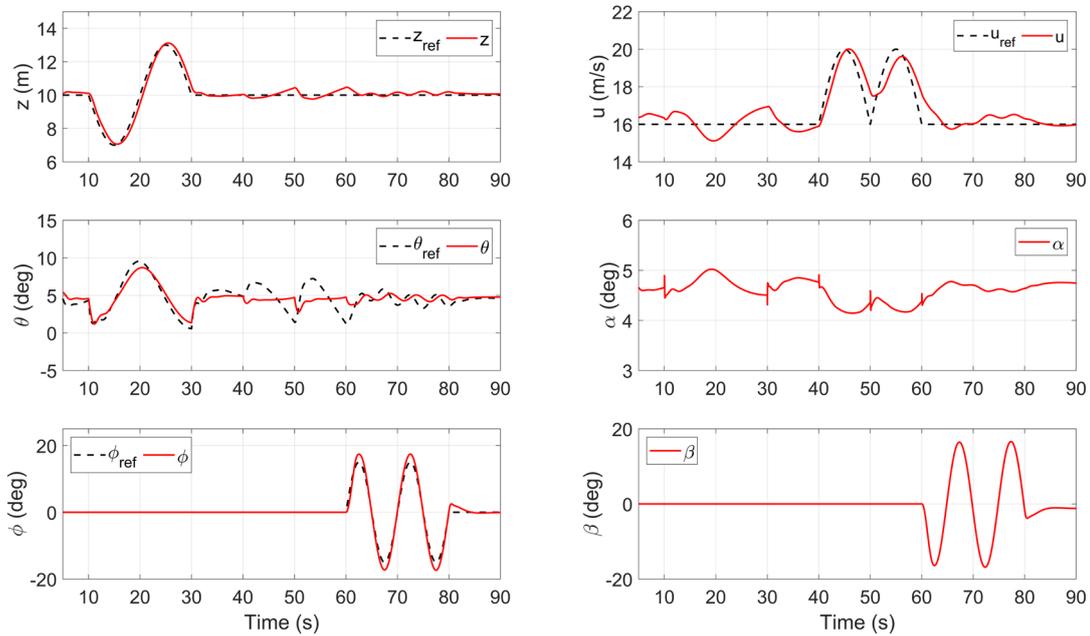


Figure 5-8 Flight simulation results of the tail-sitter UAV controlled by the longitudinal and lateral MPC controllers, which track altitude, airspeed, and roll angle.

In conclusion, this section presents the implementation of system identification and model predictive control for a tail-sitter unmanned aerial vehicle (UAV) in cruise flight. The mathematical model of longitudinal and lateral directions of the UAV are derived in the state-space form for grey-

box modeling. The least-square regression method is augmented with regularization and solved by the trust-region algorithm. Outdoor flight tests were conducted to acquire the data for system identification assisted by a signal generator module. The UAV dynamic was sufficiently excited in both longitudinal and lateral directions during the flight test. The flight data were applied to the grey box system identification, and the parameters were validated by fitting the reconstructed model to a set of flight data with a different excitation waveform. The flight controller with model predictive control was formed using the identified models for flight simulation. The results demonstrate that the system identification results are able to provide reference models for the model-based controller development of a novel-design tail-sitter UAV.

6. Transition Control

In the chapter, the operation range of the hovering control is tested in the SIL environment at the first place in chapter 6.1. With the knowledge of it, the mechanism of forward and backward transition was designed accordingly. Since the SLMPC control method can ‘adopt’ a wide range of operation conditions by consistently updating its plant model. As a result, the forward and backward transition can be handled by the cooperation of the hover control developed in chapter 4 and the cruise control setup in chapter 5. A block diagram is shown in Figure 6-1 to illustrates the mechanism of control during different flight stage. During forward transition, the control result from the left-hand-side (red) will be cut off and the controller on the right-hand-side (blue) will be activated and take over and vice versa. One of the most important issues is the timing problem and it will be explained in detail in chapter 6.2.

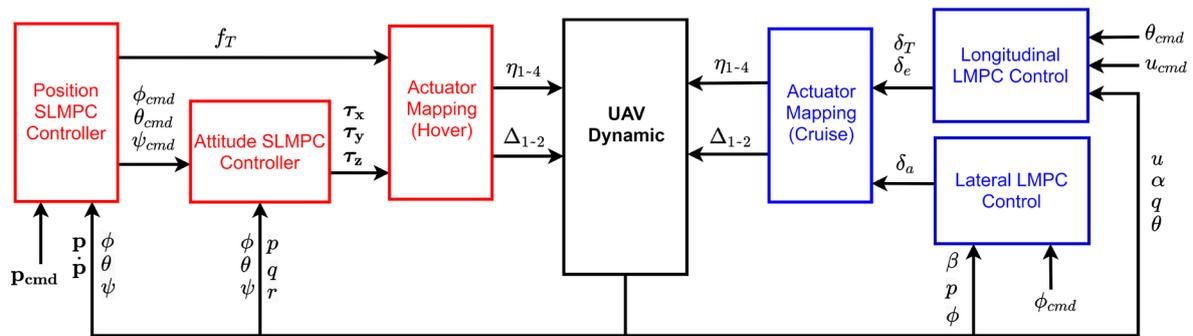


Figure 6-1 Block diagram of the control mechanism. The left (red) part handles the hover stage and the right (blue) handles the cruise period.

6.1 Operation Range of MPC controllers

The limit of the SLMPC hover control is tested in the SIL environment. Three cases of controlling the UAV in a large angle of pitch down have been conducted as shown in Figure 6-2.

Case 1 (red), the pitch down angle command is set as -50° in the attitude controller with a processing time from the 10th to the 13th seconds. There are rebounds in the theta angle when the command value is reached, and the theta angle gradually settles in the next 5 seconds. The change of altitude during the pitch down period is small since it is monitoring by the position controller. This is the beauty of a cascade structure. The external command can be given without affecting the original

performance. In the pitch down period, the vehicle has gained a forward speed of 4 m/s and have x-direction movement. The forward speed and the x-direction is excluded in the closed-loop control since they are passively controlled by the pitch angle and altitude.

Case 2 (blue) is a command of -60° pitch down in 3 seconds. In this case, it has reached the constraints of the original design of the attitude controller. In the hovering stage, the vehicle is modelled as a quadrotor and the force and moment caused by the wing are treated as disturbances. Even though the SLMPC can take the current operation condition as a new trim point and control the vehicle according to the present model, -60° is pitch or roll is considered as the limit that the quadrotor model is no longer fit. At the large pitch angle, the force to lift the vehicle is the lift force generated by the wing, rather than the thrust created by the propellers. It is unreasonable to keep using a quadrotor model at such a condition and as a result, constraints were set. From the figure, the overall performance in case 2 is as good as case 1.

Case 3 (green), the model mismatch problem is faced when a -65° pitch down command is given. The vehicle can reach the command pitch angle, but it fails to stay on it. It has lost its control in the 20th second. According to the result, the control envelope of the SLMPC can cover up to -60° in pitch. In this situation, the forward speed is around 5 m/s and it will be set as the initial condition of the cruise control.

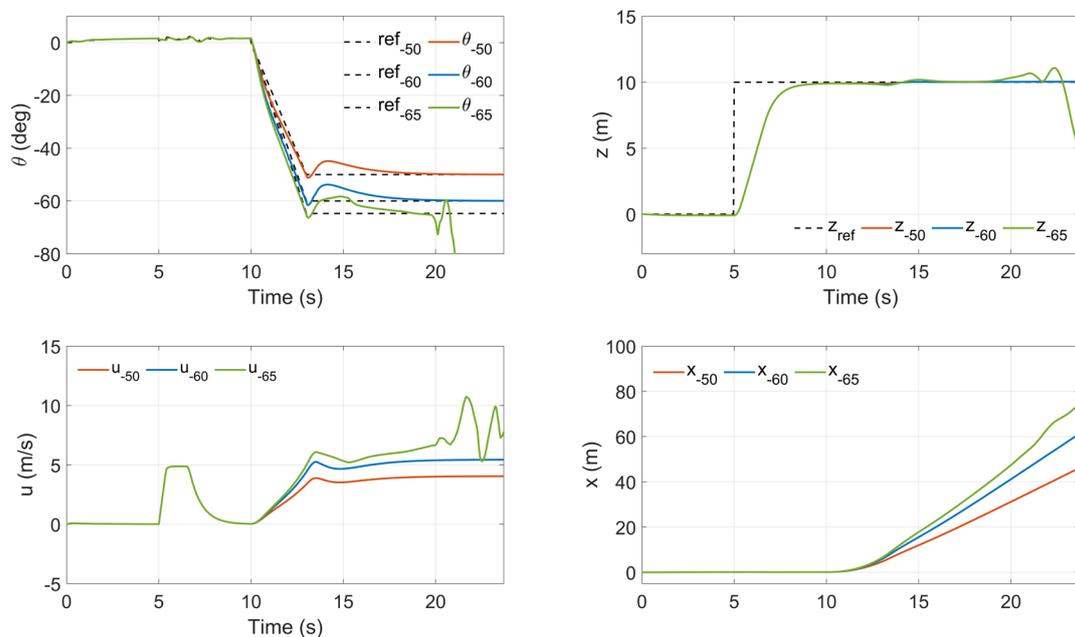


Figure 6-2 Simulation results of controlling the UAV in large pith down conditions of -50° (red), -60° (blue) and -65° (green).

From the flight data for system identification, the trim point condition of the LMPC cruise controller is -85° pitch and 16.75 m/s. The simulation results of the cruise controller started from the initial condition of -60° pitch and 5 m/s and control the vehicle to the trim point condition is shown in Figure 6-3. We can see that that takes 3 seconds to reach the trim point condition with only one-meter variation in high.

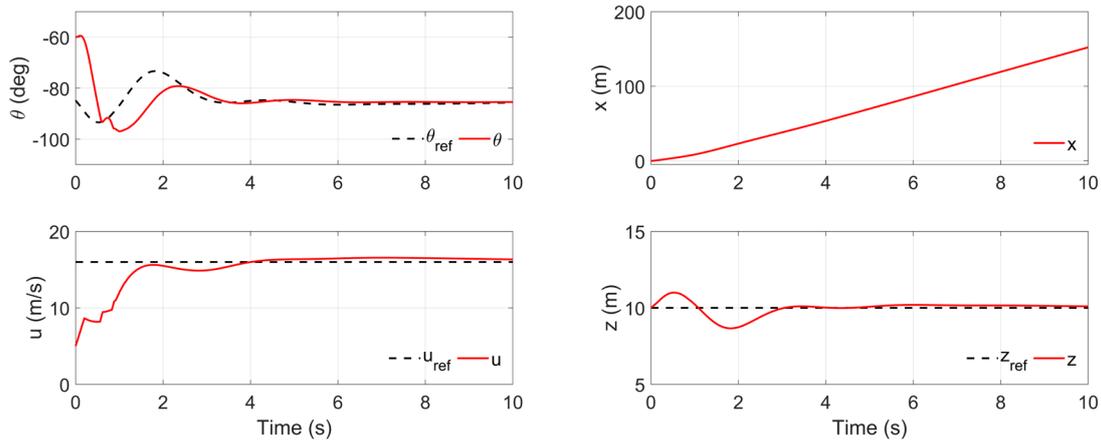


Figure 6-3 Simulation results of cruise LMPC control start from the initial condition of -60° pitch and 5 m/s forward speed.

6.2 Controller Switch Mechanism and Results

Form the previous section, it has been shown that with appropriate cooperation of the hover and cruise controllers, the vehicle can conduct transitions. Transition from hovering to cruise or cruise to hovering is all about longitudinal control and performance, as a result, the position x , z ; the forward speed u and the pitch angle θ are shown in the coming section.

A controller switch mechanism is developed to control the forward and backward transition of the vehicle and the result is shown in Figure 6-4. At the 5th second, the vehicle takes off in the hover stage and increases the height to 10 m. The forward transition starts at the 10th second, in which the vehicle is controlled to pitch down to -60° from the hover trim point condition of 0° . In the 13th second, the cruise controller is activated, and it can pick up from the current state (The red part of Figure 6-1 has switched to the blue part and the red part is deactivated) It then takes another 3 seconds to control the vehicle to the cruise trim point condition of -85° pitch and 16.75 m/s forward speed and settle down. Until this point, the forward transition is finished. During this 6 second transition period, the variation of altitude is small, and the vehicle has started to build up the x-direction movement.

In the 30th second, a step signal of pitch up is given to the cruise controller. Its reaction is fast, which the pitch angle has risen from -85° to -40° in only one second. With the sudden pitch up, the

forward speed drops from 16.75 m/s to 10 m/s and the altitude has increased from 10 m to 15 m. At this point, the situation is far away from the cruise trim point condition and it is out of the control envelope of the cruise controller. As a result, the hover control is activated at the 31st second (The blue part of Figure 6-1 has switched to the red part and the blue part is deactivated). In the next second, the hover control brings the vehicle back to the hover trim point condition of 0° pitch and zero forward. In the meantime, the altitude has increased to 20 m. It is physically impossible for the vehicle to pitch up and slow down the speed without an increase in high. From the 32nd second to the 36th second, the overshoots are degraded and settled. Up to this point, the backward transition is finished. The vehicle decreased its altitude from the 40th to 50th seconds and landed vertically.

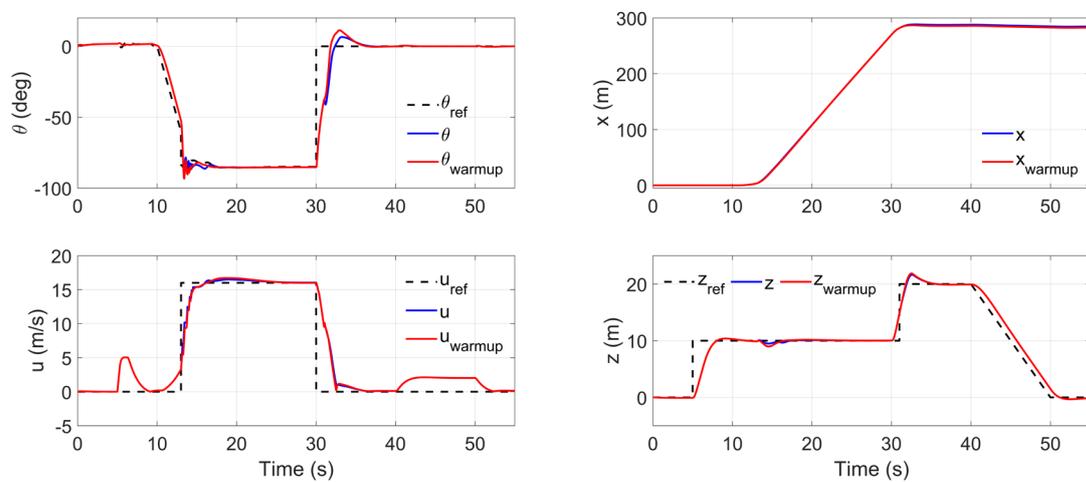


Figure 6-4 Simulation results of the whole flight process including takeoff, forward transition, cruise, backward transition and landing

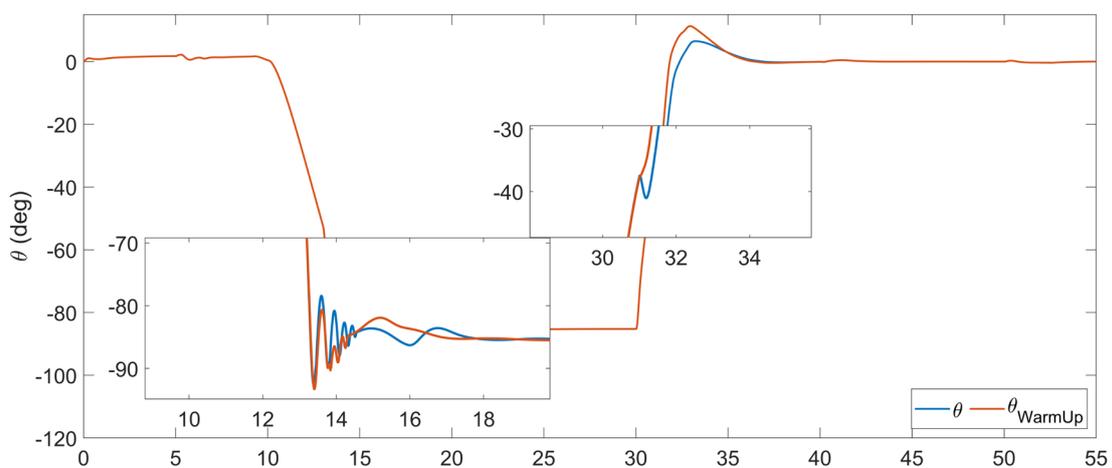


Figure 6-5 Detail illustration of the simulation result of pitch angle movement with and without controller warmup.

Figure 6-5 is the same as the upper left drawing in Figure 6-4 but showing much more detail. The controller switch mechanism illustrated above is shown in the blue line. It is obvious that between the 13th to 15th seconds when the cruise controller is just activated, there is a high-frequency oscillation with a maximum magnitude of 10° in the pitch angle. The same phenomenon has also happened when the hover control is activated in the 31st second. A sudden pitch down is very unfavorable during the pitch up period.

To suppress these control chattering, a warmup period of 10 time-step is given to the controller. The current time-step is 0.01 second and as a result, the warmup period is 0.1 second. During each transition, the controller going to be taken over the UAV control will be activated 0.1 seconds earlier. However, the control inputs generated within this 0.1 second are discharged. Take the forward transition shown in Figure 6-4 as an example. With the warmup scheme equipped, the cruise controller will be activated in 12.90 seconds and start to conduct state estimation and control input generation. Between 12.90 to 13.00 seconds, the UAV dynamic takes the control input from the hover controller. After 13.00 seconds, the hover controller deactivates, and the cruise controller takes over. By this method, a more stable and less oscillatory pitch movement is achieved in the red line of Figure 6-5. The unfavorable oscillation in the forward transition is greatly reduced and the sudden bump in backward transition is being eliminated.

In the opensource PX4 framework, during transition, open-loop control signal on throttle and pitch are given to the UAV. When both the airspeed and pitch angle reach a certain level, the cruise control takes over (ie. 10m/s and -60deg pitch), otherwise, terminate the transition and hover control takes over. The transition condition depends purely on the characteristics of the UAV. In the proposed MPC switching control, the whole transition period is under close-loop control of two MPC controllers. The transition condition depends on the overlap of control envelop of the two controllers and the feasibility and stability are generated during the whole transition period.

Figure 6-6 shows the normalized control command to actuators and control output from controllers during transition and this figure is corresponds to Figure 6-4. The blue period is under the hover SLMPC control, while the red period is under the cruise LMPC control. As we can see, at the 5th second, a command to take off and hover was given and $\eta_{(1-4)}$ as increase to 0.8~1. The suddent increase in thrust will cause increase in lift force of the wing and as a result, the airspeed (AS) increases from 0 to 0.3 during this take off period. The green line and the blue line demonstrate the control signal of motor 3 and 4. Between the 10th and 13th second, pitch down command from 0 to -60 degree was given and as we can see from the upper left diagram, there is a significant gap between these two lines, which producing the pitch down moments. During the same period, the δ command did not change,

which means there is no command to the elevator. This result is closely resembled to the hover controller design.

From the 13th to 15th second, when the cruise LPMC control first took over, very unfavourable oscillation on the $\eta_{(1-4)}$ was record from the controller. Even though it still operates and brings up the airspeed, the δ command was strongly affected, causing a vibration in the pitch angle. This is the place needs to be improved to minimize any oscillation in control command. The weight of the input increment of the cruise LMPC's cost function should be increased to penalize these dramatic change of input commands.

From the 30th to 31st second, backward transition command was given to the cruise control and we can see that the $\eta_{(1-4)}$ was dropped to zero to reduce the airspeed and the δ was at the maximum value to pitch up the vehicle. When the hover control took over, the δ was back to zero, while the motor 4 was at zero and motor 3 was at 0.4 in order to continuously provide pitch up moment. As we can see, there is no unfavourable oscillation of control command during the backward transition and the current performance is acceptable.

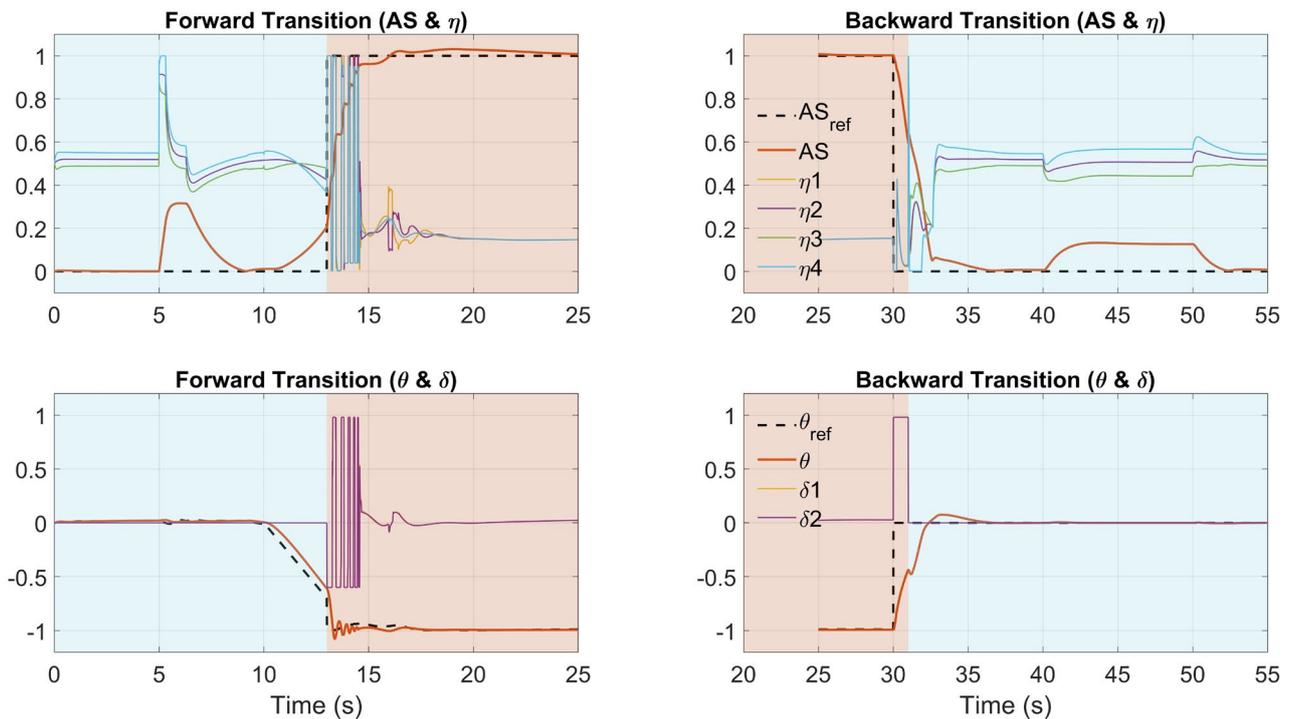


Figure 6-6 Normalized control command to actuators and control output from controllers during transition.

7. Conclusion and Future Work

7.1 Conclusion

This research is aimed to model and control a tail-sitter VTOL vehicle, which distinguishes itself from the quadrotor type as it has wing as well as the aerodynamic effect. It is also different from the fix wing aircraft as it does not has rudders. The vehicle was designed with a ‘plus’ configuration and built using commercial airframe, motors, propellers as well as a flight control board and some other electrical instruments. The vehicle’s dynamic and kinematic was described, followed by the modelling of the propulsion system. By applying the momentum theory and the method of component breakdown approach, the angle of attack and the airspeed of each wing segment were estimated. With the database of the aircraft, the lift and drag of the aerodynamic surface can be calculated at different operating conditions. Based on the sophisticated aircraft model, the MPC algorithm can be developed, started with the design of the estimated disturbance model, the prediction model and the unknown disturbance model. The objective function was set up followed by integrating the prediction model to the optimization variable and finally reach its standard 2-norm form. Constraints on each variable were also given.

The linear MPC controller with a cascade structure was developed based on the linearized dynamic model. The feasibility and performance have been tested and evaluated with SIL simulation and indoor flight experiments. A successive linearization based MPC control system is also discussed and developed, which emphasizes the capability of wind disturbance rejection. Measures of time-varying weight and the velocity reference are taken to improve performance. The simulation and indoor flight test for disturbance rejection tasks compare a traditional PID controller, a linear MPC controller, and an SLMPC controller. The results show that the SLMPC has more precise hover position control in all three axes.

To model the novel-design tail-sitter UAV, system identification has been conducted. Grey box longitudinal and lateral models have been developed in state-space forms according to the dynamics and kinematics of the UAV. The system identification was set up as the least-squares regression problem and augmented with regulation. The importance of each unknown variable can be separately adjusted. To solve the problem, the trust-region algorithm was taken to get the optimal solution. A signal generator module has been designed and set up to feed specific control commands to a flying UAV. This training line method can excite most of the dynamic of the UAV while ensuring the safety of the UAV. Two sections within a flight log have been selected for system identification and

validation. The result has shown that both identified models can capture most of the characteristics in the validation data. To further examine the identified models, they have been implemented in two MPC controllers for longitudinal and lateral controls of the UAV in cruise flight under a simulation environment. The results have shown that the MPC controllers can precisely control the UAV. The identified models are demonstrated to be sufficiently accurate. In the next step, outdoor flight tests of this UAV will be carried out for model validation and controller development.

The hover control and the cruise control were then combined in the simulation platform. A mechanism of the controller switch was designed to achieve control over the hover-transition-cruise stages. The simulation result has shown that the MPC controller set can control the tail-sitter vehicle in all operation stages and meet the requirements.

7.2 Future work

At the next step, the cruise MPC control will be implemented onto the tail-sitter UAV for outdoor flight tests. In the primary trial, The UAV will be take-off, transition and cruise by PID control from PX4. While cruise, it will be switched to the off-board mode and MPC cruise control will take over. The Up board (a lightweight onboard computer) will runs the 4 states longitudinal MPC and 3 states lateral MPC. The aircraft will stay in the cruise stage at around the trim point condition for 3 seconds (hand off the remote control) or more to validate the auto stabilization function. The challenges include replacing the PX4 PID attitude control by the MPC cruise control without knowing the performance of the auto stabilization function.

In the secondary trial, we will try to give maneuvers to the vehicle from the remote control while it is under MPC cruise control. Because the MPCs are designed to take input signals between 0-1 with a logic same as the PX4 attitude control, there is no need to do any modification for the piolet to control it. In this trial, we are aiming to check and record the performance of MPC control and compare it to the PID from PX4. The challenges are focus on parameter tuning while the aircraft is airborne. We have to figure out a set of communication solutions between the onboard computer and the ground control station in order to facilitate the real-time parameter tuning. Otherwise, it is impossible for us to change a single controller parameter every time after the aircraft is landed.

In the third trial, the performance of MPC hover control in -60 deg condition will be tested at the first stage. Because the forward transition starts from this condition. It necessary to validate the controllability of the MPC hover controller in real flight tests to see does it has the same performance

as the simulation result shown. With the same control allocation as the PX4 logic, the UAV can takeoff under PX4, the pilot can switch to the off-board mode and the MPC hover controller will take over. The UAV will climb up until at least 50-meter-high and pitch down to a large angle. Then it can be switched to transition and the MPC cruise controller takes over. Whenever there is an unexpected maneuver shown by the UAV, it will be switched back to the PX4 control immediately and this is the major challenge. The pilot commands the UAV to transition under MPC control, but how about the PX4 control? The PX4 control does not know that the vehicle has already transitioned into the cruise stage. If the pilot suddenly leaves the off-board mode, the hover PID control from the PX4 will take over, rather than the cruise PID control and the consequence is unknown. This transition flight test is full of risk. The threat of vehicle crushing is always amounting us. This challenge has to be overcome before the conduction of the experiment. Meanwhile, at this stage, risk in other aspects may not be identified yet and much more effort should be put into the safety measures for risk prevention.

For a longer stage, the nonlinear MPC can be studied and attempted afterward. It can be tried to control the tail-sitter vehicle without the use of a number of controllers. The objective function will be defined according to need and solve by some higher performance QP solvers. The control algorithm can be tested and modified by simulation until an acceptable performance. Indoor and outdoor flight experiments can be conducted and compare the results to the current MPC control method.

References

- [1] B. Li, W. Zhou, J. Sun, C. Wen, and C. Chen, "Model Predictive Control for Path Tracking of a VTOL Tail-sitter UAV in an HIL Simulation Environment," in *AIAA Modeling and Simulation Technologies Conference*(AIAA SciTech Forum: American Institute of Aeronautics and Astronautics, 2018.
- [2] B. Li, W. Zhou, J. Sun, C.-Y. Wen, and C.-K. Chen, "Development of Model Predictive Controller for a Tail-Sitter VTOL UAV in Hover Flight," *Sensors*, vol. 18, p. 2859, 08/30 2018.
- [3] W. Zhou, B. Li, J. Sun, C.-Y. Wen, and C.-K. Chen, "Position control of a tail-sitter UAV using successive linearization based model predictive control," *Control Engineering Practice*, vol. 91, p. 104125, 2019/10/01/ 2019.
- [4] B. Li, J. Sun, W. Zhou, C. Y. Wen, K. H. Low, and C. K. Chen, "Transition Optimization for a VTOL Tail-Sitter UAV," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2534-2545, 2020.
- [5] W. Zhou, S. Chen, C.-W. Chang, C.-Y. Wen, C.-K. Chen, and B. Li, "System Identification and Control for a Tail-Sitter Unmanned Aerial Vehicle in the Cruise Flight," *IEEE Access*, vol. 8, 12/09 2020.
- [6] B. McCormick, *Aerodynamics of V/STOL Flight* (no. 3). New York 10003: Academic Press Inc., 1967.
- [7] J. Gertler, *V-22 Osprey Tilt-Rotor Aircraft Background and Issues for Congress*. Congressional Research Service, 2011.
- [8] S. M. B. Holder, *Straight Up: A History of Vertical Flight* (Schiffer Military/Aviation History). Schiffer Publishing, 2000.
- [9] N. Tesla, "Tesla Patent 1,655,114 Apparatus for Aerial Transportation," 1927.
- [10] "Hiller X-18 Tilt-wing VTOL cargo transport experiment," ed, 1961.
- [11] "Russian drone tiltrotor RHV-35," RHV-35, Ed., ed, 2016.
- [12] "MLB V-Bat VTOL Unmanned Aerial Vehicle (UAV)," ed, 2014.
- [13] "CW-30 JOUAV," ed, 2015.
- [14] "WingtraOne mapping drone," ed, 2020.
- [15] "Alfa Pixhawk 2 Tail-sitter VTOL ", ed, 2020.
- [16] "Bell APT 70," ed, 2020.
- [17] R. Suzuki *et al.*, "Teleoperation of a tail-sitter VTOL UAV," presented at the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.
- [18] "A Class of Modular Aerial Robots," presented at the *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, 2011.
- [19] R. N. Francesco Forte, Andrea Serrani, Lorenzo Marconi, "Control of Modular Aerial Robots: Combining Under- and Fully-Actuated Behaviors," presented at the *51st IEEE Conference on Decision and Control*, Maui, Hawaii, USA, 2012.
- [20] P. Sinha, P. Esden-Tempski, C. A. Forrette, J. K. Gibboney, and G. M. Horn, "Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions," in *2012 IEEE Aerospace Conference*, pp 1-17, 2012.
- [21] "Paparazzi UAV," in *Paparazzi Project*, ed, 2016.
- [22] X. Wang, Z. Chen, and Z. Yuan, "Modeling and control of an agile tail-sitter aircraft," *Journal of the Franklin Institute*, vol. 352, no. 12, pp. 5437-5472, 2015.
- [23] E. Lavretsky and K. A. Wise, *Robust and Adaptive Control with Aerospace Applications* (Advanced Textbooks in Control and Signal Processing). Springer, 2012.
- [24] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd Edition* (Modern Design Techniques). Wiley-Interscience, 2015.
- [25] J. Arthur E. Bryson, "New concepts in control theory 1959-1984 - Dryden Lecture for 1984," in *22nd Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 1984.
- [26] U.-L. Ly, A. E. Bryson, and R. H. Cannon, "Design of low-order compensators using parameter optimization," *Automatica*, vol. 21, no. 3, pp. 315-318, 1985.
- [27] Dagfinn Gangsaas, Kevin R. Bruce, J. D. Blight, and U. L. Ly, "Application of Modern Synthesis to Aircraft Control," *IEEE Transactions on Automatic Control* vol. AC31, no. 11, pp. 995-1014, 1986.
- [28] J. Doyle and G. Stein, "Multivariable feedback design: Concepts for a classical/modern synthesis," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 4-16, 1981.
- [29] S. D. Ishmael and C. C. R. McMonagle, "AFTI/F-16 Flight Test Results and Lessons," Nation Aeronautics and Space Administration, 1983.
- [30] H. P. W. J. Y. A. Kezer, *Design of model-reference adaptive control systems for aircraft (R-164)*. Massachusetts Institute of Technology, Instrumentation Laboratory : Jackson & Moreland, 1958.

- [31] H. P. Whitaker, Osburn, P.V., Kezer A, "New Developments in the Design of Model Reference Adaptive Control Systems," *Institute of the Aerospace Sciences, Easton, PA*, pp. 39-61, 1961.
- [32] B. Shackcloth and R. L. Butchart, "Synthesis of Model Reference Adaptive Systems by Liapunov's Second Method," in *Theory of Self-Adaptive Control Systems: Proceedings of the Second IFAC Symposium on the Theory of Self-Adaptive Control Systems*, National Physical Laboratory Teddington, England, P. H. Hammond, Ed. Boston, MA: Springer US, pp. 145-152, 1966.
- [33] P. Parks, "Liapunov redesign of model reference adaptive control systems," *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 362-367, 1966.
- [34] A. M. A. Kumpati S. Narendra, *Stable Adaptive Systems*. Dover Publications, New York, 2005.
- [35] B. F. Petros Ioannou, *Adaptive Control Tutorial* (Advances in Design and Control). SIAM, Society for Industrial and Applied Mathematics, 2006.
- [36] W. L. Jean-Jacques Slotine, *APPLIED NONLINEAR CONTROL*. Englewood Cliffs, New Jersey: Prentice Hall, 1991.
- [37] I. K. Miroslav Krstic, Petar V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: John Wiley and Sons, Inc., 1995.
- [38] M. Athans, "On the LQG Problem," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 528-528, 1971.
- [39] C. B. A. Eduardo F. Camacho, *Model Predictive Control* (Advanced Textbooks in Control and Signal Processing). Springer-Verlag London, 2007.
- [40] C. B. A. Eduardo F. Camacho, *Model Predictive Control*, 2 ed. (Advanced Textbooks in Control and Signal Processing). Springer-Verlag London, p. 405, 2007.
- [41] M. Simulink. Explicit MPC [Online]. Available: <https://www.mathworks.com/help/mpc/ug/explicit-mpc.html?requestedDomain=www.mathworks.com>
- [42] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Model predictive quadrotor control: attitude, altitude and position experimental studies," *IET Control Theory & Applications*, vol. 6, no. 12, pp. 1812-1827, 2012.
- [43] N. Poursafar, H. D. Taghirad, and M. Haeri, "Model predictive control of non-linear discrete time systems: a linear matrix inequality approach," *IET Control Theory & Applications*, vol. 4, no. 10, pp. 1922-1932, 2010.
- [44] Y. Liu, J. M. Montenbruck, P. Stegagno, F. Allgöwer, and A. Zell, "A robust nonlinear controller for nontrivial quadrotor maneuvers: Approach and verification," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5410-5416, 2015.
- [45] M. Bangura and R. Mahony, "Real-time Model Predictive Control for Quadrotors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11773-11780, 2014.
- [46] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520-2525, 2011.
- [47] C. Liu, H. Lu, and W. H. Chen, "An explicit MPC for quadrotor trajectory tracking," in *2015 34th Chinese Control Conference (CCC)*, pp. 4055-4060, 2015.
- [48] P. Anna and K. Erdal, "An MPC-based position controller for a tilt-rotor tricopter VTOL UAV," *Optimal Control Applications and Methods*, vol. 39, no. 1, pp. 343-356, 2018.
- [49] M. Abdolhosseini, Y. M. Zhang, and C. A. Rabbath, "An Efficient Model Predictive Control Scheme for an Unmanned Quadrotor Helicopter," *Journal of Intelligent & Robotic Systems*, journal article vol. 70, no. 1, pp. 27-38, 2013.
- [50] M. B. Mina Kamel, Roland Siegwart, "Linear vs Nonlinear MPC for Trajectory Tracking Applied to RotaryWing Micro Aerial Vehicles," *arXiv*, p. arXiv:1611.09240, Cornell University, 2016.
- [51] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Robust Model Predictive Flight Control of Unmanned Rotorcrafts," *Journal of Intelligent & Robotic Systems*, journal article vol. 81, no. 3, pp. 443-469, 2016.
- [52] W. Dong, G. Gu, X. Zhu, and H. Ding, *High-Performance Trajectory Tracking Control of a Quadrotor with Disturbance Observer*. Sensors and Actuators A: Physical: ELSEVIER, 2014.
- [53] F. Zhao and Y. P. Gupta, "A simplified predictive control algorithm for disturbance rejection," *ISA Transactions*, vol. 44, no. 2, pp. 187-198, 2005.
- [54] K. Alexis, A. Tzes, and G. Nikolakopoulos, "Model predictive quadrotor control: attitude, altitude and position experimental studies," *IET Control Theory & Applications*, vol. 6, no. 12, pp. 1812-1827, 2012.
- [55] P. N. Chikasha and C. Dube, "Adaptive Model Predictive Control of a Quadrotor," *IFAC-PapersOnLine*, vol. 50, no. 2, pp. 157-162, 2017.
- [56] B. Zhu and Z. Zuo, "Adaptive model predictive control for set-point tracking of a single-DOF flapping-wing unmanned aerial vehicle," in *2017 13th IEEE International Conference on Control & Automation (ICCA)*, pp. 861-866, 2017.
- [57] S. Zhao, X. Wang, D. Zhang, and L. Shen, "Model Predictive Control Based Integral Line-of-Sight Curved Path Following for Unmanned Aerial Vehicle," in *AIAA Guidance, Navigation, and Control Conference*(AIAA SciTech Forum: American Institute of Aeronautics and Astronautics, 2017.

- [58] W. Dong, G.-Y. Gu, X. Zhu, and H. Ding, "High-performance trajectory tracking control of a quadrotor with disturbance observer," *Sensors and Actuators A: Physical*, vol. 211, pp. 67-77, 2014.
- [59] C. Liu, W.-H. Chen, and J. Andrews, "Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers," *Control Engineering Practice*, vol. 20, no. 3, pp. 258-268, 2012.
- [60] D. Ma, Y. Xia, T. Li, and K. Chang, "Active disturbance rejection and predictive control strategy for a quadrotor helicopter," *IET Control Theory & Applications*, vol. 10, no. 17, pp. 2213-2222, 2016.
- [61] Y. Lindberg, "A Comparison Between MPC and PID Controllers for Education and Steam Reformers," M.Sc. Department of Signals and Systems, Chalmers University of Technology, Goteborg, Sweden, 2014.
- [62] K. Gryte, R. Hann, M. Alam, J. Roháč, T. A. Johansen, and T. I. Fossen, "Aerodynamic modeling of the Skywalker X8 Fixed-Wing Unmanned Aerial Vehicle," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 826-835, 2018.
- [63] N. Hoffer, C. Coopmans, A. Jensen, and Y. Chen, "A Survey and Categorization of Small Low-Cost Unmanned Aerial Vehicle System Identification," *Journal of Intelligent & Robotic Systems*, vol. 74, 2014.
- [64] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2677-2682, 2010.
- [65] M. S. Gandhi, L. Whitcher, E. Theodorou, and E. N. Johnson, "Practical System Identification for Small VTOL Unmanned Aerial Vehicle," AIAA Scitech 2019 Forum, 2019.
- [66] E. A. Morelli, "Real-Time Parameter Estimation in the Frequency Domain," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 812-818, 2000.
- [67] Y. Yu, P. Tang, T. Song, and D. Lin, "A two-step method for system identification of low-cost quadrotor," *Aerospace Science and Technology*, vol. 96, p. 105551, 2019.
- [68] V. Klein and E. A. Morelli, *Aircraft System Identification: Theory And Practice* (AIAA education series). American Institute of Aeronautics and Astronautics, pp. 306-322, 2006.
- [69] A. Dorobantu, A. Murch, B. Mettler, and G. Balas, "System Identification for Small, Low-Cost, Fixed-Wing Unmanned Aircraft," *Journal of Aircraft*, vol. 50, pp. 1117-1130, 2013.
- [70] S. Chumalee and J. F. Whidborne, "Unmanned aerial vehicle aerodynamic model identification from a racetrack manoeuvre," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 224, no. 7, pp. 831-842, 2010.
- [71] K. Y. W. Scheper, G. Chowdhary, and E. N. Johnson, "Aerodynamic system identification of fixed-wing UAV," presented at the AIAA Atmospheric Flight Mechanics (AFM) Conference, Boston, MA, United States, 2013.
- [72] M. Gandhi, L. Whitcher, E. Johnson, and E. Theodorou, "Implementing system identification for a vtol unmanned aerial vehicle," presented at the AIAA Scitech 2019 Forum, San Diego, United States, 2019.
- [73] A. B. F. Borrelli, M. Morari, *Predictive Control for linear and hybrid systems* (Receding Horizon Control), p. 427, 2014.
- [74] J. Maciejowski, *Predictive Control with Constrains*. Prentice Hall, p. 346, 2002.
- [75] C. Schmid, Biegler, L. T., "Quadratic programming methods for reduced hessian SQP," *Computers & Chemical Engineering*, vol. 18, no. 9, pp. 817-832, 1994.
- [76] M. Ławryńczuk, *Computationally Efficient Model Predictive Control Algorithms*, 1 ed. (Studies in Systems, Decision and Control). Springer International Publishing, pp. XXIV, 316, 2014.
- [77] D. P. Raymer, "AIRCRAFT DESIGN ; A Conceptual Approach BY DANIEL P RAYMER," 4th ed. American Institute of Aeronautics and Astronautics, vol. 2, no. 4, 2006.
- [78] S. Shkarayev, J. M. Moschetta, and B. Bataille, "Aerodynamic Design of Micro Air Vehicles for Vertical Flight," *Journal of Aircraft*, vol. 45, no. 5, pp. 1715-1724, 2008.
- [79] X. Penghui, "Aerodynamics of VTOL UAV," MSc Thesis, Mechanical Engineering, The Hong Kong Polytechnic University, July, 2017.
- [80] J. Sun, B. Li, L. Shen, C.-K. Chen, and C.-Y. Wen, "Dynamic Modeling and Hardware-In-Loop Simulation for a Tail-Sitter Unmanned Aerial Vehicle in Hovering Flight," in *AIAA Modeling and Simulation Technologies Conference*(AIAA SciTech Forum: American Institute of Aeronautics and Astronautics, 2017).
- [81] Matlab Simulink. Choose Sample Time and Horizons, Controller Creation, MPC Design [Online]. Available: <https://www.mathworks.com/help/mpc/ug/choosing-sample-time-and-horizons.html>
- [82] Matlab Simulink. Tune Weights, Controller Creation, MPC Design, Model Predictive Control Toolbox [Online]. Available: <https://www.mathworks.com/help/mpc/ug/tuning-weights.html>
- [83] Matlab Simulink. Control of an Inverted Pendulum on a Cart, Model Predictive Control Toolbox Examples, Documentation [Online]. Available: <https://www.mathworks.com/help/mpc/examples/control-of-an-inverted-pendulum-on-a-cart.html>

- [84] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, "Model Predictive Control in Aerospace Systems: Current State and Opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541-1566, 2017.
- [85] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967-2986, 2014.
- [86] J. Sun, B. Li, C. Wen, and C. Chen, "Model-Aided Wind Estimation Method for a Tail-Sitter Aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 2, pp. 1262-1278, 2020.
- [87] H. Li, C. Li, H. Li, Y. Li, and Z. Xing, "An Integrated Altitude Control Design for a Tail-Sitter UAV Equipped With Turbine Engines," *IEEE Access*, vol. 5, pp. 10941-10952, 2017.
- [88] J. Zhong, B. Song, Y. Li, and J. Xuan, "L1 Adaptive Control of a Dual-Rotor Tail-Sitter Unmanned Aerial Vehicle With Input Constraints During Hover Flight," *IEEE Access*, vol. 7, pp. 51312-51328, 2019.
- [89] Z. Li, W. Zhou, H. Liu, L. Zhang, and Z. Zuo, "Nonlinear Robust Flight Mode Transition Control for Tail-Sitter Aircraft," *IEEE Access*, vol. 6, pp. 65909-65921, 2018.
- [90] R. C. Nelson, *Flight stability and automatic control*, 2nd ed. New York: McGraw-Hill Education, pp. 104-110, 1998.
- [91] Y. Peixin and S. Baohuai, "Least Square Regressions with Coefficient Regularization," in *2011 Fourth International Conference on Intelligent Computation Technology and Automation*, vol. 1, pp. 167-170, 2011.
- [92] A. K. M. E. Saleh, M. Arashi, and B. M. G. Kibria, *Theory of Ridge Regression Estimation with Applications*, 1st ed. (Wiley Series in Probability and Statistics). Wiley, pp. 143-144, 2019.
- [93] Y. Yuan, "A Review of Trust Region Algorithms for Optimization," *ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics*, 1999.
- [94] E. A. Morelli and V. Klein, "Application of System Identification to Aircraft at NASA Langley Research Center," *Journal of Aircraft*, vol. 42, no. 1, pp. 12-25, 2005.
- [95] E. A. Morelli, "Flight Test of Optimal Inputs and Comparison with Conventional Inputs," *Journal of Aircraft*, vol. 36, no. 2, pp. 389-397, 1999.
- [96] N. V. Hoffer, "System Identification of a Small Low-Cost Unmanned Aerial Vehicle Using Flight Data From Low-Cost Sensors," Master of Science, Mechanical Engineering, Utah State University, 2014.
- [97] S. Carnduff, "System Identification of Unmanned Aerial Vehicles," Ph.D., Department of Aerospace Sciences, Cranfield University, 2008.
- [98] S. Saderla, Y. Kim, and A. K. Ghosh, "Online system identification of mini cropped delta UAVs using flight test methods," *Aerospace Science and Technology*, vol. 80, 2018.