CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Robust Model Predictive Control for Autonomous Landing of Fixed Wing UAV

by

Zohaib Latif

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

in the

Faculty of Engineering

Department of Electrical Engineering

2023

Robust Model Predictive Control for Autonomous Landing of Fixed Wing UAV

By

Zohaib Latif

(DEE183004)

Dr. Shen Yin, Professor
Norwegin University of Science and Technology, Norway
(Foreign Evaluator 1)

Dr. Fernando A. Fontes, Associate Professor
University of Porto, Portugal
(Foreign Evaluator 2)

Dr. Aamer Iqbal Bhatti
(Thesis Supervisor)

Dr. Amir Shahzad
(Thesis Co-Supervisor)

Dr. Noor Muhammad Khan
(Head, Department of Electrical Engineering)

Dr. Imtiaz Ahmad Taj
(Dean, Faculty of Engineering)

DEPARTMENT OF ELECTRICAL ENGINEERING
CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
ISLAMABAD
2023

I dedicate my dissertation work to my beloved brothers, the late Sohail Latif (1984-2005) and the late Shoaib Latif (1986-2006), who have gone forever away from our loving eyes and left a void never to be filled in our lives. Though your life was short, I will make sure your memory lives on as long as I will live. I love you and miss you both beyond words. May Allah (SWT) grant you Jannah Firdaws, Amen.

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD
Expressway, Kahuta Road, Zone-V, Islamabad
Phone:+92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: https://www.cust.edu.pk

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled "**Robust Model Predictive Control for Autonomous Landing of Fixed Wing UAV**" was conducted under the supervision of **Dr. Aamer Iqbal Bhatti**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Electrical Engineering, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Electrical Engineering**. The open defence of the thesis was conducted on **June 15, 2023**.

Student Name : Zohaib Latif (DEE183004)

The Examining Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a)	External Examiner 1:	Dr. Abdul Qayyum Khan, Professor PIEAS, Islamabad	
(b)	External Examiner 2:	Dr. Iftikhar Ahmad Rana, Associate Professor SEECS NUST, Islamabad	
(c)	Internal Examiner :	Dr. Fazal ur Rehman Professor CUST, Islamabad	Fazal Rehman

Supervisor Name :	Dr. Aamer Iqbal Bhatti Professor CUST, Islamabad	MAIBhatti
Name of HoD :	Dr. Noor Muhammad Khan Professor CUST, Islamabad	
Name of Dean :	Dr. Imtiaz Ahmad Taj Professor CUST, Islamabad	

AUTHOR'S DECLARATION

I, **Zohaib Latif (Registration No. DEE183004)**, hereby state that my PhD thesis entitled, '**Robust Model Predictive Control for Autonomous Landing of Fixed Wing UAV**' is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/ world.

At any time, if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my PhD Degree.

(Zohaib Latif)

Dated: 15 June, 2023

Registration No : DEE183004

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "**Robust Model Predictive Control for Autonomous Landing of Fixed Wing UAV**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD Degree, the University reserves the right to withdraw/ revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/ University Website on which names of students are placed who submitted plagiarized thesis.

(Zohaib Latif)

Registration No : DEE183004

Dated: 15 June, 2023

List of Publications

It is certified that following publication(s) have been made out of the research work that has been carried out for this thesis:-

1. **Z. Latif**, A. Shahzad, A.I. Bhatti, J.F. Whidborne and R. Samar, "Autonomous Landing of an UAV Using H_∞ Based Model Predictive Control," *Drones*, no. 6, vol. 12, pp. 416, 2022.
2. **Z. Latif**, A. Shahzad, R. Samar and A.I. Bhatti , "Lateral Parameter-Varying Modelling and Control of a UAV on-Ground," *4th IFAC Workshop on Linear Parameter Varying Systems (LPVS) Milan, Italy*, no. 8, vol. 54, pp. 130-135, 2021.

(Zohaib Latif)

Registration No:DEE183004

Acknowledgement

In the name of Allah, The Most Beneficent, The Most Merciful. First and foremost, I would like to express gratitude to Allah SWT, the Almighty God, for the blessing, kindness, and inspiration in lending me to pursue my doctoral studies.

I am indebted to my teachers whose teachings have brought me to this stage. Certainly, it is a pleasant obligation to thank my supervisor Prof. Dr. Aamer Iqbal Bhatti, for his valuable guidance, strong encouragement, and kind support towards my study and research. I have learned not only from his insight, deep technical knowledge, and practical experience but many things from him. It has been a real honor to work under his supervision and attend lectures and talks. I also like to express my gratitude to my co-supervisor, Dr. Amir Shahzad, whose keen observance and directives helped me to develop and enhance my skills and complete my research work on time. I would especially like to thank Prof. Dr. Raza Samar for his valuable guidance, encouragement, and kind support towards my research and study. I am particularly grateful to Prof. Dr. James Ferris Whidborne for his kind supervision and support during my visit to Cranfield University, UK. I am grateful to all Control and Signal Processing Research (CASPR) group members whose valuable company always kept me focused and motivated. Especially, I recognize the valuable support from Dr. Raheel Anjum, Usman Zafar, Farhan Hanif, Faheem Manzoor, Abrar Hashmi, Yasir Naeem, Sidra Ghayour Bhatti, Faheem Gulzar, and Ahmed Mehmood.

I also acknowledge the intimate support of my parents throughout my educational career. Their prayers and moral and financial support always motivated me toward my goal. I also appreciate my spouse and children for their cooperation, love, and support and for providing a stress-free atmosphere during my doctorate tenure. Moreover, I am thankful to my sisters and cousin brother Jaweed Aslam for their constant support and encouragement throughout my graduate studies. I am also grateful to my brothers-in-law, Mohammad Harouf and Wasim Ahmed, for their continuous support and care during the UK trip.

Last but not least, I would like to thank the Centre of Aeronautics, Cranfield University, UK, for providing me an opportunity to work there. I am also thankful to the Higher Education Commission (HEC) for financial support under International Research Support Initiative Program (IRSIP) during my stay at Cranfield University.

(Zohaib Latif)

Abstract

Over the past few decades, Unmanned Air Vehicles (UAVs) have gained popularity in various applications, from military operations to public safety. Nowadays, UAVs can perform commercial tasks such as image and data acquisition in disaster areas, communication delays, traffic surveillance, map building, search and rescue, and so on. With the rise in popularity, the number of incidents involving UAVs has increased drastically due to inexperienced operators. Most UAVs have autonomous take-off and cruise but limited autonomous landing capabilities due to reliability issues and high risks. In the autonomous landing phase, the UAV performs many sensitive and vital tasks in different environmental conditions. Therefore, UAVs must maintain their performance in the presence of external disturbances (e.g., wind, fog).

The landing task involves airborne (glide and flare maneuver) and ground taxi phases. In glide maneuver, the UAV must descend along the predefined straight-line path in the longitudinal plane with a fixed negative flight path angle between -4^o and -3^o toward the runway. When the UAV reaches around $25 - 30$ m altitude, the flare maneuverability is initiated. In this phase, the UAV is required to reduce the descent rate and follow a curved path. It is necessary to bring the flight path angle near zero for a smooth touchdown and the minimum impact on the landing gears. After the touchdown, the ground taxi phase begins, in which the UAV needs to steer the centerline of the runway to stop.

In this work, first, the H_∞ based Model Predictive Control (MPC) is proposed for the airborne landing phase in the presence of wind disturbance. The UAV encounters the wind at the start of the glide slope, and its effects last until the flare ends. After the touchdown, the ground effects start contributing to the dynamics of the UAV. As longitudinal velocity changes during the taxi phase, UAV dynamics also change. At high velocities, aerodynamics contribute more, and ground dynamics have more effects at low velocities. Due to this coupling between aerodynamics and tire-ground friction forces, dynamics and control on the ground are more complex. To deal with these issues, a structured Linear Parameter

Varying (LPV) model and H_∞ based LPV-MPC is proposed for the taxi phase. The cost function matrices for the MPC are designed by solving the H_∞ based inverse optimal problem. The designed MPC inherits the small-signal properties (stability margin and closed-loop performance) of the H_∞ controller when the constraints are inactive (i.e., the perturbation around the equilibrium point due to which the system remains within saturation limits). This control law is successfully evaluated in the non-linear simulation environment, and the results are compared with the benchmark controllers. Moreover, the quantitative comparison has been done by taking the normalized root mean square error (NRMSE) between the outputs of the taxi phase that are taken by both techniques (H_∞ and H_∞ based LPV-MPC). The NRMSEs between the outputs are 1, 1, and 0.9999 (1 for 100% matching and 0 for no matching) for cross distance (p_y), yaw (ψ), and yaw rate (R) respectively.

Contents

List of Figures

List of Tables

Abbreviations

ACL	Autonomous Carrier Landing
ANN	Artificial Neural Network
ARE	Algebraic Riccati Equation
ATOL	Autonomous Take-off and Landing
CG	Center of Gravity
CMAC	Cerebellar Model Articulation Controller
CPSO	Chaotic Particle Swarm Optimization
DoF	Degree of Freedom
GPS	Global Positioning System
HIL	Hardware in Loop
INS	Inertial Navigation System
IPM	Interior Point Method
KF	Kalman Filter
KKT	Karush Kuhn Tucker
LiDAR	Light Detection and Ranging
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LSDP	Loop Shaping Design Procedure
LFT	Linear Friction Transform
MPC	Model Predictive Control
NED	North East Down

NRMES	Normalised Root Mean Square Error
PDIPM	Primal Dual Interior Point Method
PI	Proportional Integral
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization
SMC	Sliding Mode Control
TDC	Tracking Deviation Correction
UAV	Unmanned Air Vehicle
VTOL	Vertical Take-off and Landing

Symbols

α	Angle of attack
l_m	Distance of main wheel from CG
l_n	Distance of nose wheel from CG
l_l	Distance of left wheel from the center of main wheel axle
l_r	Distance of right wheel from the center of main wheel axle
m	Mass
$\bar{c}$	Mean aerodynamics chord
J_y	Moment of inertia about y-axis
J_z	Moment of inertia about z-axis
θ	Pitch angle
γ_r	Reference glide path angle
ϕ	Roll angle
S	Wing area
b	Wing span
ψ	Yaw angle
τ	Time Constant
ρ	Air mass density

Chapter 1

Introduction

Unmanned aerial vehicles (UAVs), commonly known as drones, are aircraft without onboard human pilots. The UAV system includes a ground controller and a communication system. The UAVs may operate remotely by a human pilot or with autonomous flight control systems.

1.1 Background

Britain and the USA developed the first pilotless vehicles during the First World War. In March 1917, the British Aerial Target was tested, while the U.S. Kettering Bug, a small radio-controlled aircraft, was flown for the first time in October 1918. Despite their promising performances during flight tests, neither was used during the war operation [1].

The UAVs developed through the 20th century were originally for military purposes and were too "dirty, dull or dangerous" for humans. However, they become essential to most militaries by the 21th century. Many UAVs were developed and tested during the interwar period. The British produced radio-controlled aircraft in 1935 for training purposes. At this time, the term 'drone' was coined from the name of one of these models, the DH.82B Queen Bee [2]. The United States also manufactured radio-controlled drones for target practice and training.

During the Vietnam War, reconnaissance UAVs were first deployed in large numbers. The UAVs also began to be used in various roles, such as launching missiles against fixed targets, acting as decoys in combat, and dropping leaflets for psychological operations. Following the Vietnam War, UAV technology began to be explored by countries outside the United States and Britain [3]. In new models, performance and endurance were improved, along with the ability to maintain a greater height. Several models have been developed in recent years that use solar power to fuel longer flights.

The UAVs now have many functions, ranging from filming, photography, delivering goods, and monitoring the changes after natural disasters to carrying out search operations. However, their most controversial and well-known use is by the military for surveillance, reconnaissance, and targeted attacks. The United States has significantly increased its use since the 9/11 attack. They are mostly used for surveillance in terrain and areas where troops cannot travel safely. However, they are also used as weapons to attack and killing of suspected militants [4].

The academic history of UAVs involves research, development, and advancements made in UAVs technology, as well as the scientific principles underlying their design and operation. One of the earliest academic works on UAVs was published in 1970 by Dr. Paul Bikle, who wrote a book titled "Unmanned Aircraft Systems," which explored the history, technology, and potential applications of UAVs. Since then, numerous academic institutions and researchers have contributed to the development of UAVs, both in terms of hardware and software.

In the 1990s, the academic community started focusing on the development of autonomous UAVs that can operate without human intervention. The research efforts in this area were driven by the need to improve UAV reliability and reduce the workload on operators. Many universities and research institutions, such as the Massachusetts Institute of Technology and Carnegie Mellon University, began to develop autonomous UAVs with advanced sensors, control systems, and software.

With the increasing interest in UAVs, academic institutions began offering specialized programs in UAV engineering and related fields. For instance, Embry-Riddle

Aeronautical University offers a Bachelor of Science in Unmanned Aircraft Systems Science, which covers the fundamentals of UAV design, operation, and maintenance. Similarly, the University of North Dakota offers a Bachelor of Science in Aeronautics with a major in Unmanned Aircraft Systems Operations. These initiatives have created exciting opportunities for research and development in various types of UAVs.

In recent years, the academic community has also been focusing on developing UAVs for civilian applications, such as mapping, surveying, and monitoring natural resources. This has led to the emergence of interdisciplinary research areas, such as remote sensing and geospatial analysis, which combine UAV technology with other fields such as environmental science, geology, and geography. Overall, the academic history of UAVs has been characterized by a growing interest in autonomous UAVs and the development of specialized programs in UAV engineering. As UAV technology continues to advance, we can expect further contributions from the academic community in developing innovative applications and improving the safety and efficiency of UAVs.

1.2 Application of UAVs

Over the past 40 years, the military UAV revolution has produced technological advancements that have accelerated the development of the civilian UAV industry. This growth is fuelled by a large number of UAVs application in the civilian sector and potential users for each application [5], as shown in Figure 1.1. There are four major categories of civil UAV applications: commercial, civil, security, and scientific. The potential civilian UAV applications in each category are highlighted in Table 1.1.

With the rise in popularity, the number of incidents involving UAVs has increased drastically due to inexperienced operators [6]. The advancements in technology, sensors, communication, and control systems have significantly increased the safety and development of a wide range of UAVs, varying in characteristics, configuration, shape, and size.

Table 1.1: Examples of Civilian Applications for UAVs

Application	**Description**
Commercial	
Agriculture	Monitoring harvest areas
Marine fisheries	Sensing and monitoring
Commercial imaging	High-altitude imaging and surveillance
Cargo delivery	FedEx and United Parcel Service (UPS) delivery
Telecommunications	Wide-band internet and temporary networks
Multiple vehicle coordination	Coordinating planes and ships for collaboration
Aerial surveying	Photographic surveying
Civil	
Monitoring of critical infrastructure	Monitoring the pipelines, aqueducts, dams, power lines, ports, etc., from the sky
Traffic monitoring	Sea and land traffic monitoring
Real-time disaster observation	Assess the damage after disasters and detecting the methods to provide relief
Land imaging	Inspecting land areas for resources
Search and Rescue	Rescuers and prevent accidents
Forestry and fire monitoring	The UAVs can utilize to detect fires in forests
Emergency communications	Can be used as means of communication by setting temporary networks during emergencies
Security	
Surveillance and reconnaissance	Homeland Security
Border and Harbour patrols	Monitoring harbors and borders for violations
Law enforcement	Helpful, especially in open or rural areas
Scientific	
Meteorology weather and	Monitoring weather conditions
Geological surveys	Mapping geological data
Environmental monitoring	Monitoring environmental conditions
Hurricane reconnaissance	Damage prevention and aiding relief efforts

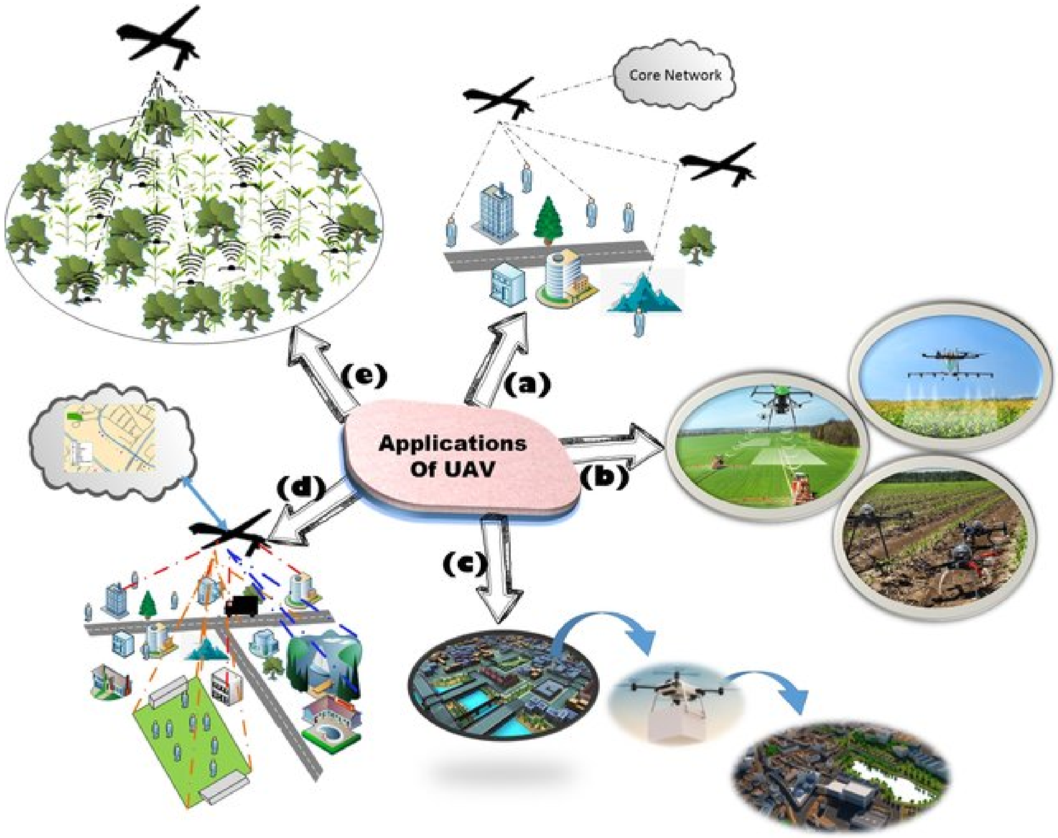


Figure 1.1: Civilian Applications of UAV (a)-Communication, (b)-Agriculture, (c)-Transportation (delivery), (d)-Surveying, (e)-Data Dissemination

1.3 Types of UAV

The UAVs have different capabilities, sizes, and models, from recreational mini-drones to large surveillance and control drones. UAVs are categorized according to their specifications. In general, UAVs are classified according to their wing types [7]. The main types of UAVs are single-wing, multi-rotor, fixed-wing, and fixed-wing hybrid Vertical Take-Off and Landing (VTOL) at different scales [8]. These types of UAVs are:

1.3.1 Single-Rotor UAVs

Single-rotor UAV types are durable and strong. The structure and design of these UAVs are similar to the helicopters. Figure 1.2 shows one main rotor on top for lift and a tail rotor for stability and direction control.

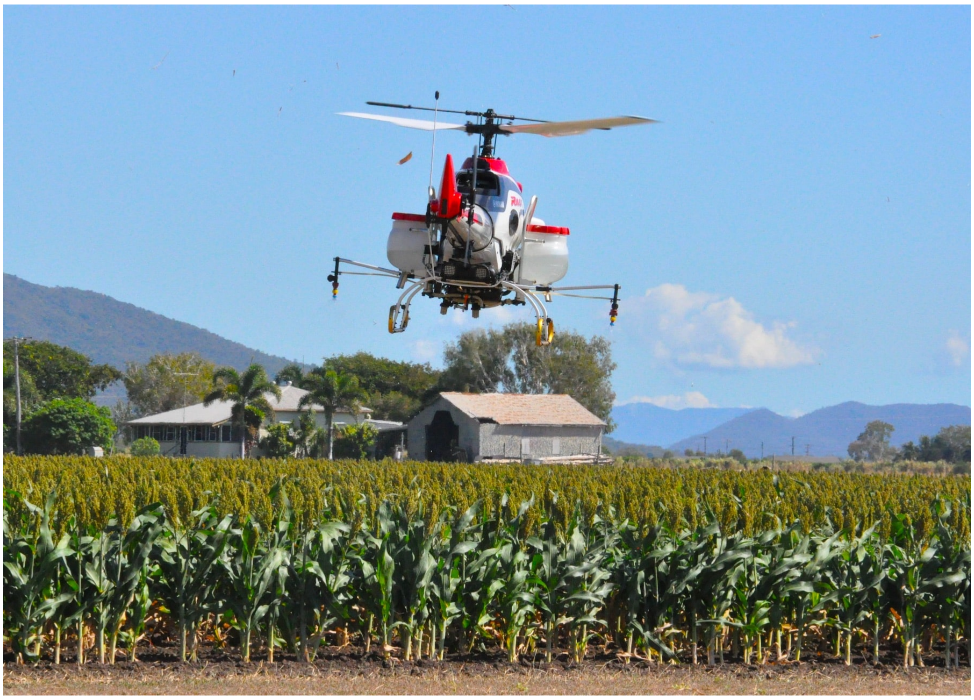


Figure 1.2: Single Rotor Drone for Agriculture Use

1.3.2 Multi-Rotor UAVs

Multi-rotor UAVs are the cheapest and easiest option for an “eye in the sky”. They also offer greater position control and framing; hence, they are perfect for aerial surveillance and photography. They are known as multi-rotors because they have more than one rotary wind, most commonly tri-copters (3 rotors), quad-copters (4 rotors), hexacopters (6 rotors), and octocopters (8 rotors). A quadcopter shown in Figure 1.3 is the most popular type of multi-rotor UAV.

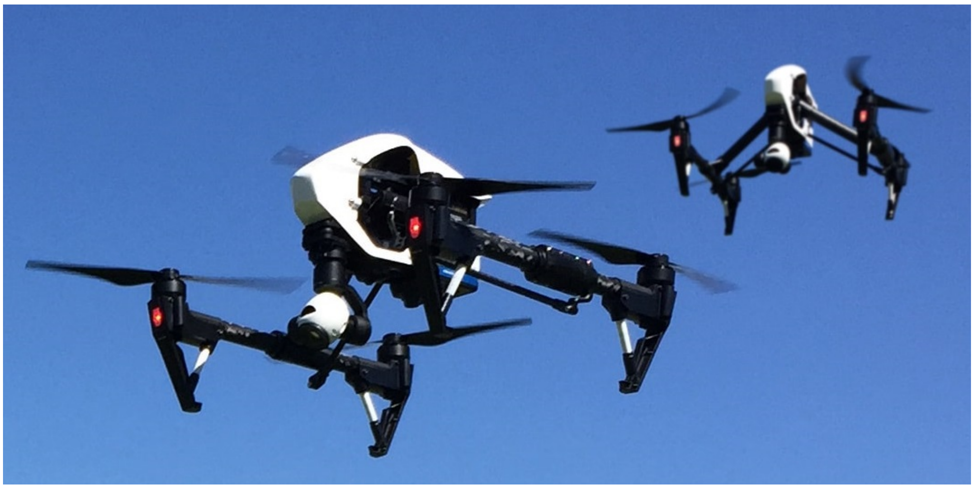


Figure 1.3: Two Multi-rotor Drones Flying at Height

1.3.3 Fixed-Wing UAVs

The fixed-wing UAV shown in Figure 1.4 has one rigid wing, similar to an airplane, which provides lift with propellers rather than vertical lift rotors. Therefore, they only need the energy to move forward and not to maintain altitude. Thus, they are energy-efficient.

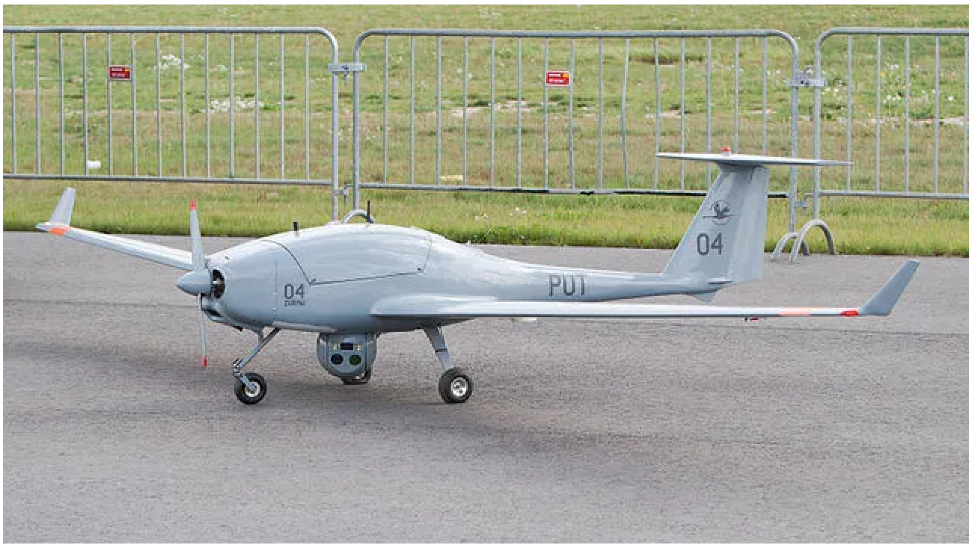


FIGURE 1.4: Fixed Wind UAV

1.3.4 Fixed-Wing Hybrid VTOL

Hybrid VTOL UAVs merge the benefits of rotor-based and fixed-wing designs. This UAV-type hovers, take-off, and lands vertically with rotors attached to fixed wings. Currently, only a few hybrid-design UAVs are in the market, but as technology advances in the coming years, they may become more popular. Figure 1.5 shows an example of a fixed-wing hybrid VTOL with an application of Amazon's Prime Air delivery drone.

A brief analysis based on the pros and cons of different types of UAVs is presented in Table 1.2.

In this work, our focus will be on fixed-wing UAVs that can fly for a long distance/duration at a higher speed than rotary-wing UAVs. However, sometimes

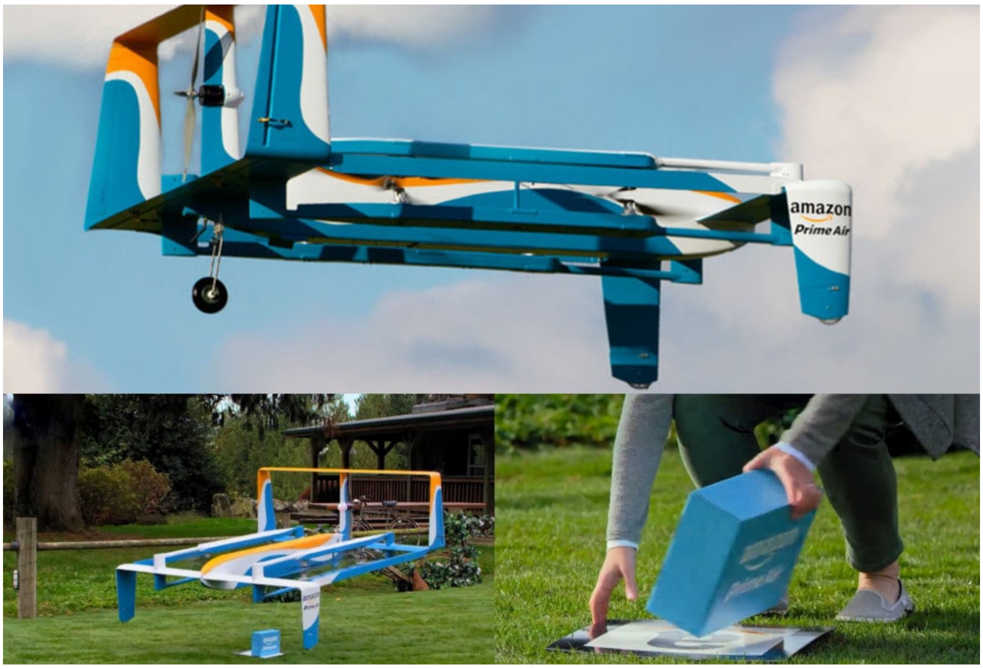


Figure 1.5: Unique Fixed-wing Hybrid VTOL UAV used by Amazon Prime Air

they required a runway for take-off and landing. This type of UAV has different phases during the flight operation, which are explained in the next section.

1.4 UAVs Flight Phases

In aeronautics, UAVs are one of the main research areas as they have advantages over manned air vehicles. They can perform a high-precision flight for a long period without being affected by factors such as visibility and pilot fatigue. The future plans for UAVs are to form a larger system through which the guidance and flight control should be autonomous, and the human will tackle only critical decisions of the mission. The flight envelopes of UAVs consist of different phases, namely, take-off (taxi, departure), cruise, and landing (descent, final approach, taxi) [9], as shown in Figure 1.6. Most UAVs have autonomous take-off and cruise but limited autonomous landing capabilities due to reliability issues and high risks. In the autonomous landing phase, the UAV must perform many sensitive and vital tasks in different environmental conditions. Therefore, the performance of UAVs must not fail in the presence of external disturbances (e.g., crosswinds, fog).

Table 1.2: Pros and Cons of Different Types of UAVs

Drone Type	**Pros**	**Cons**
Multi-Rotor	• Accessibility • Ease of use • VTOL and hover flight • Good camera control • Can operate in a confined area	• Short flight times • Small payload capacity
Fixed-Wing	• Long endurance • Large area coverage • Fast flight speed	• Launch and recovery needs much space • No VTOL/hover • Harder to fly, more training needed • Expensive
Single-Rotor	• VTOL and hover flight • Long endurance (with gas power) • Heavier payload capability	• More dangerous • Harder to fly, more training needed • Expensive
Fixed-Wing Hybrid	• VTOL • Long-endurance flight	• Not perfect at either hovering or forward flight • Still in development

Landing the UAV is considered one of the most difficult tasks as it requires tight control of the UAV's multiple states. Figure 1.7 shows the accident statistics of commercial aviation [10]. Almost half of all accidents occur during the final approach and landing stages. This amplifies in the case of UAVs because of the unavailability of the onboard pilot. As shown in Figure 1.8, the landing task

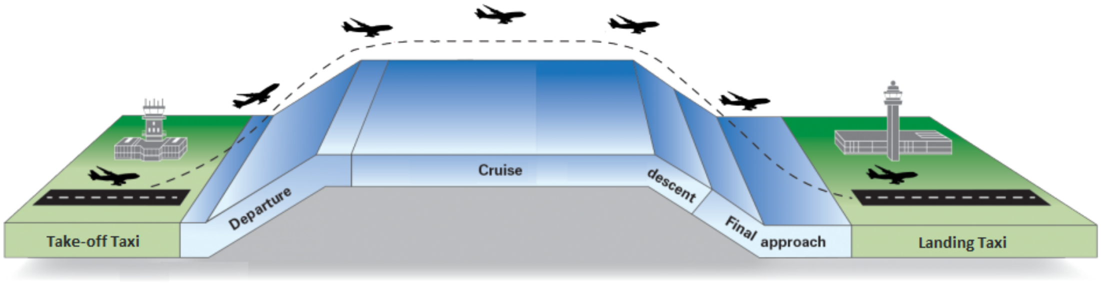


FIGURE 1.6: Phases of UAV

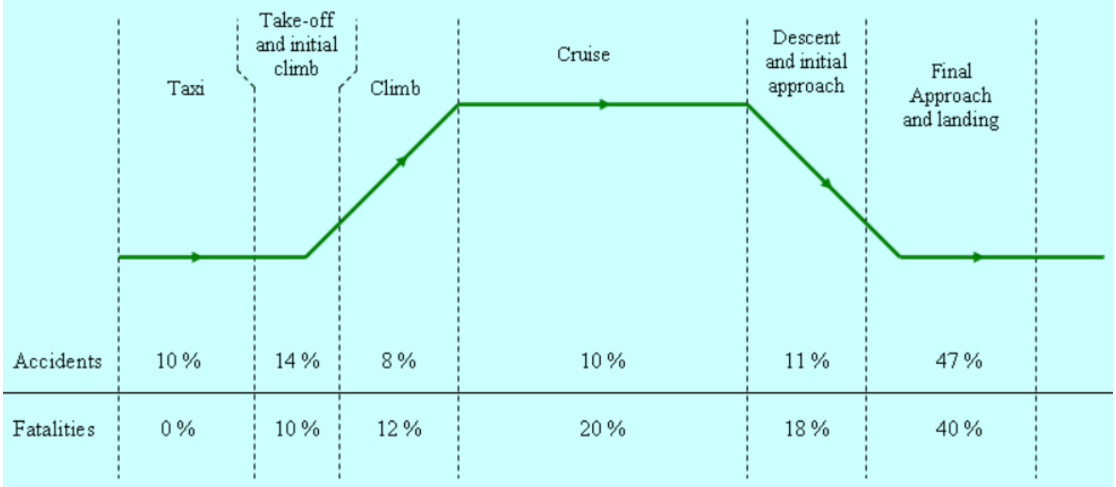


FIGURE 1.7: Statistics Observed on the 2004-2013 Period

has two phases: airborne (glide and flare maneuver) and ground taxi. In glide maneuver, the UAV must descend along the predefined straight-line path in the longitudinal plane with a fixed negative flight path angle between -4^o and -3^o, toward the runway [11]. Sometimes, the glide maneuverability is performed in two steps. Initially, a higher decent angle is taken, and a lower descent angle is attained in the latter step. When the UAV reaches around $25 - 30$ m altitude, the flare maneuverability is initiated. In this phase, the UAV is required to reduce the descent rate and follow a curved path. It is necessary to bring the flight path angle near zero for a smooth touchdown and the minimum impact on the landing gears. Figure 1.9 illustrates the typical landing maneuver with the glide and flare path indication. After the touchdown, the ground taxi phase begins in which the UAV needs to steer the centerline of the runway to stop [12].

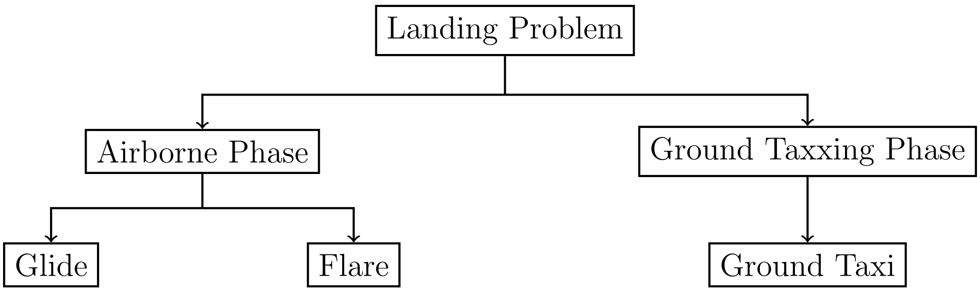


FIGURE 1.8: UAV Landing Problem

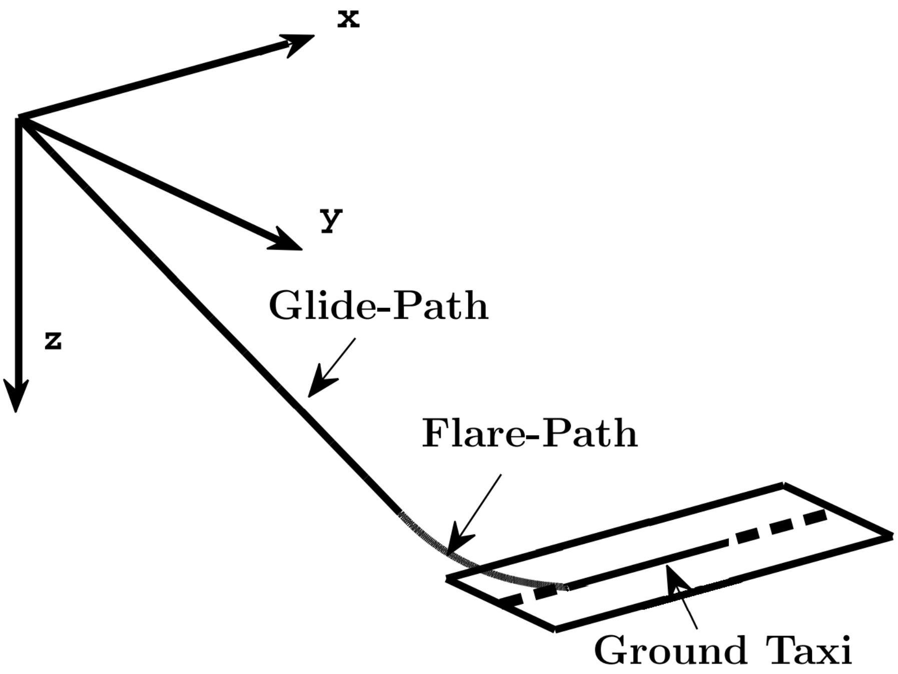


FIGURE 1.9: Glide, Flare and Ground Taxi Phases

In typical UAVs, a human pilot is needed to handle the landing. This is usually done to reduce the system cost and complexity and mitigate the risks. Operational experiences have shown that most UAV disasters are due to human errors. Furthermore, it takes several hours of flight and significant financial investment to train a human pilot, who can handle these phases efficiently and safely. There are also severe restrictions on the human pilot during flight operations; for example, he cannot land the UAV in thick fog and high crosswind. Thus, equipping the UAV with the ability to perform landing autonomously increases the system's complexity but can potentially render more versatile UAVs. It also can reduce the long-term costs and risks involved in the landing. Moreover, external disturbances, such as wind gusts, shear, and downbursts, can be tackled more efficiently.

1.5 Motivation

Working on UAV landing can be a highly motivating and challenging task, as it involves solving a number of complex problems related to control, navigation, and perception. Here are a few potential motivations for working on UAV landing:

- **Safety and Reliability:** Landing is one of the most critical phases of UAV operation, and ensuring that UAVs can land safely and reliably is of utmost importance. For individuals interested in improving the safety and reliability of UAVs, working on UAV landing can provide an opportunity to develop new technologies and algorithms that can improve the performance of these aircraft.
- **Autonomous Navigation:** Landing a UAV autonomously requires sophisticated algorithms and sensors that can accurately estimate the UAV's position and orientation relative to the landing site. Working on UAV landing can be highly motivating and rewarding for individuals interested in developing advanced navigation algorithms and perception systems.
- **Real-World Applications:** UAVs have a wide range of practical applications, from aerial surveying to package delivery. Landing is a critical component of many of these applications, and developing technologies that enable UAVs to land safely and reliably can significantly impact the feasibility and effectiveness of these applications.
- **Academic Research:** Landing is a complex and challenging problem that has been the subject of extensive research in the academic community. For individuals interested in pursuing a career in academic research, working on UAV landing can provide an opportunity to contribute to the development of new knowledge and technologies in this field.
- **Career Opportunities:** As the market for UAVs continues to grow, there is a high demand for individuals with expertise in UAV landing and related technologies. Working on UAV landing can provide an opportunity

to develop highly valued skills and knowledge, leading to exciting career opportunities in industry and academia.

1.6 Research Objectives

The main objective of this work is to reduce the risk induced by a human pilot. With the steadily increasing applications of UAVs, it is also necessary to have an autonomous flight control system that provides a proper solution in degraded environmental conditions. According to [13], Human error is responsible for roughly 60% of UAV accidents during operation, and surprisingly 50% of the incidents are during the take-off and landing procedure. One of the advantages of autonomous take-off and landing systems is the elimination of human error. Eliminating the operator from manually controlling the UAV during take-off and landing procedures and replacing them with an autonomous system will greatly increase safety during operations. Due to the lack of onboard human pilots, the UAVs depend on the control system for navigation and obstacle avoidance.

1.7 Research Scope

The goal of this work is to design the control system for a fixed-wing UAV that enable it to land autonomously. It is similar to the aircraft landing, where an Instrument Landing System (ILS) assists the pilot in landing safely. In the case of UAVs, the unavailability of onboard pilot increase the risk involved in the landing. The problem could be worsened by physical hindrances such as wind shear or gusts. An autonomous landing system aims to respond fast and accurately to these types of scenarios and reduce the risk of an error in judgment by a human pilot. It is done with the help of H_∞ based model predictive control technique. The control system is evaluated for its ability under moderate and severe downburst during the glide and flare maneuverability. After the touchdown, the effects of ground forces contributing to the dynamics of the UAV are also tackled, and a lateral taxi control is designed. These contributions allow a complete, consistently accurate, safer landing system for a UAV.

By the end of this work, the author should understand flight dynamics and the control algorithms applied to UAVs in practice.

1.8 Thesis Contributions

The major contributions of this thesis have been divided into two parts: the airborne phase and the ground taxi phase. The following individual contributions lead to these objectives.

1.8.1 Airborne Phase

In the airborne phase, the UAV glides toward the runway, and for smooth touch, it performs flare maneuverability when it reaches near to the ground. The wind shear is a major challenge during this phase. The H_∞ based MPC is designed to tackle these effects. This contribution has resulted in the following journal publication:

Z. Latif, A. Shahzad, A.I. Bhatti, J.F. Whidborne and R. Samar, “Autonomous Landing of a UAV Using H_∞ Based Model Predictive Control,” *Drones*, no. 6, vol. 12, pp. 416, 2022.

1.8.2 Ground Taxi Phase

After the touchdown on the runway, the ground forces and moments start contributing to the dynamics of the UAV. The UAV decelerates/accelerates during the ground taxi phase, which makes the control more complex due to the coupling between aerodynamics and tire-ground friction forces. Moreover, the lateral control during the deceleration phase becomes more challenging due to the high initial longitudinal velocity at touchdown and uncertainty in the tire-ground friction model.

These effects are significantly amplified in high wind or gust conditions. The controller could damage or lose the system if it does not perform well for all ground velocities. In this work, the non-linear taxi model has been established and linearized analytically, and a simplified and accurate LPV model is proposed. Then, H_∞ based LPV-MPC is proposed for the directional control. This contribution has led to the following research publications:

Z. Latif, A. Shahzad, R. Samar and A.I. Bhatti, "Lateral Parameter-Varying Modelling and Control of a UAV on-Ground," *4th IFAC Workshop on Linear Parameter Varying Systems (LPVS) Milan, Italy*, no. 8, vol. 54, pp. 130-135, 2021.

Z. Latif, A. Shahzad, A.I. Bhatti and R. Samar, "Lateral Control of an UAV on the Ground using H_∞ based Linear Parameter Varying Model Predictive Control," (submitted).

1.9 Thesis Outline

The rest of the thesis is structured as follows:

- The literature review related to the mathematical modeling of taxi phase and control techniques is given in Chapter 2. A comprehensive literature survey has been carried out for the control of the UAV during the airborne phase (glide and flare maneuverability) under wind shear effects, ground taxi modeling, control, and tuning methods of MPC. Based on the literature survey, gaps are identified, and a problem statement has been formulated.

- Chapter 3 is dedicated to explain the formulation of H_∞ based MPC technique. The design steps for observer plus state feedback control synthesis are presented. Linear Quadratic Regulator (LQR) based inverse optimal problem is formulated, and Linear Matrix Inequalities (LMIs) are solved to design the cost function matrices.

- Control design for the airborne phase is presented in Chapter 4. The mathematical models for landing trajectory, UAV, and wind shear are presented. The H_∞ based MPC presented in Chapter 3 is developed for the glide and flare phase. The designed algorithm is assessed in a simulation environment under moderate and severe wind shear effects, and results are compared with benchmark work.

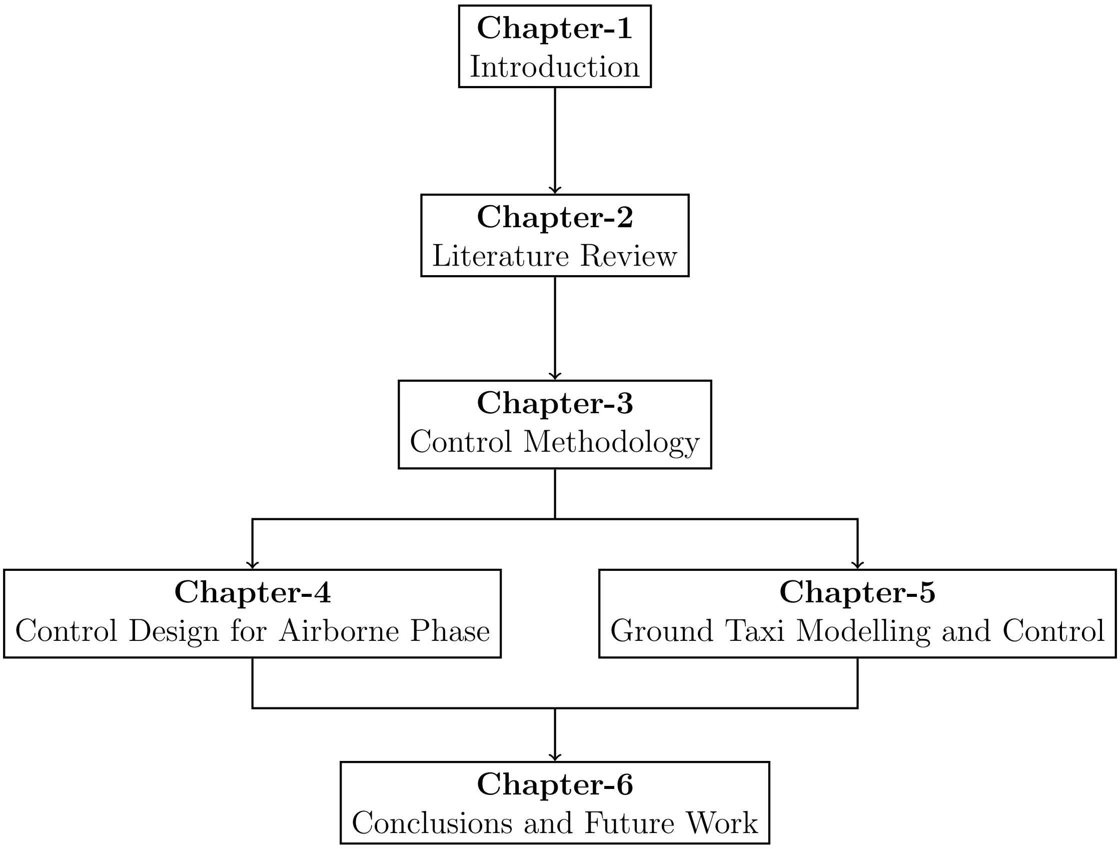


FIGURE 1.10: Thesis Layout

- The ground taxi modeling and control is discussed in Chapter 5. First, a complete non-linear taxi model has been derived, including the forces generated by aerodynamics and the interaction of the tires and runway. The model is analytically linearized and shows that it is explicitly dependent on the UAV's forward velocity, which continuously varies during both take-off and landing roll. Then the LPV model is derived for the taxi phase. Since a single controller designed at a particular velocity point does not satisfy the design requirements of all other velocity points, an H_∞ based LPV-MPC is proposed.

- Finally, the conclusions of the thesis and the future directions are presented in Chapter 6.

Figure 1.10 presents the overall thesis layout.

Chapter 2

Literature Review

This chapter discusses the research relevant to the landing problem of fixed-wing UAVs. The landing problem has two maneuverability: airborne and ground taxi phase and the problem is reviewed separately for both. Moreover, the MPC tuning has also been discussed with reference to the previously published work. The breakdown of the survey is shown in Figure 2.1. The research gaps are identified based on the survey, and the problem is formulated at the end of the chapter.

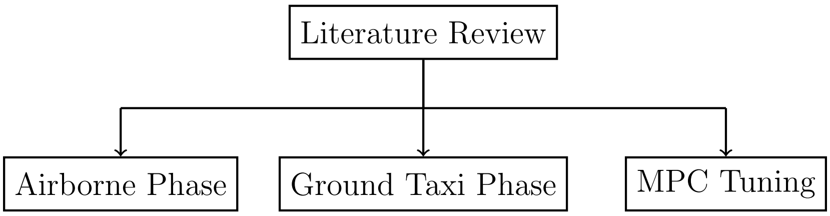


FIGURE 2.1: Literature Review Breakdown

2.1 Airborne Phase

In the airborne phase, the UAV initially glides with a constant negative slope to decrease the altitude and initiate the flare phase when it reaches the ground for a smooth touchdown. This maneuver is important because any disturbance may distract the UAV and damage or lose the system. Therefore, a robust flight control system is required that can keep the UAV on the desired trajectory.

Classical control techniques are preferred for autonomous landing because of their reliability and simplicity. However, these techniques do not satisfy the desired performance and robustness requirements. In recent years, with the developments in the hardware and computing capacity, the difficulties in implementing modern control techniques have been significantly reduced [14]. Therefore, the modern control techniques base flight system is now a mature research area. The researcher implemented many methodologies, from vision-based navigation to fuzzy logic control algorithms, to achieve a safe and smooth landing.

A flight algorithm for the autonomous landing of a UAV is proposed in [15]. Different control loops are used to address the specific task separately, e.g., angle of sideslip control or heading control. A multivariable H_∞ controller is designed for flare maneuverability. At the top level, a non-linear guidance control law provides the 3D path following capability. A control law is proposed in [16] for Autonomous Carrier Landing (ACL) system. The non-linear landing model is transformed into a poly-topic model, and the arresting and ground approach risks are proposed and integrated using Kalman Filter (KF). The risk-state MPC is presented based on the landing risk gradient. In [17], a framework is proposed for landing in an uncertain environment based on point cloud in a coarse to fine manner. It has four modules: preprocessing point cloud, selection of course landing site, evaluation of fine terrain, and optimal landing model. An ACL problem is solved in [18] with input constraints and external disturbance. A relative motion model between the ideal glide path and the UAV is established, and the wave disturbance to the carrier is considered an external disturbance. A backstepping control law is proposed having input constraints, and the Lyapunov function is used to prove the stability of the closed-loop system.

In [19], a model following technique based on LQR is given to track the glide and flare trajectory. In [20], deep-stall landing for the fixed-wing UAV is proposed. The UAV is guided along the predefined custom path and performs landing at low speed with good precision. The landing control of high-speed UAV under wind interference is presented in [21]. The Monte Carlo simulations are used to show the system's robustness under wind disturbance. In [22], an automatic landing technique is presented which uses an H_2/H_∞ technique along a dynamic inversion method. An optimal observer is considered to tackle sensor errors and other disturbances. The design aims to develop a landing control system that not only cancels the negative effects of sensor errors and disturbances but also gives a good result when the number of states is more than the number of sensors. The PID controllers (conventional and fuzzy variant) are compared with the dynamic inversion concept in [23]. The PID controllers are tuned for pitch, altitude, and velocity. The controllers are tested for errors in the gyro sensors and wind shears. Numerical simulations validate the results. In [24], a sliding mode control (SMC) technique is designed for autonomous landing. The landing maneuver is divided into glide path capture and flare. The two phases specify the flight path in a longitudinal landing plane, a straight line, and an exponential curve. The obtained controller is validated by simulating the landing maneuver using a non-linear model with a large offset in the initial position from the reference landing trajectory. The results are compared with the conventional PID controller.

A hierarchical control structure for the autonomous landing is developed in [25]. Active disturbance rejection for attitude control, proportional guidance law for height tracking, and PID is used for heading angle, flight path angle, and taxi control. The Hardware In the Loop (HIL) simulation and field experiments are conducted to demonstrate the performance of the entire test system and the proposed approach. An intelligent landing system based on an Artificial Neural Network (ANN) is given in [26]. The weights are searched using the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) algorithm. The ANN is used to control the pitching torque and thrust of the UAV to obtain a smooth landing. The same algorithm is used in the case of high wind conditions. All possible flight conditions should be considered in the training data for efficient training of ANN. A fuzzy logic-based autonomous landing technique of UAV is proposed in [27]. Three fuzzy logic modules control the UAV's global position (longitude-latitude) against the runway for altitude, speed, and position. An SMC with a Cerebellar Model Articulation Controller (CMAC) for the landing of a UAV is presented in [28]. The parameters of the SMC are adjusted by the use of a Genetic Algorithm (GA), Chaotic Particle Swarm Optimization (CPSO), and Particle Swarm Optimization (PSO). An intelligent landing scheme based on different CAMCs is presented in [29]. The lyapunov theory is used for stability analysis and to obtain adaptive learning rules.

2.2 Taxi Phase

Modeling and control of UAVs are currently one of the main research areas in industry and academia. The flight control of UAVs in the air has been studied extensively. However, the research in controlling a UAV on-ground to keep it on the runway centerline remains limited. The on-ground dynamics and control are more complex due to the coupling between aero and ground dynamics. Moreover, the lateral control during the deceleration phase becomes more challenging due to the high initial longitudinal velocity at touchdown and the uncertain effects of the tire-ground friction model. These effects are significantly amplified in high wind or gust conditions. If the controller does not perform well for all ground velocities, it could lead to damage or loss of the system. Therefore an efficient model and robust flight control system are required that can steer the vehicle on the runway during take-off and landing.

Many researchers have studied the UAV ground dynamics and autonomous take-off control. In [30], an LPV PID control technique is presented for the taxi control of UAVs. The grid LPV modeling approach tackles the varying mechanism of the model. The non-linear model is linearised at each gridding point, covering the entire parameter space. The scheduling depends on longitudinal velocity, and the whole parameter space is divided into three subsets. The ground forces are ignored in this work and considered external disturbances. A power-by-wire braking system during the taxi phase is proposed in [31]. Instead of hydraulic, the electrical interface is used, which enhances safety and maintainability. The varying effect of the model is compensated by using a non-linear adaptive backstepping controller. The Tracking Deviation Correction (TDC) problem is solved in [32]. The proposed algorithm is limited to high-speed taxing only.

The lateral control during on-ground deceleration is discussed in [33]. A weighted model predictive control law is proposed to control the parameter-varying system. Based on prior time sequences of the state vector, it provides plenty for the weight coefficient. The varying velocity of the model would increase the error in the reference model. The previous prediction step is more important than the following ones and needs a large plenty weight coefficient. To improve the on-ground dynamics of the aircraft, a parameter-varying anti-windup control design technique is developed in [34]. Typically the anti-windup applications appear on the actuator's saturation. However, here the author used to represent the non-linear effects of ground forces which strongly depend on the runway state. This operation is handled by a Linear Friction Transform (LFT) based estimator. The model is also assumed to perform well on a large operating range. In [35], a directional control law is developed for UAVs. The primary control inputs are nose-wheel deflection, main wheel differential braking, and rudder for low, medium, and high speed. A priority matrix is developed to decide the secondary control if the primary controller fails.

The ground steering dynamic model of the UAV is built in [36] by using aerodynamics and ground forces. The bifurcation theory is used to analyze some key parameters' effects on the UAV's steering stability. Moreover, the kinetic characteristics and loading features are analyzed to check the instability mechanism of the model. A non-linear taxi model is presented in [37] for the implementation of ATOL algorithms. A successive linearization is used to handle the non-linearities of airspeed in the lateral dynamics. It allows us to design and gain scheduled control laws. In [38], the ground run phase has been divided into different portions to make the control more efficient. A simple proportional gain controller is used to control the heading of the UAV. A non-linear mathematical model for ground taxing is proposed in [39]. Directional control was designed for high velocity by differential braking.

In [40], a non-linear model of the aircraft during the taxi phase is presented by using both dynamics and kinematics analysis of the system. Then the model is reduced to the 3 Degree of Freedom (DoF) by considering that there is no roll and pitch moment in the taxi phase. A proportional-integral (IP) controller is used to design the directional control and keep the UAV on the centerline. In [41], a non-linear model for the taxi phase has been derived by taking the damping and stiffness model of landing gears. The model has a nose wheel and ruder as low and high velocity control actuators, respectively. A mathematical model for a diamond-shaped UAV is presented in [42] along the lateral control during ground roll. Three different gain setpoints have been used to consider the effects of the varying velocity of UAVs. In [43], an integrated control law is proposed for the take-off roll in which the gain varies with the forward speed of the UAV. To track the trajectory and yaw rate of UAV with varying longitudinal velocity, an integrated controller based on sliding mode control is given in [44]. In [45], the yaw rate of the aircraft is controlled during low ground speed. The controller was designed using feedback linearization based on the non-linear directional model.

2.3 MPC tuning

The classical control techniques give closed-loop stability, performance, and a certain degree of robustness but generally do not handle the constraints. Some modifications are proposed, e.g., anti-windup schemes to handle the input constraints properly. However, these may affect the closed-loop performance, complicate the design (especially for the multivariable system), and are usually restricted to a specific class of constraints. A more systematic way to deal with the constraints is the MPC strategy. In every control cycle, the MPC utilizes the current state's information to predict the model's evolution over the prediction horizon. Accordingly, the MPC designs a control input sequence that provides the best performance and satisfies the constraints. However, it is difficult to characterize MPC's robustness, stability, and frequency-domain properties in contrast to linear feedback controllers. In literature, different techniques are presented to tune the MPC to behave like any favorite linear feedback controller when constraints are inactive.

The tuning of MPC has been considered in several previously published articles. In [46], the MPC tuning is based on $\mathcal{H}_\infty$ loop-shape synthesis. The author solved the inverse optimization problem to compute the state estimator parameters and MPC cost function. By assuming $R = I$, he computes the Q_{opt} by solving the Algebraic Riccati Equation (ARE) and LQR equation simultaneously. He does not consider the constraints on output and augment the disturbance and reference into the plant model, assuming that the disturbance must be measured or estimated. Based on this work, the author of [47] considered the input and output constraints. He also considered the unmeasured disturbance without incorporating the disturbance model into the plant model. All the plant augmentations are based on the $\mathcal{H}_\infty$ loop-shaping compensator. In [48] and [49], the controller matching and finding a cost function problem is considered. The output and state feedback controllers are considered in these works, respectively. To find the cost function matrices P, Q, and R, a LMI problem is solved.

In [50], the inverse LQR problem is solved to find the cost function matrices. It utilized the LMI method to determine the solution of the general inverse LQR problem when both weights R and Q are unknown. It also proposed the gradient base least-squares minimization method to find the approximate solution when the LMIs are infeasible. To meet the closed-loop performance, the tuning of cost function matrices of MPC are given in [51]. Two semidefinite programming problems are solved sequentially, one of which is solved in the frequency domain. In the unconstrained case, the tuning parameters guarantee a nominal robust closed-loop performance. In [52], an MPC tuning method based on transfer function formulation is given. In the first step, MPC's gain matches the gain of some favorite controller by equating the respective transfer function coefficients. In the second step, a convex optimization problem with LMI constraints is solved to find the tuning parameters.

The paper [53] outlines a systematic approach to selecting the Model Predictive Control (MPC) stage cost. The method involves matching the MPC feedback law with a proportional-integral (PI) controller that is efficiently tuned using high-performance Monte Carlo (MC) simulation. The MC simulation tuning of the PI controller minimizes two objectives, namely, the 2-norm tracking error and a bi-objective comprising the 2-norm tracking error and a penalty on the 2-norm input rate of movement. The paper [54] presents an innovative technique for determining the weights of the MPC based on personal driving data through an automatic tuning method. Naturalistic driving conditions, which encompass car-following scenarios and driver behaviors, offer the personal driving data. These data enable the creation of a reference model that mirrors the driver's driving style. The automatic tuning problem is defined as an optimization problem that involves minimizing the difference between the controller's response and the reference model by utilizing optimal weight factors. To solve this optimization problem, the Particle Swarm Optimization (PSO) algorithm is employed.

The paper [55] presents a tuning procedure for multi-input multi-output (MIMO) systems using MPC. The method involves two steps based on a hybrid approach: the goal attainment method and a variable neighborhood search. In the first step, the weights of the MPC objective function are determined by minimizing the square error between a predefined reference trajectory and the closed-loop response of the internal controller model. In the second step, the integer variables of the problem, including prediction and control horizons, are optimized by minimizing the square error between the closed-loop response and an optimal trajectory to obtain a low computational cost controller with good performance. The proposed method is tested on two benchmark processes using different MPC formulations, and the results demonstrate its effectiveness. In [56] the authors developed an analytical method to calculate the cost function matrices of MPC in cases where the prediction horizon is short. They applied this method to control an empirically-based wave energy converter (WEC) model and successfully tuned the MPC to adhere to existing linear control laws and meet input and state constraints, such as actuator force and actuator stroke.

2.4 Gap Analysis

Most UAVs accidents that occur during the landing maneuver are due to severe weather conditions. During a flight, the UAV must cope with many natural uncertainties and disturbances, such as wind gusts, wind shear, sensor noise, parametric uncertainties, etc. In the worst-case scenario, the UAV might diverge from the reference path, e.g., in [57], where the path error reaches about 39 ft (approx. 12 m) and can create problems for the UAV. So, a well-adjusted and robust control strategy is required to tackle such scenarios and improve the tracking performance of the system by reducing path error. After the touchdown, the ground forces start contributing to the system's dynamics. The longitudinal velocity of UAVs varies over a wide range during the ground roll. Such characteristics lead to parameter-varying problems. However, most of the aforementioned papers do not consider these issues. Even a few took the gain scheduling but did not provide any explicit parameter varying structure for vehicle model, controller, or scheduled gain.

In literature, many control techniques are proposed for autonomous landing. The classical control techniques provide closed-loop stability and a certain degree of performance and robustness but generally do not consider the constraints. With some modification, e.g., an anti-windup scheme for input saturation, the constraints can be handled, but this reduces the closed-loop performance. A more systematic way to handle the constraints is the MPC strategy. MPC provides the best performance along the constraints satisfaction. However, MPC's robustness, stability, and frequency domain properties are more difficult to characterize than linear feedback controllers. Properties of both techniques can be achieved by solving an inverse optimal control problem using a linear state feedback control law. The problem is to design the weighting matrices (performance index) of the MPC controller so that it behaves like a linear state feedback controller (favorite controller) when the constraints are inactive. Hence, the closed-loop properties of MPC control match the classical control for the perturbations due to which states and inputs remain in the admission range.

2.5 Problem Statement

As discussed in Section 1.8, the landing problem is divided into airborne and ground taxi phases. Thus, the fundamental objectives of this work have two folds:

- Design an H_∞ based MPC for the airborne phase in the presence of wind shear effects
- Develop an LPV model and H_∞ based LPV-MPC for lateral control during the taxi phase to address the challenge of varying speed

2.6 Summary

This chapter presents a detailed literature review of the landing control of fixed-wing UAVs. The research openings are identified in the airborne phase control and modeling and control design for the ground taxi phase. Moreover, the tuning of MPC is also explored in the chapter. Based on this review, the gap and the research problem have been formulated.

In the next chapter, the control methodology is presented. The algorithm and the design steps of H_∞ based MPC are discussed.

Chapter 3

Control Methodology

In this chapter, we present practical procedures for multivariable controller design, which are relatively straightforward to apply and which, in our opinion, have an important role to play in industrial control.

3.1 Introduction

The classical control techniques provide closed-loop performance, stability, and a certain degree of robustness but generally do not handle the constraints [58]. Some modifications, e.g., anti-windup schemes, can be introduced to handle the constraints on input saturation [59]. However, these make the design complicated (especially for multivariable systems), limited to a restricted class of constraints, and may yield reduced closed-loop performance [49]. A more systematic way to handle the constraints is the MPC strategy. At each time step, the MPC uses the current state's information and predicts the system's evolution over a desired future horizon. Accordingly, the MPC designs the best input control sequences resulting in constraint satisfaction and the best performance. However, as mentioned in [60], it is more difficult to characterize the MPC's frequency-domain properties, robustness, and stability compared to many other linear state feedback techniques. It reduces the application of the MPC technique [61].

In the literature, [48, 49, 52], different techniques are presented for the selection of the objective/cost function matrices (P, Q and R) of the linear MPC controller in such a way that it behaves like any favorite linear controller (e.g., H_∞ controller for our case) when the constraints are inactive. Hence, for any disturbance around the equilibrium point for which the inputs and states remain within the admissible range, the closed-loop properties of the MPC match the H_∞ synthesis. The advantage of H_∞ based MPC is that, contrary to the H_∞, the resulting technique can handle the constraint properly during the transients along with the frequency domain properties of H_∞ controller when constraints are inactive. To obtain the cost function matrices based on H_∞ controller, an LQR-based inverse optimal problem is formulated and solved in this work. Now, one can design the cost function matrices (P, Q and R) more sensibly.

Once the objective function has been designed, state information is required at each time step. Like many other applications, the lateral taxi control's full state information is unavailable. A state observer is required to initialize the prediction model. It is a well-known fact that the KF can be used to estimate the state in a full state feedback system e.g., in Linear Quadratic Gaussian (LQG), which is the basis of MPC due to quadratic objective function. However, the KF does not guarantee stability margin [62]. Generally, this fact has been overlooked in the MPC literature, and typically the design methodology emphasizes constraints enforcement and performance satisfaction [47]. In this work, we used the normalized coprime factor robust stabilization approach, which provides a controller that can be written as plant observer form and have an optimized stability margin [63].

The rest of the chapter is defined as follows. In Section 3.2, the H_∞ Loop Shaping Design Procedure (LSDP) is explained. The LQR-based inverse optimal problem is formulated and solved in Section 3.3. The MPC problem is given in Section 3.4, and design steps for H_∞ based MPC are presented in Section 3.5.

3.2 H_∞ Loop Shaping Design Procedure

The H_∞ LSDP is a powerful and sensible procedure to stabilize the plant robustly. It uses the classical open-loop shaping frequency response to have the desired loop

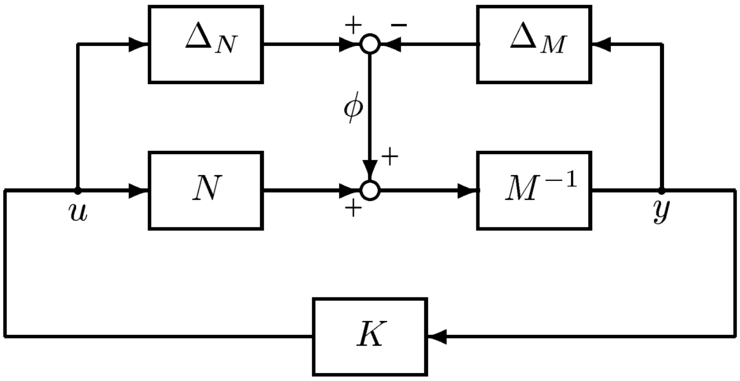


Figure 3.1: Robust Stability Block Diagram

shape. It has a two-step design procedure: first, the open loop plant is augmented to the desired shape of singular values with pre and post-compensators, and then the shaped plant is robustly stabilized with respect to coprime factor uncertainty with H_∞ optimization. The advantage of this procedure is that no problem-dependent weight selection or uncertainty modeling is required in the second step.

Consider a plant G which needs to stabilize with its normalized left coprime factorization as:

$$G = M^{-1}N \tag{3.1}$$

The perturb plant G_p can be written as

$$G_p = (M + \Delta_M)^{-1} (N + \Delta_N) \tag{3.2}$$

where Δ_N and Δ_M are unknown stable transfer functions, which show the uncertainty in nominal plant G. The objective of robust stabilization is not only to stabilize the nominal plant G but also a family of perturbed plants G_p around the nominal plant defined by:

$$G_p = \{(M + \Delta_M)^{-1} (N + \Delta_N) : \|[\Delta_N \quad \Delta_M]\|_\infty < \epsilon\} \tag{3.3}$$

where ϵ is the stability margin. The robust stabilization of coprime factors of a plant is to maximize the stability margin. The perturbed feedback system shown in Figure 3.1 is robust if and only if the nominal feedback system is stable and,

$$\gamma \triangleq \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_{\infty} \leq \frac{1}{\epsilon} \tag{3.4}$$

The minimum achievable value of γ, corresponding to the maximum stability margin ϵ is given by Glover and McFarlane (1989) as [63]:

$$\gamma_{min} = \epsilon_{max}^{-1} = \{1 - \|[N \quad M]\|_H^2\}^{-\frac{1}{2}} = (1 + \rho(XZ))^{\frac{1}{2}} \tag{3.5}$$

where $\| \ . \ \|_H$ is Hankel norms, ρ denotes the maximum eigen value of minimal state-space realization $(A, B, C\&D)$ of system G. Z is the unique positive definite solution of the algebraic Riccati equation:

$$\begin{aligned}(A - BS^{-1}D^TC)Z + Z(A - BS^{-1}D^TC)^T - ZC^TR^{-1}CZ \\ +BS^{-1}B^T = 0\end{aligned} \tag{3.6}$$

where,

$$R = I + DD^T, \quad S = D^TD$$

X is the unique positive definite solution of the algebraic Riccati equation:

$$\begin{aligned}(A - BS^{-1}D^TC)^TX + X(A - BS^{-1}D^TC) - XBS^{-1}B^TX \\ +C^TR^{-1}C = 0\end{aligned} \tag{3.7}$$

The controller guarantees

$$\left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_{\infty} < \gamma \tag{3.8}$$

for $\gamma > \gamma_{min}$

$$K \overset{S}{=} \left[\begin{array}{c|c} A + BF + \gamma^2(L^T)^{-1}ZC^T(C + DF) & \gamma^2(L^T)^{-1}ZC^T \\ \hline B^TX & -D^T \end{array}\right] \tag{3.9}$$

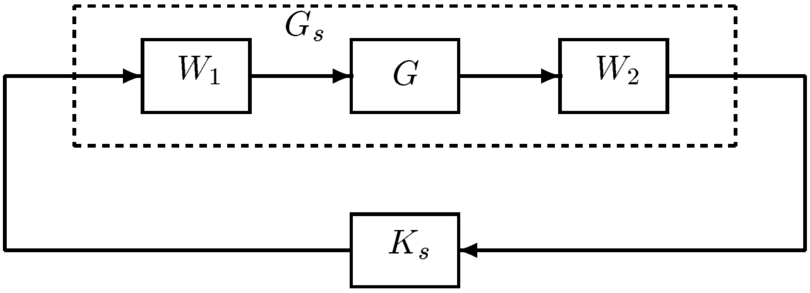


FIGURE 3.2: Shaped Plant and Controller

where

$$\begin{aligned} F =& \; -S^{-1}(D^T C + B^T X) \\ L =& \; (1-\gamma^2)I + XZ \end{aligned}$$

It can easily be implemented by MATLAB function `ncfsyn` and can find the robust controller for the given plant.

In robust stabilization, the designer is unable to specify the performance requirements. The author of [64] proposed per and post-compensating weights to shape the open loop singular values of the plant. Then, the shaped plant is robustly stabilized by using the above method. If pre and post weights are W_1 and W_2 respectively, then shaped plant G_s is given by

$$G_s = W_2 G W_1 \tag{3.10}$$

as shown in Figure 3.2. The controller K_s is synthesized by solving the robust stabilization problem for the shaped plant G_s. The feedback controller K:

$$K = W_1 K_s W_2 \tag{3.11}$$

is used to stabilize the plant G.

Weight selection is an important step in LSDP. Skills are required to select pre and post-weights, which shape the singular values as per requirement [65]. Typically, it is required to have a high gain at low frequencies for noise attenuation, a roll-off rate of approximately -20db per decade (slope of -1) at the crossover frequency, and a high roll-off at high frequencies. Hit and trail are involved in the selection of these weights. The author of [66] has shown that the single DoF H_∞ loop-shaping controller can be written as observer plus state feedback control law which is explained in the next subsection.

3.2.1 H_∞ Loop Shaping Design using Observer Structure

In general, the H_∞ controllers do not have an explicit structure. However, the normalized coprime factor robust stabilization method provides a controller that can be written in the form of a plant observer [67]. The resulting controller can be written as an exact plant observer and state feedback:

$$\begin{aligned} \dot{\hat{x}}_s &= A_s\hat{x}_s + H_s(C_s\hat{x}_s - y_s) + B_s u_s \\ u_s &= K_s\hat{x}_s, \end{aligned} \tag{3.12}$$

where A_s, B_s, and C_s are the state-space realizations of the shaped plant, $\hat{x}_s$ is the observer state vector, u_s is the input and y_s is the output vector of the shaped plant. The observer H_s and controller K_s for the shaped plant are:

$$\begin{aligned} H_s =& -Z_s C_s^T \\ K_s =& -B_s^T\left[I - \gamma^{-2}I - \gamma^{-2}X_s Z_s\right]^{-1} X_s \end{aligned} \tag{3.13}$$

where Z_s and X_s are the solutions of the following algebraic Riccati equations,

$$\begin{aligned} A_s Z_s + Z_s A_s^T - Z_s C_s^T C_s Z_s + B_s B_s^T = 0 \\ A_s^T X_s + X_s A_s - X_s B_s B_s^T X_s + C_s^T C_s = 0 \ . \end{aligned} \tag{3.14}$$

The maximum stability margin is calculated with $\epsilon_{\max} = \frac{1}{\gamma_{\min}}$, where $\gamma_{\min} = \sqrt{1 + \rho(X_s Z_s)}$, where ρ is the maximum eigenvalue of matrix $X_s Z_s$. If $\gamma_{\min} < 4$ the design is usually successful. The controller K_s is used to formulate and solve the inverse optimization problem to design the cost function matrices for MPC, which is presented in the next section.

3.3 Inverse Optimal Problem

In the optimal control theory, the LQR is used to find infinite horizon full-state feedback control law for both continuous and discrete time LTI systems [50]. The formulation for the shaped discrete-time Linear Time-Invariant (LTI) system is,

$$x_{k+1} = A_s x_k + B_s u_k \tag{3.15}$$

which minimizes the objective function

$$J = \sum_{k=0}^{\infty} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}. \tag{3.16}$$

The optimal feedback control is

$$u_k = -K_{\text{LQR}} x_k, \tag{3.17}$$

where,

$$K_{\text{LQR}} = (B_s^T P_s B_s + R_s)^{-1} B_s^T P_s A_s \tag{3.18}$$

here P_s is the unique positive semi-definite solution of the Discrete-time Algebraic Riccati Equation (DARE) given as:

$$A_s^T P_s A_s - P_s - (A_s^T P_s B_s)(B_s^T P_s B_s + R_s)^{-1} B_s^T P_s A_s + Q_s = 0 \tag{3.19}$$

where, Q_s is semi-positive definite and R_s is a positive-definite matrix. Now, if this problem is solved in the reverse order by assuming that we have an optimal feedback controller K_{LQR} and our objective is to find the cost function matrices for the shaped plant. The methods for solving this problem are explained in the coming subsection.

3.3.1 Solution of Inverse Optimal Problem

The basic aim is to find the cost function matrices (P_s, Q_s, and R_s). The inverse optimal problem is formulated with (3.18) and (3.19) by taking $K_{\text{LQR}} = K_s$. The resulting equations become as follows:

$$\begin{aligned} A_s^T P_s A_s - P_s - (A_s^T P_s B_s) K_s + Q_s = 0 \\ B_s^T P_s A_s - (B_s^T P_s B_s + R_s) K_s = 0 \end{aligned} \tag{3.20}$$

In (3.20), there are two equations and three unknowns (P_s, Q_s, and R_s). The author of [47] gave an analytical solution to the above problem by taking $R_s = I$. It may yield numerically ill-conditioned cost function matrices, which result in an ill-conditioned real-time optimization problem. An LMIs-based solution to this problem is presented in [50] by defining an additional criterion, the optimal solution must reduce the condition number of the cost function matrices. The LMIs-based optimization problem for the discrete-time system is defined as follows:

$$(\hat{Q}_s,\ \hat{R}_s,\ \hat{P}_s,\ \hat{\alpha}) = \arg \min_{Q_s, R_s, P_s, \alpha} \alpha^2,$$

such that

$$\begin{aligned} P_s \geq 0 \\ A_s^T P_s A_s - P_s - (A_s^T P_s B_s) K_s + Q_s = 0 \\ B_s^T P_s A_s - (B_s^T P_s B_s + R_s) K_s = 0 \\ I \leq \begin{bmatrix} Q_s & 0 \\ 0 & R_s \end{bmatrix} \leq \alpha I \end{aligned} \tag{3.21}$$

The LMI optimization problem (3.21) can be solved efficiently using the SeDuMi [68] package with the YALMIP modeling toolbox [69] in MATLAB. In contrast to the standard procedure for selecting cost function matrices for MPC, where usually diagonal terms are used based on performance, this method gives an off-diagonal term. These matrices are used to realize the MPC problem presented in the next section.

3.3.2 Limitation of Technique

When applying the cost function designed for an infinite time horizon in LQR-based inverse optimization to a finite time horizon scenario like Model Predictive

Control (MPC), several effects can be observed.

1. **Finite horizon objective:** The cost function in LQR is derived for an infinite time horizon, aiming to optimize long-term performance. In MPC, the objective is typically defined over a finite time horizon, focusing on optimizing short-term behavior. As a result, the cost function needs to be modified to reflect the desired objectives within the limited time frame.

2. **Dynamic optimization:** In MPC, the control inputs are recalculated at each time step, considering the current state of the system and a predictive model. This enables adaptability and responsiveness to changing conditions. Unlike LQR, which solves for a steady-state control law, MPC allows for dynamic optimization by incorporating time-varying constraints and objectives.

3. **Terminal conditions:** In LQR, the infinite time horizon implies a terminal condition of reaching a steady state. However, in MPC, the finite time horizon requires explicit specification of desired terminal conditions or setpoints. This ensures that the system converges to a desired state at the end of the time horizon.

4. **Computational complexity:** MPC involves solving an optimization problem at each time step, which can be computationally intensive. The finite time horizon increases the number of optimization iterations required compared to LQR, which only requires a one-time solution for the steady-state control law. Therefore, the computational complexity of the MPC algorithm may be higher.

5. **Performance trade-offs:** The selection of the time horizon in MPC involves a trade-off between short-term and long-term performance. A shorter time horizon allows for more responsive control actions but may sacrifice long-term stability, while a longer time horizon provides better stability but might be less responsive to immediate changes. Selecting an appropriate time horizon is crucial to achieve the desired control objectives.

In summary, when applying LQR-based inverse optimization to a finite time horizon scenario like MPC, modifications to the cost function, considerations for dynamic optimization, terminal conditions, computational complexity, and performance trade-offs are necessary to ensure effective control and achieve desired system behavior.Our study addressed the performance requirements by solving the LQR-based inverse optimization problem. The results indicate that the Model Predictive Control (MPC) approach closely adheres to the H_∞ control law.

3.4 MPC Realization

The MPC is a feedback control methodology in which an optimization problem is solved online to minimize the objective function subject to input, output, and/or state constraints. At each time step, a control input sequence is computed over a finite-time horizon that minimizes the objective function subject to constraints. The first element of the sequence is applied to the system on that sample time, and the process is repeated on every sampling step in a recursive, receding horizon manner.

The MPC control methodology is based on the following steps.

1. A prediction model,

$$x(k+1) = Ax(k) + Bu(k) \tag{3.22}$$

$$y(k) = Cx(k) \tag{3.23}$$

   where, $x(k)$ is the state vector of plant, $u(k)$ is the control input vector of the system and $y(k)$ is the output vector of the system at time kT, where T is the sample time.

2. An objective function,

$$J(u) = x^T(N)Px(N) + \sum_{k=1}^{N-1} x^T(k)Qx(k) + u^T(k)Ru(k) \tag{3.24}$$

The matrices $P \geq 0,\ Q \geq 0$, and $R > 0$ are designed to meet the desired performance requirements, and N is the prediction horizon.

3. The Inequality constraints,

$$v_{\min} \leq v(k) \leq v_{\max}, \tag{3.25}$$

here, $v(k)$ is constraint inputs, outputs and/or states and $v_{\min}$ and $v_{\max}$ are maximum and minimum limits on constraints.

4. An optimization algorithm to minimize the objective function.

The designed steps for the overall H_∞ based MPC are presented in the next section.

3.5 H_∞ based MPC Framework

The H_∞ based MPC methodology has the following design steps.

1. Design the input and output weights (W_1 and W_2) for the model to meet the desired closed-loop specifications. It is the basic step of the H_∞ loop shape design procedure, as explained in Section 3.2.

2. Synthesize the H_∞ compensator that has an observer-based state feedback structure. As a result, we have controller matrix K_s and observer matrix H_s for the shaped plant. The observer matrix H_s is used to estimate the state vector to initialize the prediction model at each time step in MPC.

3. By using the controller matrix K_s, from Step 2, formulate and solve an inverse optimization problem as presented in Section 3.3 to design cost function matrices P_s, Q_s and R_s.

4. Solve the MPC problem using the cost function matrices designed in Step 3.

This methodology inherits the transient response, reference tracking, disturbance rejection, and stability margin of H_∞ controller in the region where the constraints are inactive. As the observer is designed by H_∞ synthesis, it does not require additional tuning. In this formulation, the first three steps can be done offline. Once the cost function matrices are designed, the MPC algorithm uses them to solve an online optimization problem.

3.6 Summary

In this chapter, the control methodology used for landing the UAV is given in detail. The H_∞ LSDP is discussed along with the observer plus state feedback structure. The LQR-based inverse optimal problem is formulated, and the algorithm for the solution is described. Finally, the design steps for H_∞ based MPC are presented.

In the next chapter, the control is designed for the airborne landing phase by using the above algorithm, and the results are compared with the benchmark work.

Chapter 4

Control Design for Airborne Phase

Possibly the most critical phase of a UAV flight is landing. Autonomous landing systems can be used to reduce the risk due to pilot error. Environmental disturbances like wind shear can jeopardize safe landing; therefore, a well-adjusted and robust control system is required to maintain the performance requirements during landing. This work proposes a H_∞ based MPC approach for autonomous UAV landings.

The chapter is organized as follows. In Section 4.1, the mathematical models for landing trajectory, UAV, and wind shear are presented. In Section 4.2, the H_∞ controller is designed for the glide and flare phase. The MPC is realized for the landing control of UAV in Section 3.4. The results are discussed in Section 4.4 and summarized in Section 4.5.

4.1 Mathematical Models of the System

In addition to a model of the UAV dynamics, a mathematical model of the landing trajectory, as shown in Figure 1.9, is also necessary. It is the reference path that the UAV will follow during landing. The performance of the UAV is affected by wind disturbances, and a wind model is needed to calculate this effect. This section presents the mathematical model of the landing trajectory, the non-linear and linear UAV, and the wind-shear model.

4.1.1 Landing Trajectory Model

There are two maneuvers in the airborne phase: glide and flare. During the glide phase, the UAV descent with the glide path angle of -3^o to -4^o. When the UAV reaches an altitude of about 30 m, the flare maneuverability is executed. The descent rate of the UAV is decreased for a comfortable and smooth touchdown.

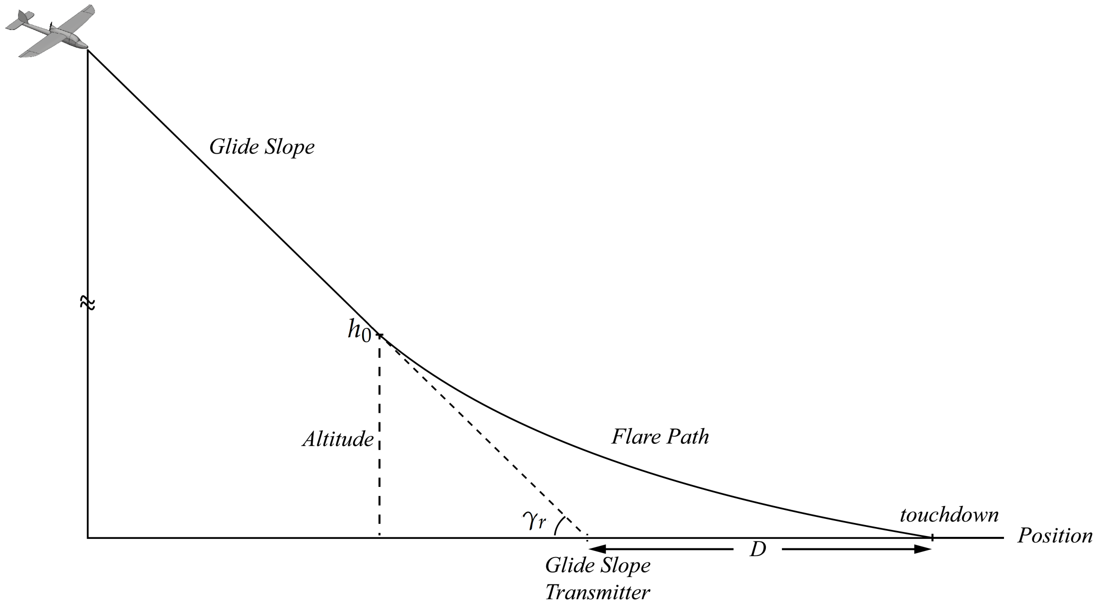


FIGURE 4.1: Reference Glide and Flare Path

For the glide phase ($h \geq h_0$, h_0 is the altitude at which the glide phase ends and flare begins), the reference altitude h_r and actual altitude h is:

$$\begin{aligned} \dot{h}_r &= V_0 \sin\gamma_r \approx V_0\gamma_r \\ \dot{h} &= V_0 \sin\gamma \approx V_0\gamma \end{aligned} \tag{4.1}$$

where $\gamma = \theta - \alpha$ is the actual glide path angle of the UAV during the first phase of landing and $\gamma_r = \theta_r - \alpha$ is the reference glide path angle, θ is the pitch angle,

α is the angle of attack, and V_0 is the nominal flight speed of the UAV. Generally $\gamma_r \approx 3^0$ hence the small angle approximations are justified in (4.1). The dynamics of the flare phase are presented in detail in [22]. The reference altitude h_r is usually an exponential decay function given as $h_r = h_0 \exp(-t/\tau)$, with τ as the time constant that defines the exponential curve of flare. The derivative of the above equation is:

$$\begin{aligned} \dot{h}_r &= -\frac{1}{\tau} h_0 \exp(-t/\tau) \\ &= -\frac{1}{\tau} h_r. \end{aligned} \tag{4.2}$$

By using equations (4.1) and (4.2), the reference pitch angle θ_r for both phases can be determined:

$$\theta_r \approx \alpha + \frac{\dot{h}_r}{V_0} \approx \begin{cases} \alpha + \gamma_r, & h \geq h0, \\ \alpha - \dfrac{1}{V_0 \tau} h_r, & h < h_0. \end{cases} \tag{4.3}$$

4.1.2 UAV Model

The UAV is assumed to be a rigid body in still air with decoupled lateral and longitudinal dynamics. The detailed mathematical model is presented in [70]. Only the longitudinal dynamics are considered in this work because the landing is mostly related to the longitudinal part. The longitudinal dynamics are given as follows:

$$\begin{aligned} \dot{U} &= -QW - g_0 \text{sin}\theta + \frac{F_x}{m} \\ \dot{W} &= QU + g_0 \text{cos}\theta + \frac{F_z}{m} \\ \dot{\theta} &= Q \\ \dot{Q} &= \frac{M_y}{J_y} \end{aligned} \tag{4.4}$$

where U is the longitudinal body-axes velocity, W is the vertical body-axes velocity, θ is the pitch, Q the is pitch rate, g_0 is the gravitational acceleration, m is mass, and J_y is the moment of inertia in the pitch of the UAV. F_x and F_z are the

Table 4.1: Physical Parameters of the Test Vehicle

Description	**Parameter**	**Value**
Mass	m	350 kg
Moment of inertia about y-axis	J_y	300 kg.m^2
Wing area	S	6.5 m^2
Mean aerodynamic chord	$\bar{c}$	6.6 m

aerodynamic forces along the $x-$ and $z-$axis, respectively and M_y is the pitching moment about the $y-$axis. These forces and moments are defined as:

$$\begin{aligned} F_x &= \frac{1}{2}\rho V_0^2 S C_x + T \\ F_z &= \frac{1}{2}\rho V_0^2 S C_z \\ M_y &= \frac{1}{2}\rho V_0^2 S \bar{c} C_m \end{aligned} \tag{4.5}$$

where $V_0 = \sqrt{U^2 + W^2}$, ρ is air mass density, T is thrust, S is wing area and $\bar{c}$ is mean aerodynamic chord. In (4.5), the aerodynamic coefficients can be defined as:

$$\begin{aligned} C_x &= C_{x_0} + C_{x_\alpha}\alpha + C_{x_{\delta_e}}\delta_e \\ C_z &= C_{z_0} + C_{z_\alpha}\alpha + C_{z_{\delta_e}}\delta_e \\ C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\delta_e}}\delta_e + \frac{C_{m_q} Q \bar{c}}{V_0} \end{aligned} \tag{4.6}$$

In this work, a medium-sized UAV is used as a test vehicle. The test vehicle's physical parameters and aerodynamic coefficients are given in Tables 5.2 and 4.2, respectively.

The non-linear model given in (4.4) is linearized by numerically perturbing the states and inputs about the operating point. The algorithm introduces a small perturbation to the states and inputs, one at a time, measures the system's response, and computes the state-spaces matrices. The longitudinal model has four states. However, the states can be expanded to include x-horizontal distance and

Table 4.2: Aerodynamic Coefficients of the Test Vehicle.

Parameter	**Value**	**Parameter**	**Value**
C_{x_0}	-0.031	C_x^{α}	-0.088
$C_{x_{\delta_e}}$	-0.01	$C_z^{\delta_e}$	-0.124
C_{z_0}	-0.129	C_z^{α}	-3.368
C_{m_0}	0.003	C_m^{α}	-0.4
C_{m_q}	-2.0	$C_{m_{\delta_e}}$	0.25

h-altitude of the UAV as:

$$\begin{aligned} \dot{x} =& \; U\cos\theta + W\sin\theta, \\ \dot{h} =& \; U\sin\theta - W\cos\theta. \end{aligned} \tag{4.7}$$

The non-linear model is trimmed for the conditions of $U = 50$ m/s, $W = 3$ m/s, $\theta = 0.5$ deg, $Q = 0$ deg/s, $h = 300$ m and initial position $x = 0$ m. The inputs to the system are elevator deflection δ_e and thrust δ_{th}. The state-space model is written as follows:

$$\begin{aligned} \dot{x}_v &= A_v x_v + B_v u_v \\ y_v &= C_v x_v \end{aligned} \tag{4.8}$$

where $x_v = [u \quad w \quad \theta \quad q \quad h \quad x]$ is the perturbed state vector, $u_v = [\delta_e \quad \delta_{th}]$ is the perturbed input vector and $y_v = [u \quad q \quad \theta \quad h]$ is the perturbed output vector. A_v, B_v, and C_v are the state space matrices of the vehicle model and are given below.

$$A_v = \begin{bmatrix} -0.0399 & -0.0541 & -0.1710 & -0.0524 & 0 & 0 \\ -0.2783 & -1.9805 & -0.0045 & 0.8727 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1.1360 & -18.3577 & 0 & -1.9144 & 0 & 0 \\ 0.0087 & -1 & 0.8737 & 0 & 0 & 0 \\ 1 & 0.0087 & 0.0295 & 0 & 0 & 0 \end{bmatrix}$$

$$B_v = \begin{bmatrix} -0.0050 & -0.0598 & 0 & 9.9890 & 0 & 0 \\ 0.0143 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

4.1.3 Wind Shear Model

Wind shear is the rapid variation in wind direction and/or speed over a short time or distance. If the diameter of wind shear is more than 4 km, it is called macro-burst, otherwise micro-burst. A micro-burst may last for a few seconds, but its effects, such as variation and extreme speed, can be very dangerous for the UAVs during airborne phase [71].

Various models of micro-burst have been developed. One is the vortex-ring model, proposed by Woodfield and Wood [72]. The vortex ring induces the velocity wind, and the micro-burst is represented by the two rings symmetrically to satisfy the boundary conditions. One is called the primary vortex ring above the ground, and the other is of the same strength below the ground and is called the imaginary vortex ring. Figure 4.2 illustrates the vortex-ring model.

Parameters r_C, R, and Γ represent the finite core's radius and the vortex-ring and vortex-ring model circulation, respectively. X, Y, and H are the center coordinates of the primary ring. Multiple pairs of ring can enhance the accuracy of the wind shear. A simplified vortex-ring down-burst model with two rings is presented in [57] and is used in this work. We can take $Y = 0$ because only longitudinal dynamics are considered in our study. If h and x are vertical and horizontal points

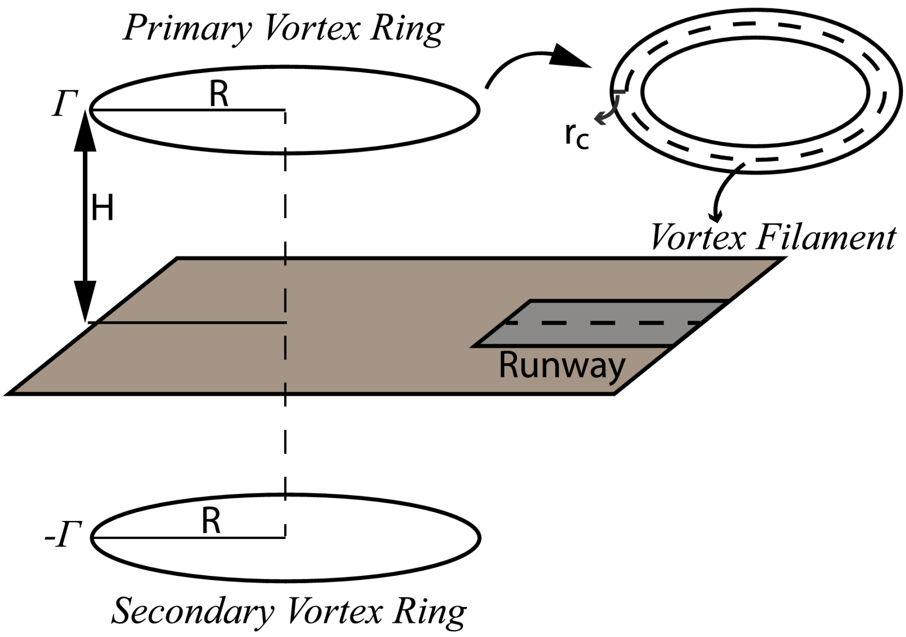


Figure 4.2: Vortex-Ring Model

of interest, respectively, then the induced velocities are computed as follows:

$$\begin{aligned}
x_1 &= x - X - R & x_2 &= x - X + R \\
h_p &= h - H & h_m &= h + H \\
r_{1p} &= x_1^2 + h_p^2 & r_{2p} &= x_2^2 + h_p^2 \\
r_{1m} &= x_1^2 + h_m^2 & r_{2m} &= x_2^2 + h_m^2 \\
r_0 &= \min\{r_{1p}, r_{2p}\} & \zeta &= 1 - \exp^{-(r_0/r_c^2)} \\
r_{xp} &= \sqrt{(x - X)^2 + h_p^2 + R^2} \\
r_{xm} &= \sqrt{(x - X)^2 + h_m^2 + R^2} \\
r_{hp} &= \left[(x - X)^2 + h_p^2 + R^2\right]^{3/4} \\
r_{hm} &= \left[(x - X)^2 + h_m^2 + R^2\right]^{3/4}
\end{aligned} \tag{4.9}$$

If $r_0 < \epsilon$, where ϵ is a small number that represents a point close to the ring filament, then,

$$W_x = 0, \qquad W_h = 0$$

Table 4.3: Downburst Parameters for Two Rings

Parameter	**Moderate**	**Severe**	**Unit**
Γ_1	18580	37160	m^2/s
R_1	1676	1524	m
H_1	610	610	m
R_{c1}	152	152	m
Γ_2	11148	26013	m^2/s
R_2	1220	1067	m
H_2	762	610	m
R_{c2}	152	91	m

Otherwise,

$$\begin{aligned} W_x =& \frac{1.182\Gamma\zeta}{2\pi}\left[\frac{R}{r_{xp}}\left(\frac{h_p}{r_{2p}}-\frac{h_p}{r_{1p}}\right)-\frac{R}{r_{xm}}\left(\frac{h_m}{r_{2m}}-\frac{h_m}{r_{1m}}\right)\right] \\ W_h =& \frac{1.576\Gamma\zeta}{2\pi}\left[\frac{R}{r_{hp}}\left(\frac{x_1}{r_{1p}^{3/4}}-\frac{x_2}{r_{2p}^{3/4}}\right)-\frac{R}{r_{hm}}\left(\frac{x_1}{r_{1m}^{3/4}}-\frac{x_2}{r_{2m}^{3/4}}\right)\right] \end{aligned} \tag{4.10}$$

where W_h and W_x are vertical and horizontal induced velocities, respectively. Two sets of parameters are shown in Table 4.3, which are used to calculate W_h and W_x for simulations. The first set of parameters represents a moderate down-burst and the second for severe down-burst. It is assumed that the UAV encounters the down-burst at the altitude of 300 m. Moderate and severe down-bursts are shown in Figure 4.6. Simulation models of UAVs need to incorporate the above mathematical representation of wind-shear effects. After embedding the wind-shear model in the equation (4.4) and (4.7), the UAVs model is as follow [57]:

$$\begin{aligned} \dot{U} &= -QW - g_0\sin\theta + \frac{F_x}{m} + \dot{W}_x \\ \dot{W} &= QU + g_0\cos\theta + \frac{F_z}{m} + \dot{W}_h \\ \dot{x} &= U\cos\theta + W\sin\theta + W_x \\ \dot{h} &= U\sin\theta - W\cos\theta + W_h \end{aligned} \tag{4.11}$$

where, $\dot{W}_x$ and $\dot{W}_h$ are the derivative of longitudinal and vertical wind components respectively and are considered as input signals.

4.2 Design of H_∞ Control

The linear model $G_\text{v} : C \mapsto C^{2\times 4}$ presented in 4.8 is used to describe the design methodology. To meet the desired specification following pre and post-compensators are selected,

$$\begin{aligned} W_1 &= \operatorname{diag}\left(\frac{3(s+1)}{s}, \quad \frac{(s+1)}{s}\right) \\ W_2 &= \operatorname{diag}\left(1, \quad 1.5, \quad \frac{(s+0.01)}{s}, \quad \frac{1.2(s+0.01)}{s}\right) \end{aligned} \tag{4.12}$$

The resulting shaped model/plant is obtained as:

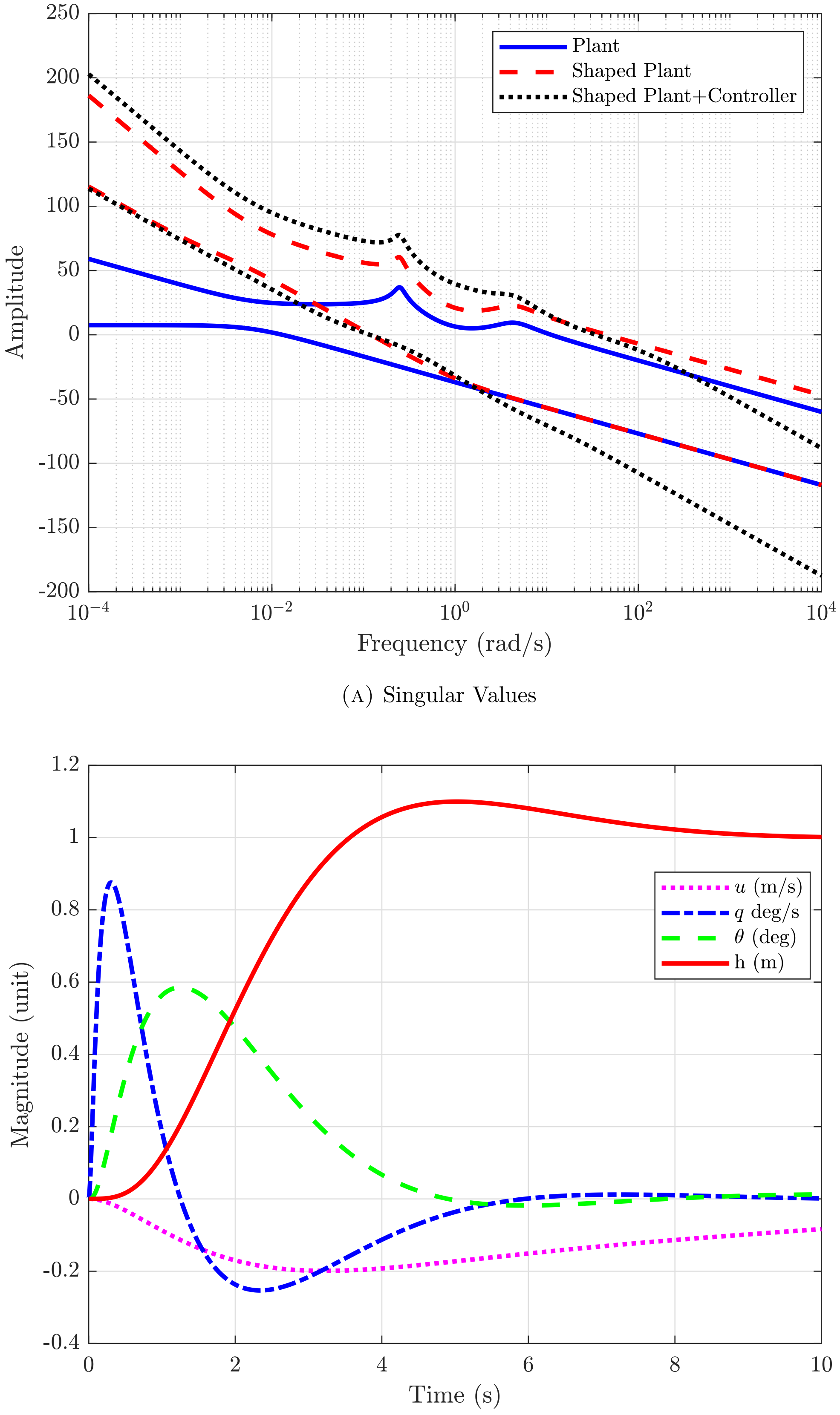
$$G_\text{sp} = W_1\, G_\text{v}\, W_2 \tag{4.13}$$

After including the weights, the state of the shaped plant has been increased. The state space representation of the shaped plant is:

$$\begin{aligned} \dot{x}_s &= A_s x_s + B_s u_s \\ y_s &= C_s x_s \end{aligned} \tag{4.14}$$

where x_s, u_s, and y_s are the shaped plant's state, input, and output vectors. The state space matrices A_s, B_s, and C_s for the shaped plant are given below.

Following the H_∞ loop shaping control procedure, as explained in Section 3.2.1, the controller K_s and observer H_s are synthesized. The normalized coprime factor robust optimization gave $\gamma = 2.81 < 4$ that meets the desired robustness criteria. The singular values of the plant, shaped plant, and shaped plant along with the controller are shown in the Figure 4.3a and the step response of altitude in the Figure 4.3b.



(A) Singular Values

(B) Step Response of Height

FIGURE 4.3: H_∞ Design Results

The state space matrices A_s, B_s and C_s the controller K_s, the observer H_s and MPC cost function matrices P_s, Q_s and R_s for the shaped plant are as follows:

$$A_s = \begin{bmatrix} 0.9992 & -0.0008 & -0.0034 & -0.0011 & 0 & -0.0010 & 0.0014 \\ -0.0052 & 0.9586 & -0.0000 & 0.0168 & 0 & 0.0026 & 0 \\ 0.0002 & -0.0036 & 1.0000 & 0.0196 & 0 & 0.0099 & 0 \\ 0.0224 & -0.3508 & -0.0000 & 0.9594 & 0 & 0.9800 & 0 \\ 0.0002 & -0.0196 & 0.0175 & 0 & 1 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_s = \begin{bmatrix} -0.0041 & 0.0105 & 0.0397 & 3.9596 & 0.0002 & 0.0800 & 0 \\ 0.0058 & 0 & 0 & 0.0001 & 0 & 0 & 0.0800 \end{bmatrix}^T$$

$$C_s = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \end{bmatrix}$$

$$K_s = \begin{bmatrix} 0.0244 & -0.56 & 1.0539 & 0.2459 & 1.1537 & 0.2479 & 0.0001 \\ 19.1416 & -12.125 & 20.0018 & 0.0215 & 32.2626 & 0.0026 & 0.2311 \end{bmatrix}$$

$$H_s = \begin{bmatrix} -2.54\times10^{-9} & -2.37\times10^{-8} & -3.68\times10^{-8} & -1.21\times10^{-6} & -2.54\times10^{-10} & -2.50\times10^{-8} & -6.22\times10^{-8} \\ 4.24\times10^{-6} & -3.79\times10^{-5} & -5.84\times10^{-5} & -1.91\times10^{-3} & -3.65\times10^{-7} & -3.95\times10^{-5} & -1.57\times10^{-8} \\ -2.21\times10^{-7} & 1.99\times10^{-6} & 3.06\times10^{-6} & 1.00\times10^{-4} & 1.91\times10^{-8} & 2.07\times10^{-6} & 1.07\times10^{-8} \\ 1.79\times10^{-9} & -1.60\times10^{-8} & -2.46\times10^{-8} & -8.07\times10^{-7} & -1.54\times10^{-10} & -1.67\times10^{-8} & 5.31\times10^{-11} \end{bmatrix}^T$$

The designed controller gain K_s is used to formulate and solve the LQR based inverse optimization problem presented in Section 3.3 to design the cost function matrices (P_s, Q_s and R_s). The resultant cost function matrices for the shaped plant are given below.

$$P_s = 10^8 \times \begin{bmatrix} 2.5343 & -0.2893 & 0.0596 & 0.0056 & 1.0865 & 0.0113 & -0.1385 \\ -0.2893 & 2.3353 & -1.5603 & -0.0701 & -1.2734 & 0.0421 & -0.0069 \\ 0.0596 & -1.5603 & 3.0612 & 0.1503 & 3.0576 & -0.1907 & 0.0417 \\ 0.0056 & -0.0701 & 0.1503 & 0.0405 & 0.1650 & -0.0587 & -0.0005 \\ 1.0865 & -1.2734 & 3.0576 & 0.1650 & 5.7402 & 0.0968 & -0.0025 \\ 0.0113 & 0.0421 & -0.1907 & -0.0587 & 0.0968 & 2.9988 & -0.0008 \\ -0.1385 & -0.0069 & 0.0417 & -0.0005 & -0.0025 & -0.0008 & 0.0105 \end{bmatrix}$$

$$Q_s = 10^8 \times \begin{bmatrix} 0.0687 & -0.0253 & 0.0667 & 0.0098 & 0.1241 & 0.0018 & -0.0029 \\ -0.0253 & 0.3092 & -0.3552 & -0.0868 & -0.3736 & -0.0166 & -0.0006 \\ 0.0667 & -0.3552 & 0.7049 & 0.1409 & 0.8329 & -0.0052 & 0.0004 \\ 0.0098 & -0.0868 & 0.1409 & 0.0390 & 0.1578 & -0.0005 & -0.0009 \\ 0.1241 & -0.3736 & 0.8329 & 0.1578 & 1.0808 & 0.0022 & -0.0001 \\ 0.0018 & -0.0166 & -0.0052 & -0.0005 & 0.0022 & 0.1176 & -0.0000 \\ -0.0029 & -0.0006 & 0.0004 & -0.0009 & -0.0001 & -0.0000 & 0.0004 \end{bmatrix}$$

$$R_s = 10^4 \times \begin{bmatrix} 0.0932 & -0.0002 \\ -0.0002 & 1.6236 \end{bmatrix}$$

4.3 MPC Design

The MPC problem presented in Section 3.4 is realized for the shaped model given in (4.14) having input constraints $[-25^\circ \;\; 25^\circ]$, and $[0 \;\; 100]\%$ for elevator deflection and percentage thrust, respectively. The dense approach presented in [73] is used for given LTI systems. The cost function and prediction model with prediction horizon N can be written as follows:(The subscript "s" is ignored in the rest of the work since we are designing MPC for a shaped system.)

$$J(z, x_0) = x_N^T P x_N + \sum_{k=1}^{N-1} x_k^T Q x_k + z_k^T R z_k$$

$$\begin{aligned} J(z, x_0) =& x_0^T Q x_0 + \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}^T \underbrace{\begin{bmatrix} Q & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & Q & 0 \\ 0 & 0 & \cdots & 0 & P \end{bmatrix}}_{\bar{Q}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} \\ &+ \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}^T}_{z} \underbrace{\begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \end{aligned}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{T}} x_0 + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\bar{S}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\begin{aligned} J(z, x_0) &= x_0^T Q x_0 + \left(\bar{T}x_0 + \bar{S}z\right)^T \bar{Q} \left(\bar{T}x_0 + \bar{S}z\right) + z^T \bar{R} z \\ &= \frac{1}{2} z^T \underbrace{2(\bar{R} + \bar{S}^T \bar{Q} \bar{S})}_{H} z + x_0^T \underbrace{2\bar{T}^T \bar{Q} \bar{S}}_{F} z \\ &= \frac{1}{2} z^T H z + x_0^T F z \end{aligned} \tag{4.15}$$

where z is the input vector. As we are designing the control system for the shaped plant, it is necessary to convert the input constraint limits from u_v to u_s. The constraints matrices for the shaped plant are as follows:

$$\underbrace{\begin{bmatrix} C_{w1}B_{w1} & 0 & \cdots & 0 \\ C_{w1}A_{w1}B_{w1} & C_{w1}B_{w1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{w1}A_{w1}^{N-1}B_{w1} & C_{w1}A_{w1}^{N-2}B_{w1} & \cdots & B_{w1} \end{bmatrix}}_{G} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ x_{N-1} \end{bmatrix} \leq \begin{bmatrix} u_{\max} \\ u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix} - \underbrace{\begin{bmatrix} C_{w1}A_{w1} \\ C_{w1}A_{w1}^2 \\ \vdots \\ C_{w1}A_{w1}^N \end{bmatrix}}_{S} x_{w1}$$

The Interior Point Method (IPM) is used to minimize the cost function (4.15) along with constraints at each time step. The detailed results are presented and discussed in the next section.

4.4 Results and Discussion

The proposed control law is implemented in a non-linear simulation environment to validate the designed algorithm. The simulation was conducted for two different scenarios. In the first case, the designed control law is tested for reference tracking in an ideal environment (without external disturbance) and the second case illustrates the disturbance rejection property of the design. The wind shear effects given in Section 4.1.3 are used as external disturbances, and results are compared to the study presented in [57].

4.4.1 Case-1

In the first case, the linear model of the UAV given in (4.8) is used to ensure that it follows the reference trajectory under ideal conditions. The altitude error is minimized by elevator deflection and thrust demand. Figure 4.5a shows that the UAV track the reference trajectory closly with minimum path error. The result also indicates that the angle of attack (α), pitch (θ), and flight path angle (γ) have been smoother, as shown in Figure 4.5b. Figure 4.5c and 4.5d show the elevator and thrust demand respectively.

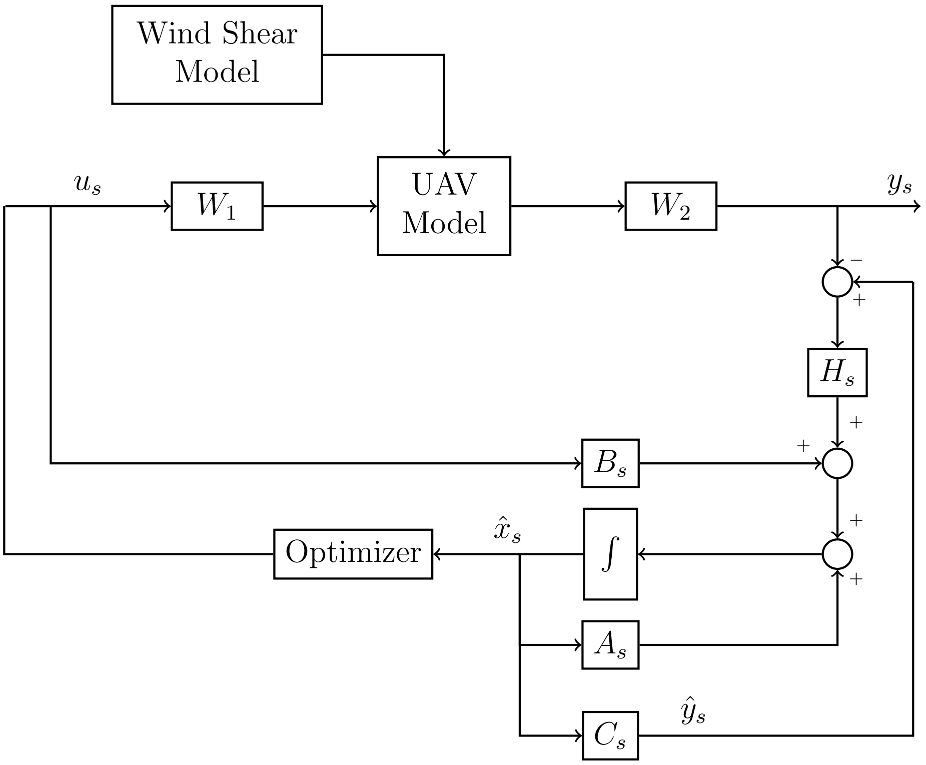
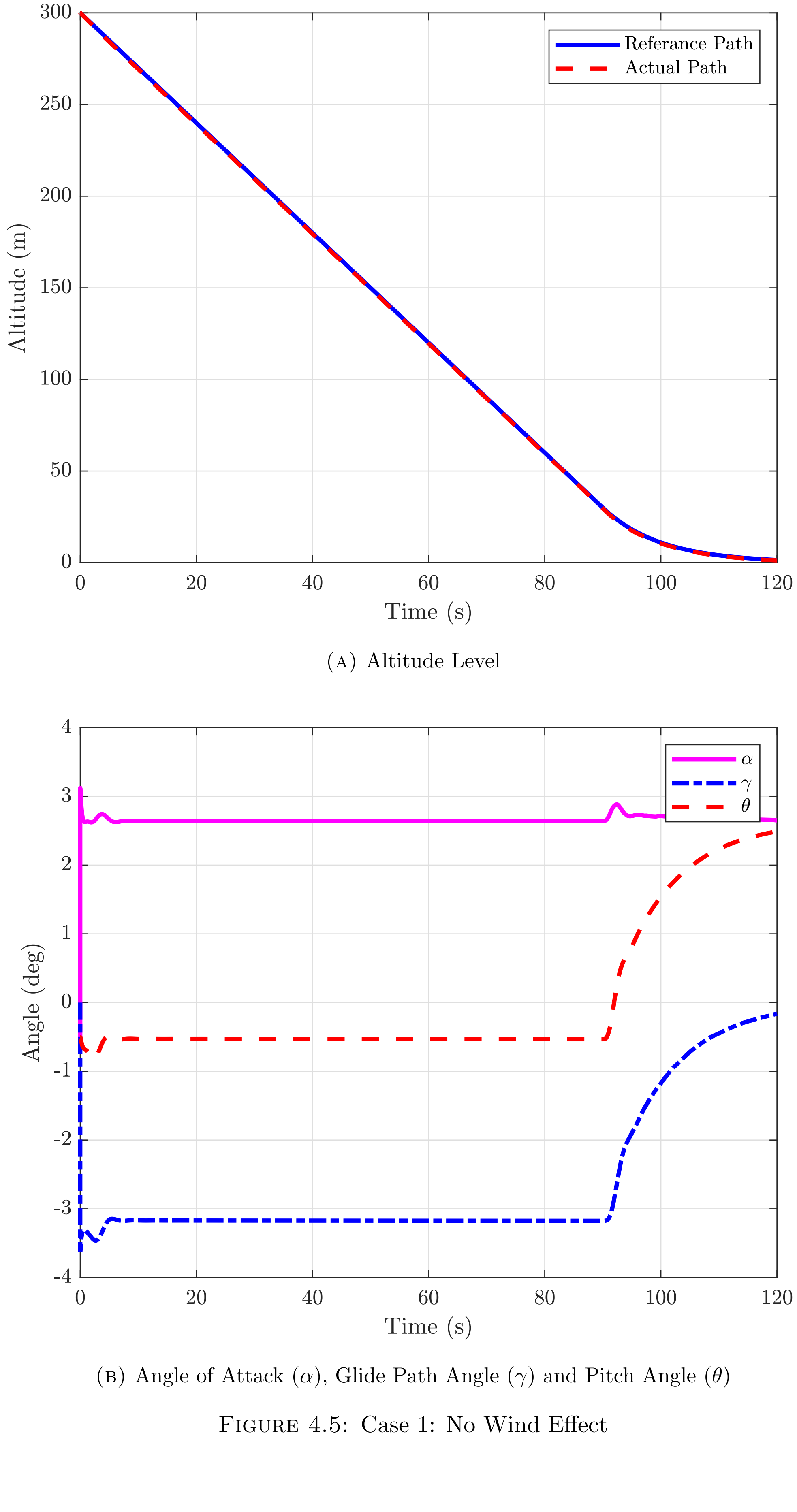


FIGURE 4.4: An Observer Based Implementation of H_∞ based MPC with Wind Shear

4.4.2 Case-2

The design algorithm is assessed under moderate and severe wind shear in the second case. The UAV encounters the wind at the start of the glide slope, and its effects last until the flare ends. It is assumed that the initial altitude of the UAV is 300 m, and the flare phase begins at an altitude of 30 m. In (4.11), winds-shear components were introduced into the dynamic equations of the UAV. The equation shows that altitude depends on U, W, θ, and W_h. The wind shear component W_h is a disturbance input that cannot be controlled; however, the controller can compensate for the wind shear effects using U, W, and θ. The controller dynamically adjusts θ and hence γ to keep the UAV on the desired path. Contrary to the ideal case, it is not required to maintain a constant angle of attack α in the presence of the wind shear.

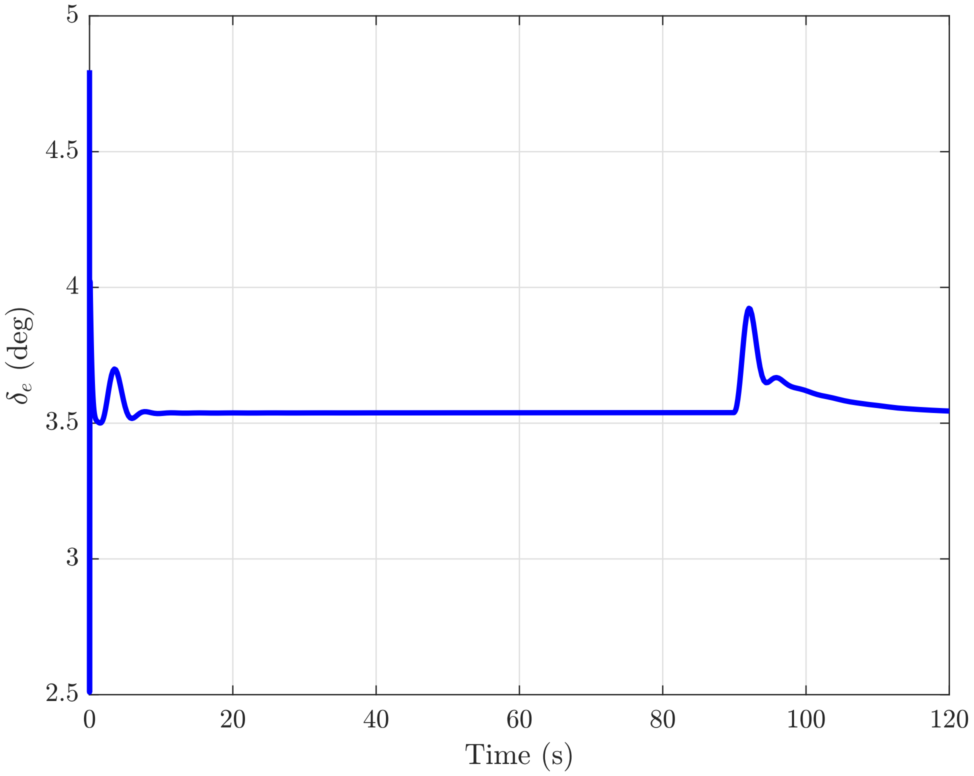
Figure 4.7a shows that the headwind increase during the first 20 seconds, and the UAV has increased the longitudinal velocity component to compensate for its effects. Figure 4.7b shows that the angle of attack also increases in response to the



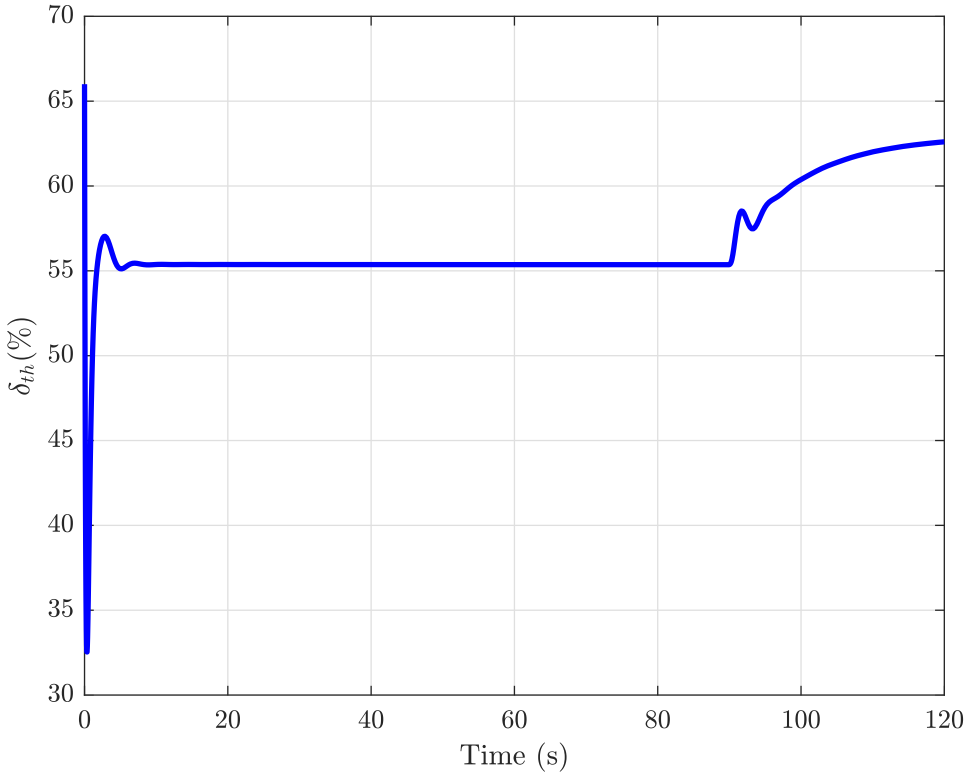
(A) Altitude Level

(B) Angle of Attack (α), Glide Path Angle (γ) and Pitch Angle (θ)

FIGURE 4.5: Case 1: No Wind Effect

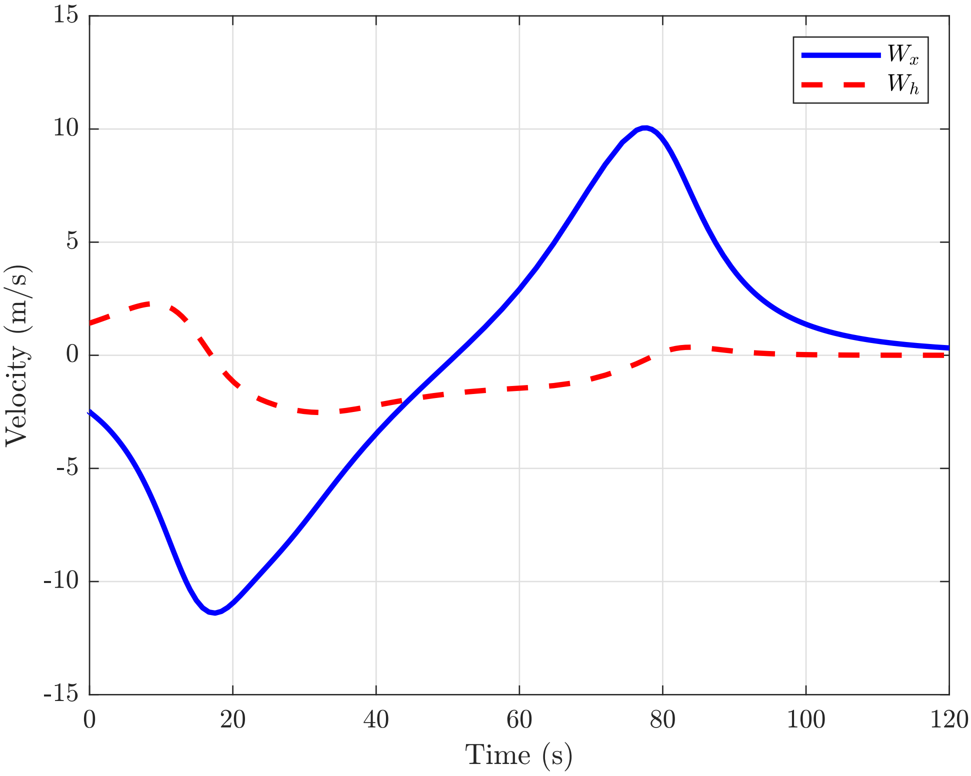


(c) Elevator Deflection

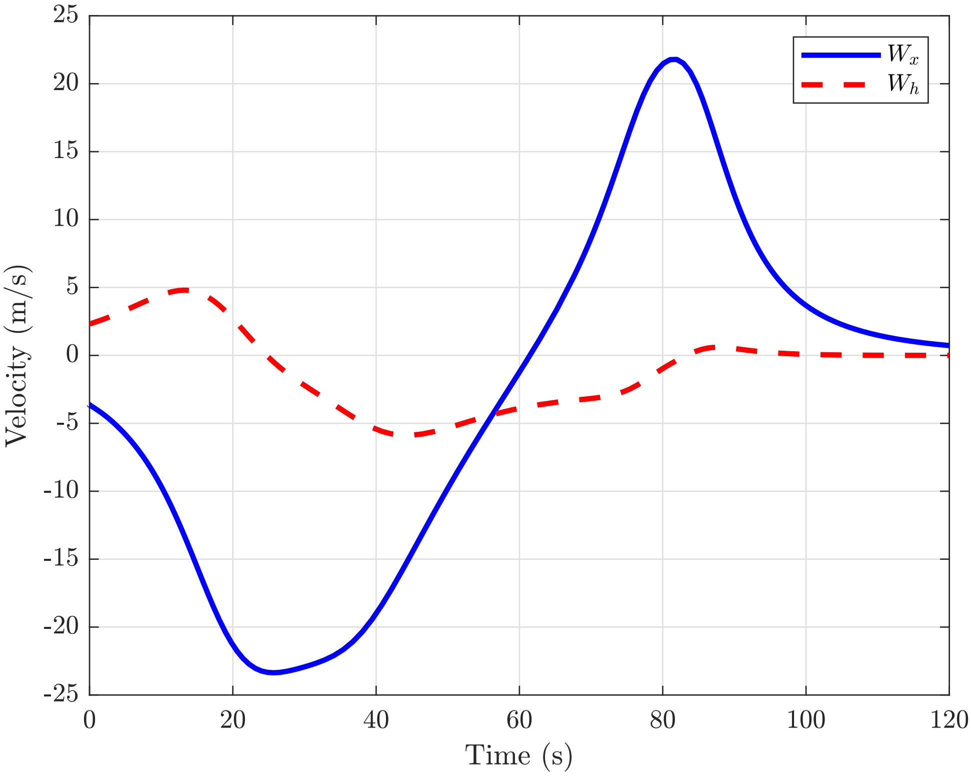


(d) Thrust Demand

Figure 4.5: (Continued)

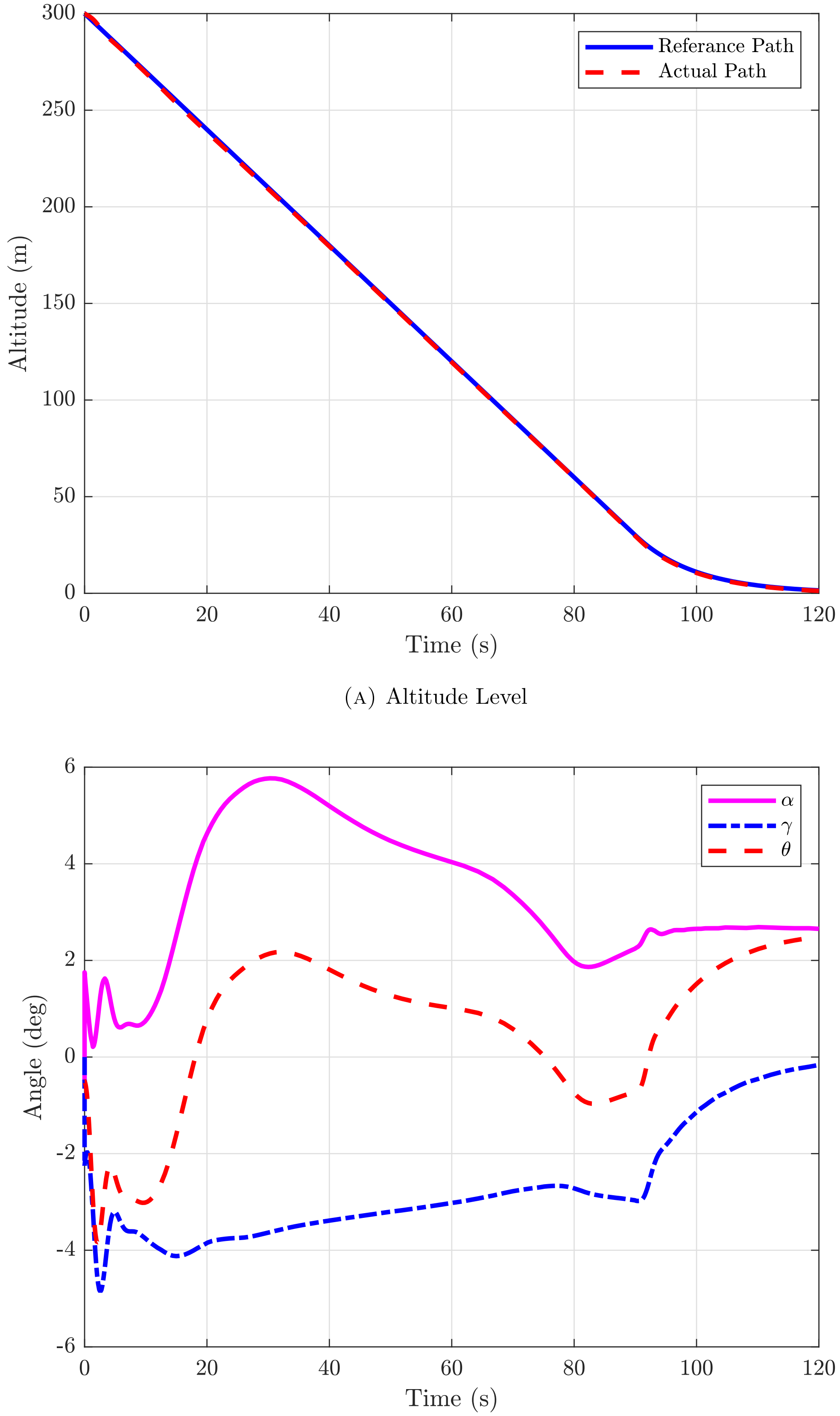


(A) Moderate Downburst



(B) Severe Downburst

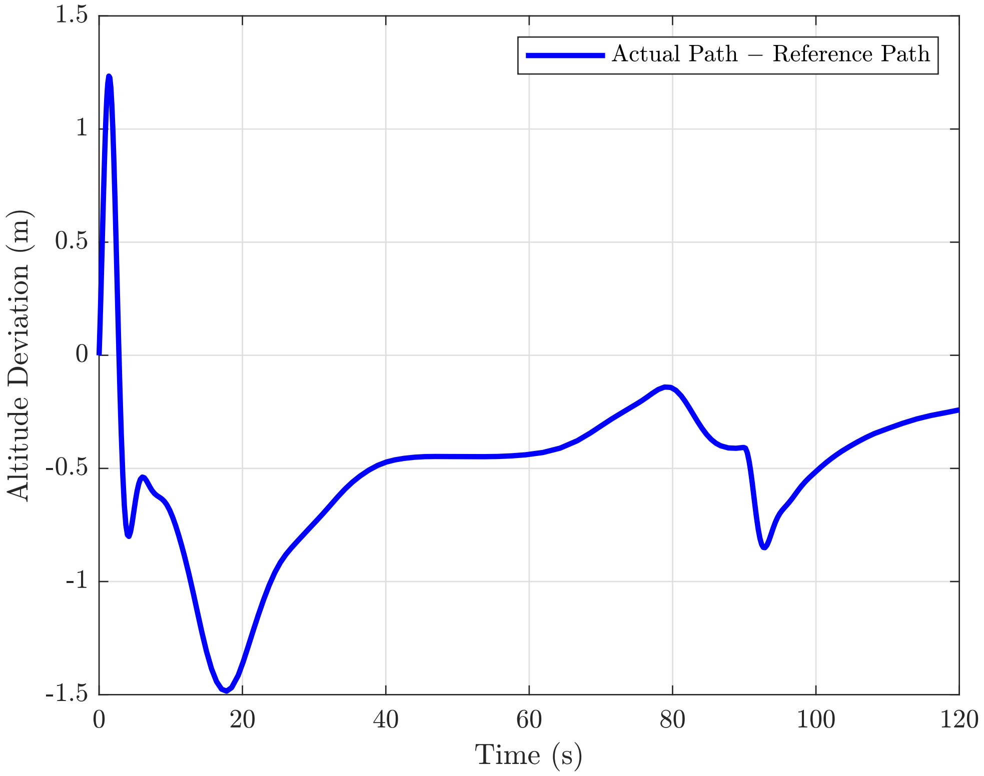
Figure 4.6: Windshear Effects



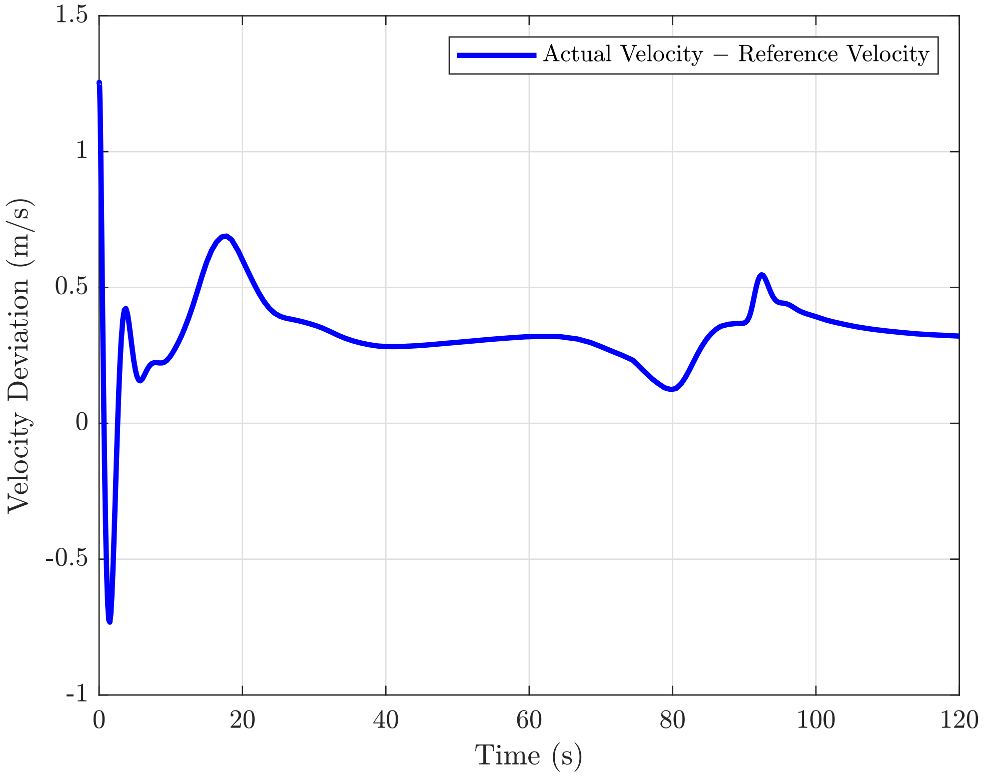
(A) Altitude Level

(B) Angle of Attack (α), Flight Path Angle (γ) and Pitch Angle (θ)

FIGURE 4.7: Case 2: Moderate Downburst

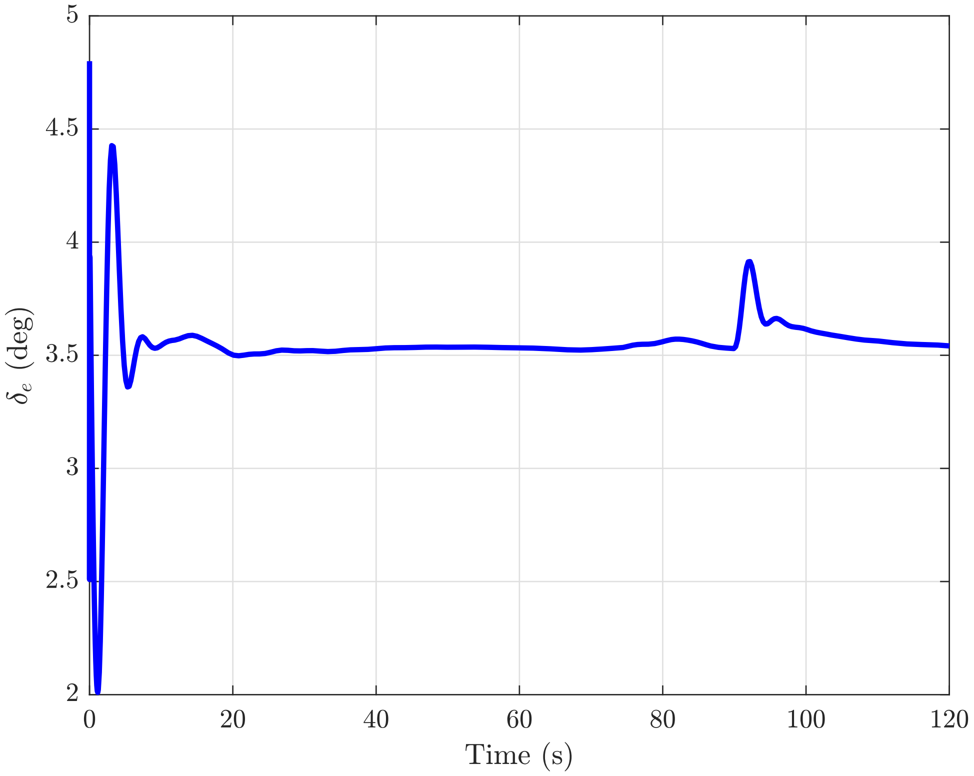


(c) Deviation between Actual Path and Reference Path

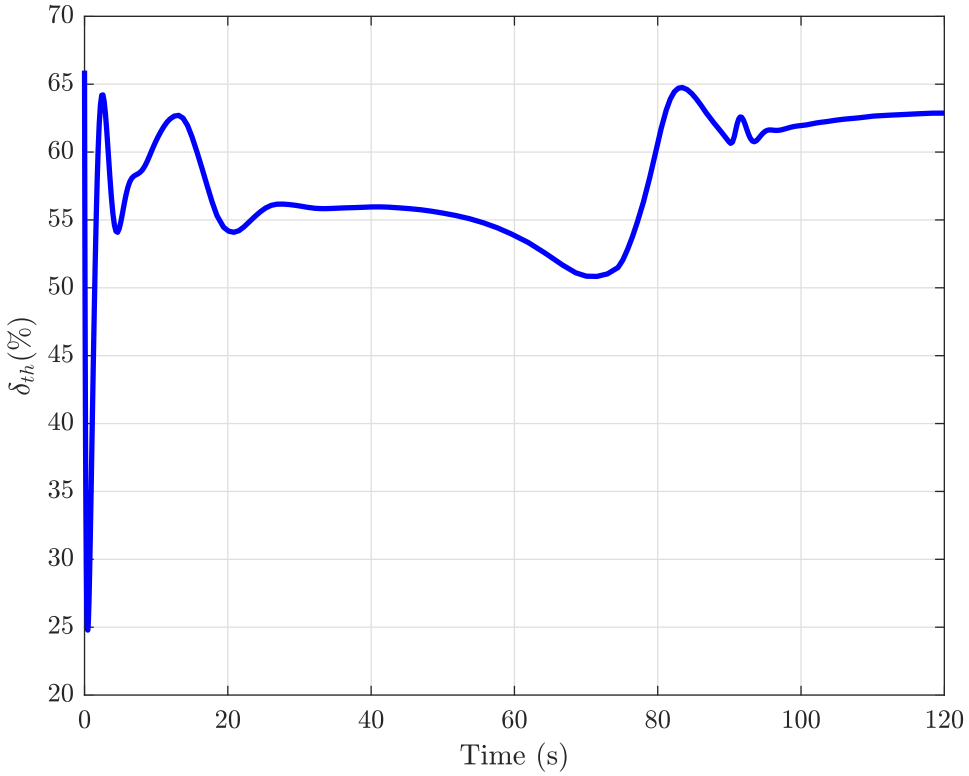


(d) Velocity Deviation from Trim Value

Figure 4.7: (Continued)

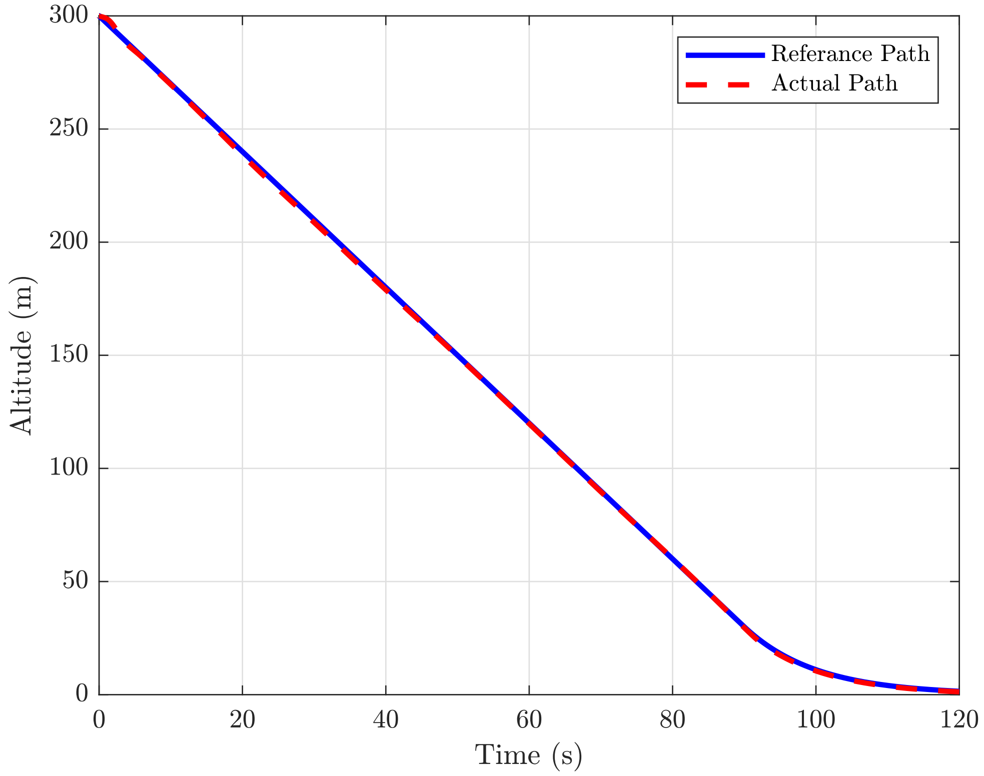


(E) Elevator Deflection

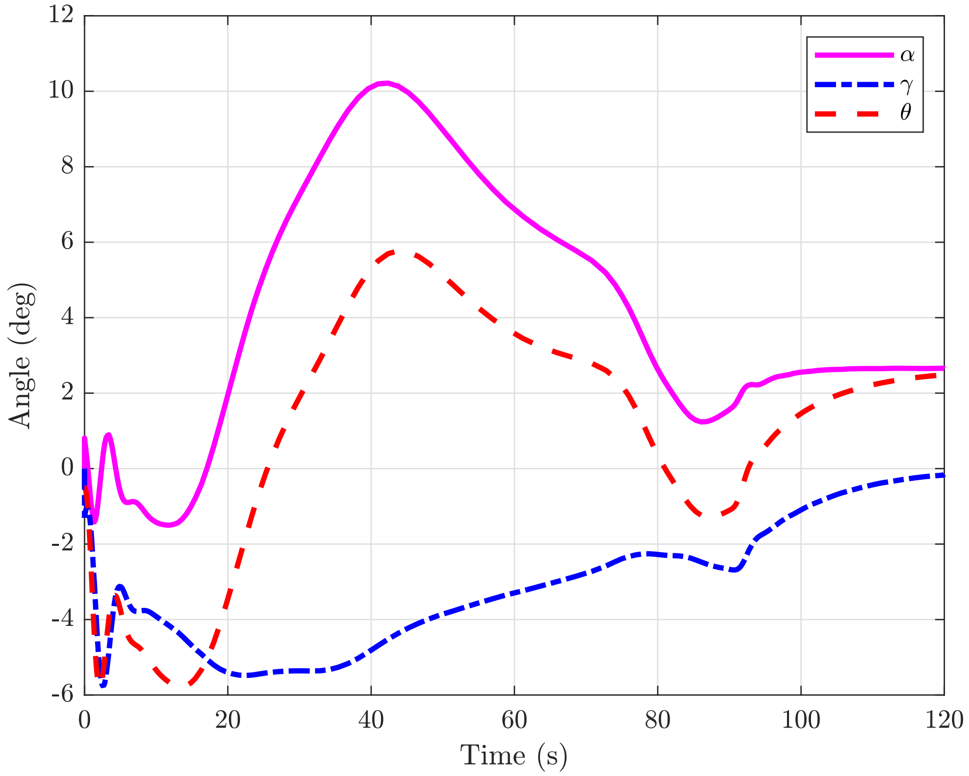


(F) Thrust Setting

FIGURE 4.7: (Continued)

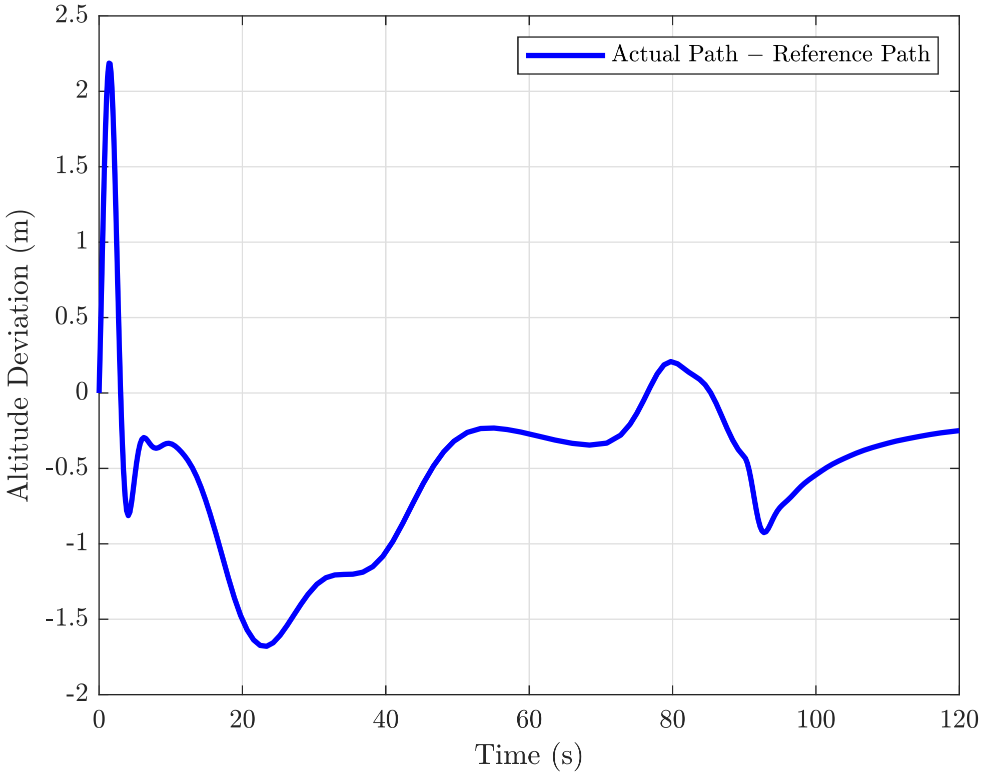


(A) Altitude Level

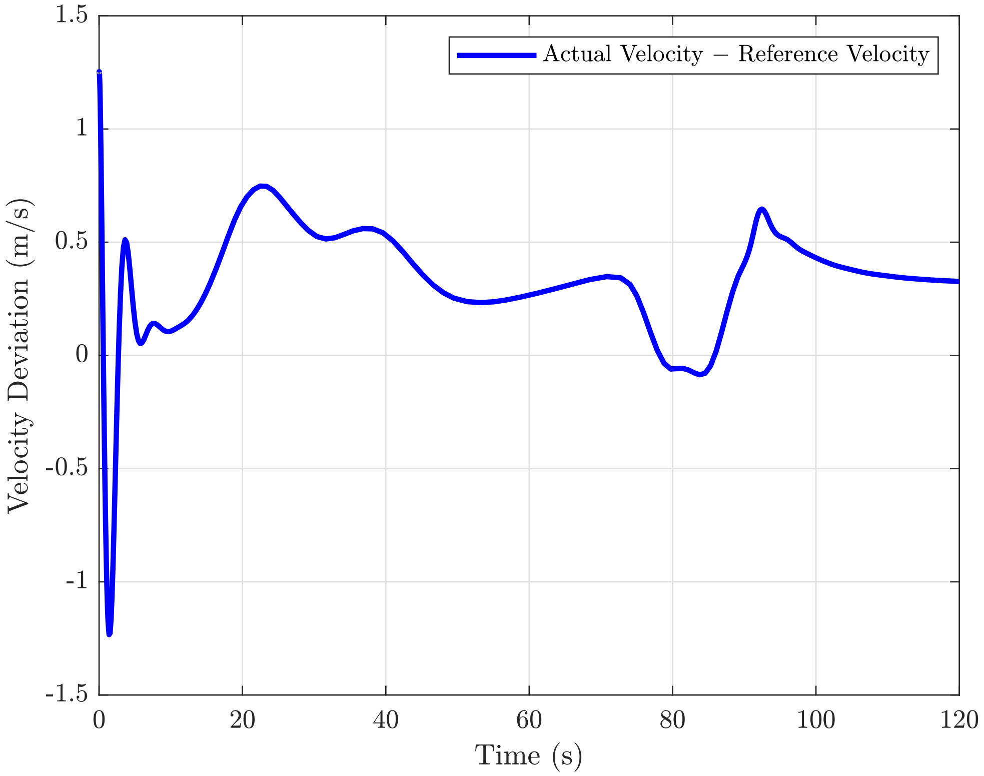


(B) Angle of Attack (α), Flight Path Angle (γ) and Pitch Angle (θ)

FIGURE 4.8: Case 2: Severe Downburst

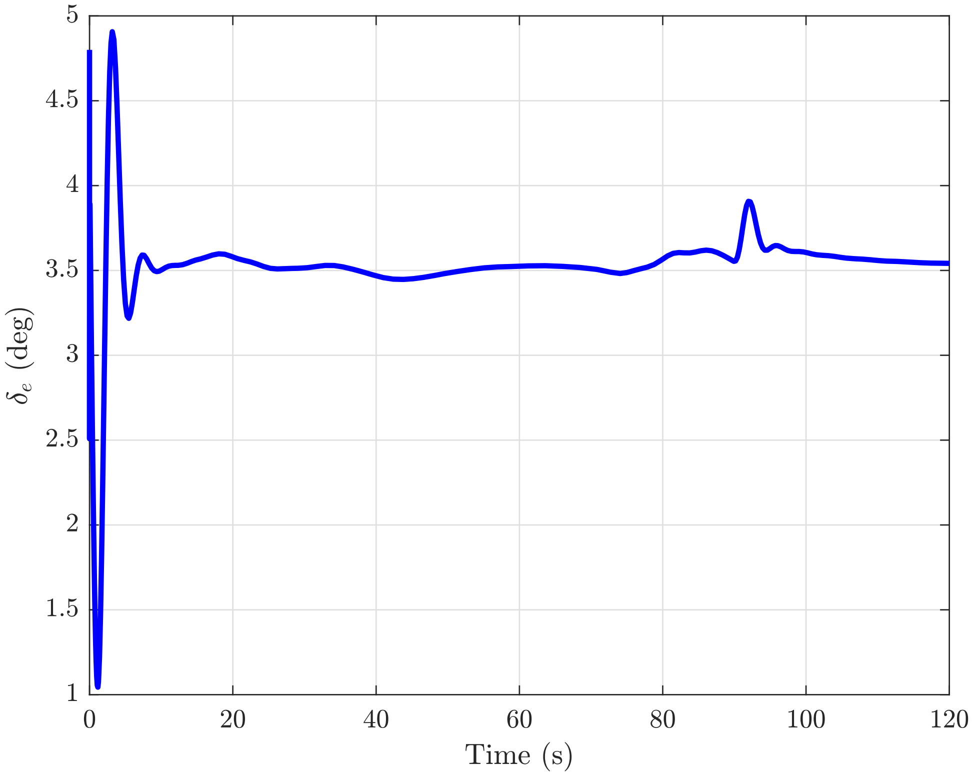


(C) Deviation between Actual Path and Reference Path

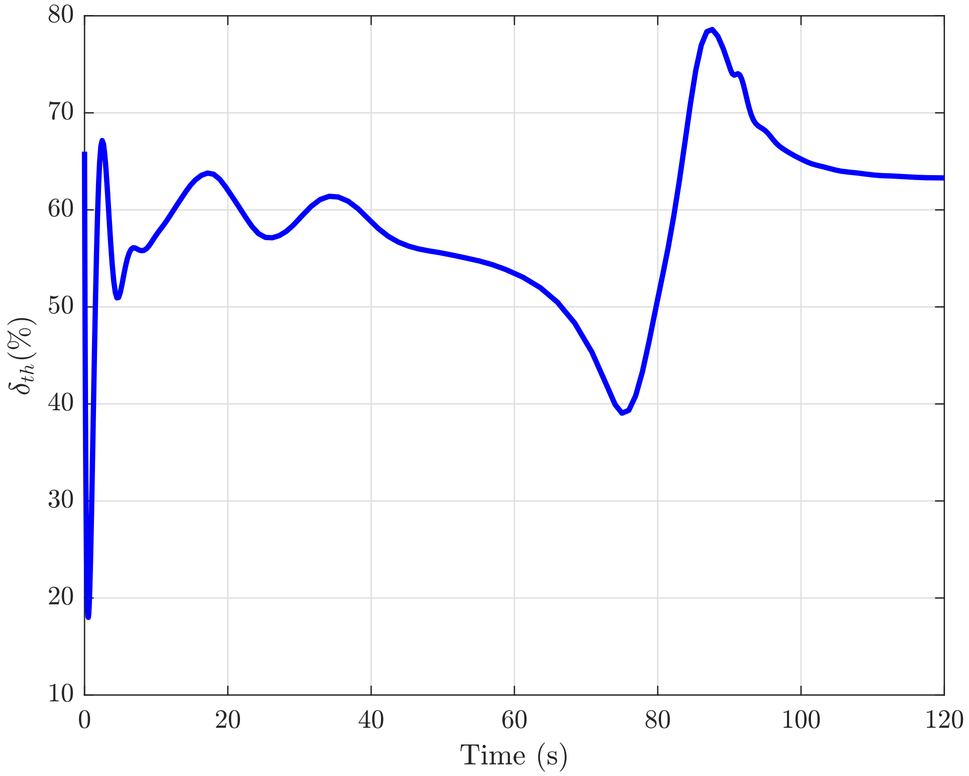


(D) Velocity Deviation from Trim Value

FIGURE 4.8: (Continued)



(E) Elevator Deflection



(F) Thrust Setting

FIGURE 4.8: (Continued)

vertical component of wind. As a result, the altitude and velocity return to the desired level with little deviation, as shown in Figures 4.7c and 4.7d, respectively. After 20 seconds, the headwind effects weaken, and the angle of attack decreases to maintain the desired altitude. The UAV starts encountering a tailwind after 50 seconds, increasing the longitudinal velocity and decreasing the angle of attack. The controller reduces the velocity and increases the angle of attack by manipulating the thrust demand and elevator deflection.

The flare phase begins after 90 seconds when the tailwind gradually loses its effects. Figure 4.5a shows that the UAV follows the reference trajectory closely during both glide and flare phases and performs a smooth touchdown. Figures 4.7e and 4.7f represent the elevator and throttle actuator displacements, which remain within limits since the UAV follows the reference landing trajectory. The controller is also tested under a severe downburst. The results in Figure 4.8 are similar to the moderate downburst case with higher magnitudes. The UAV closely follows the reference trajectory and the elevator deflection angle and throttle position remain within limits.

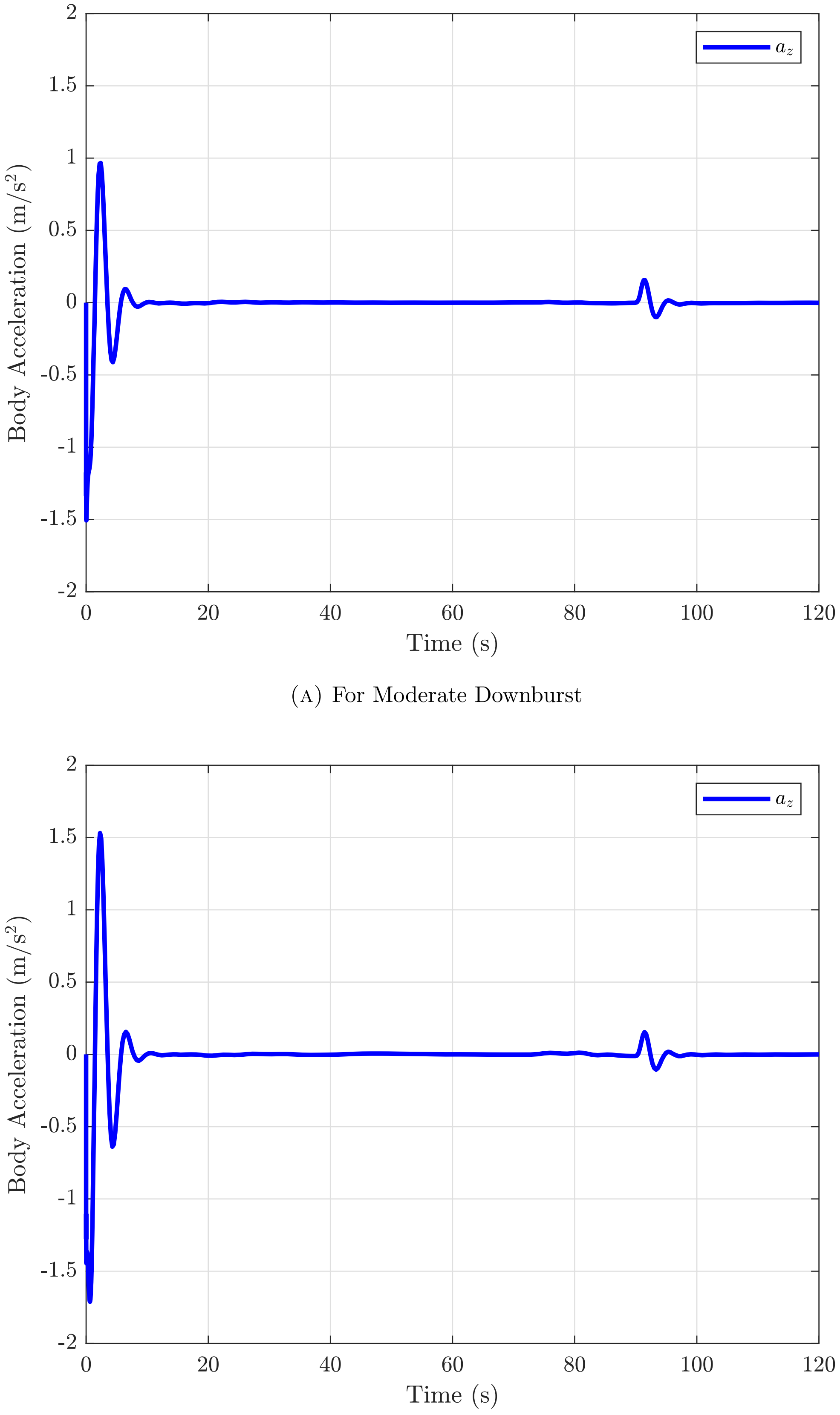
The proposed control methodology is compared with the study reported in [57]. Our UAV is different, so for a fair comparison, the wind load on both vehicles is calculated, and the controller performance is assessed under the severe downburst effects. The wind load F_{wind}, on the air vehicle is calculated by:

$$F_{\text{wind}} = \frac{1}{2}\rho V_{\text{wind}} S C_L \tag{4.16}$$

where ρ represents the air density, V_{wind} is the wind velocity, S is the wing reference area, and C_L is the lift coefficient. The acceleration a_{zw} along the z-axis, due to wind load is defined as:

$$a_{\text{zw}} = \frac{F_{\text{wind}}}{m} \tag{4.17}$$

The acceleration a_z of the test vehicle throughout the landing trajectory is given in Figure 4.9. Moreover, it is shown in Table 4.4 that the vehicle's acceleration (a_{zw})



(A) For Moderate Downburst

(B) For Severe Downburst

FIGURE 4.9: Body Acceleration along $z-$axis

TABLE 4.4: Comparison of Test Vehicle with Reference

Parameter	**Reference Vehicle** [57]	**Test Vehicle**	**Unit**
Mass	73482	350	kg
Wing reference area	201.6	6.5	m^2
Wind velocity	23	23	m/s
Wind load	63988	268.19	N
a_{zw}(wind)	0.871	0.766	$\mathrm{m/s}^2$
a_z(controller)	2.1	0.00627	$\mathrm{m/s}^2$

due to wind load is nearly the same for both systems. The controller did not react properly in the reference study, which worsened the situation. Contrary to this, our proposed controller reacts efficiently and cancels the wind effect significantly.

4.5 Summary

A different approach for landing fixed-wing UAVs is presented in this work. Glide and flare reference trajectories are predefined, and the UAV is forced to follow the altitude of the reference trajectory. First, the model is robustly stabilized using the normalized coprime factorization method, and the controller K_s and observer H_s are obtained. Then, the LQR-based inverse optimal problem is formulated and solved to design the cost function matrices, which are further utilized to realize MPC. The designed MPC inherits the small-signal properties (stability margin and closed-loop performance) of the H_∞ controller when the constraints are inactive (i.e., perturbation around equilibrium points that keep the system within saturation limits). Various scenarios are simulated and studied. The results demonstrate the proposed techniques' effectiveness and correctness even under moderate and severe wind shear effects. A qualitative analysis is also performed to compare the results with the benchmark work. The landing results demonstrated a significant improvement.

Chapter 5

Taxi Modeling and Control

Modeling and control of UAVs are currently one of the main research areas in industry and academia [74]. The flight control of UAVs in the air has been studied extensively. However, the research on the control of a UAV on-ground to keep it on the runway centreline remain limited. The on-ground dynamics and control are more complex due to the coupling between aerodynamics and tire-ground friction forces [75]. Moreover, the lateral control during the deceleration phase becomes more challenging due to the high initial longitudinal velocity at touch-down and uncertainty in the tire-ground friction model. These effects are significantly amplified in high wind or gust conditions. The controller could damage or lose the system if it does not perform well for all ground velocities. Therefore, an accurate LPV model and robust flight control system are required to steer the vehicle on the runway during take-off and landing taxi.

The chapter is organized as follows. In Section 5.1, a complete non-linear taxi model has been derived, including the aerodynamic forces and the interaction of the tires and runway. In Section 5.2, the model is analytically linearized and shows that it is explicitly dependent on the UAV's forward velocity, which continuously varies during both take-off and landing taxi. Then, an LPV model is derived for the taxi phase. The H_∞ controller is designed in Section 5.3. It is shown that a single controller designed at a particular velocity point does not satisfy the design requirements at all other velocity points. An LPV H_∞ controller is proposed to address this issue. In Section 5.4, the inverse optimal problem is solved to design LPV cost function matrices. Then, an H_∞ based LPV-MPC problem is formulated and solved in this section. The control law is successfully evaluated in the non-linear simulation environment. The comparative results of LPV H_∞ and H_∞ based LPV-MPC are presented in Section 5.4.4.

5.1 Non-Linear Taxi Modelling

This section establishes a non-linear taxi model for the UAV on the ground. A reference frame must be defined to develop motion equations for UAVs [76]. During the taxi phase, the UAV must interact with both atmosphere and the ground. The necessary reference frames relative to which the motion is considered are given in the coming subsection.

5.1.1 Reference Frame and Axes

Mathematical models for UAVs require a system of coordinates that describes their motion relative to that system. The coordinate systems required for the mathematical modeling of UAVs are discussed here.

5.1.1.1 Body Fixed Coordinate System

It is necessary to define the UAV's body-fixed coordinate system to describe its motion and orientation. It originates at the center of mass (center of gravity, CG) of the UAV with the x-axis aligned to the nose of the UAV, and the $x - z$ plane is symmetric. The y-axis is starboard, and the z-axis is downward. In Figure 5.1, x_b, y_b and z_b show the direction of the body-fixed axis frame. The rotations and translations of this coordinate system are directly linked to the motion of the UAV in air.

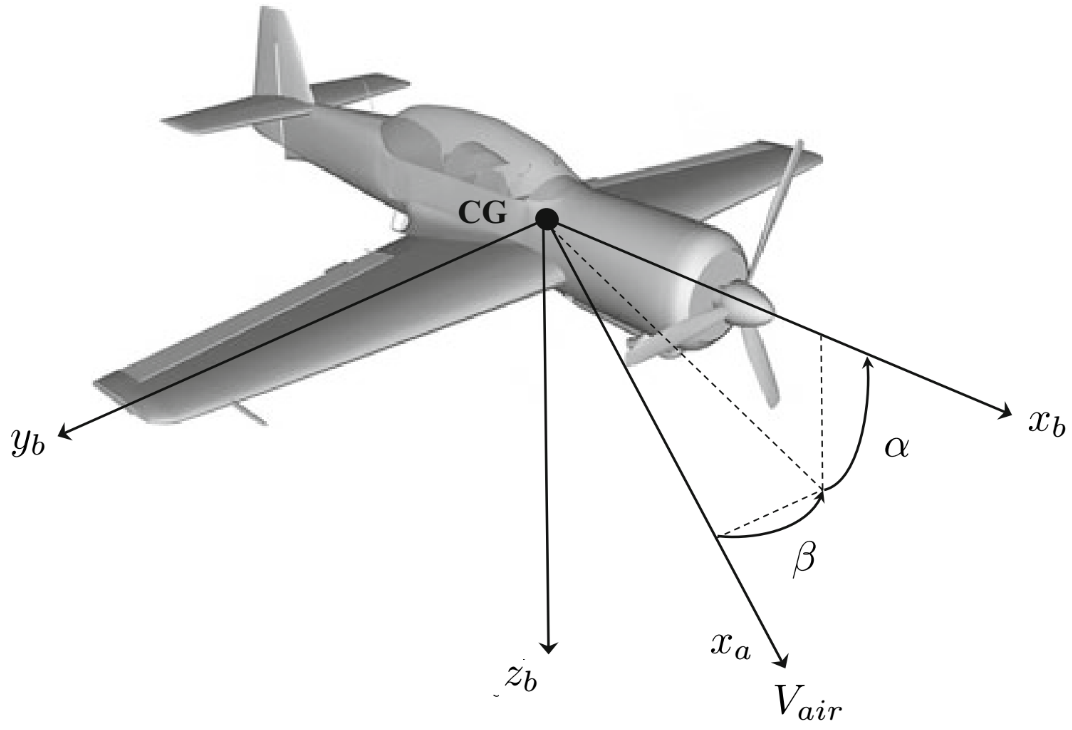


FIGURE 5.1: Body-fixed and Aerodynamics Coordinate Systems

5.1.1.2 Aerodynamic Coordinate System

This coordinate system is also linked with the body of the UAV. Its origin is at the center of the mass of the UAV and is associated with the orientation of the UAV's velocity vector relative to the air mass. As shown in Figure 5.1, its x-axis, (x_a) is aligned in the direction of the air velocity vector V_{air}. It can be transformed into the body coordinate system by rotating an angle α (angle of attack) around the y-axis and angle β (side-slip) around the z-axis.

5.1.1.3 Earth's Coordinate System

An earth reference frame is required to describe the UAV's motion on the runway. All the navigations of UAVs are described relative to the earth coordinate system. The distance covered by the UAV is small enough to approximate the earth as a flat surface. Figure 5.2 shows that the earth coordinate system is defined with its origin at the start of the point of the runway. The x-axis is aligned with the runway direction, the y-axis toward the right, and the z-axis downward. If the runway has a direction toward the north, then this axis can be called the north, east, and downward (NED axis).

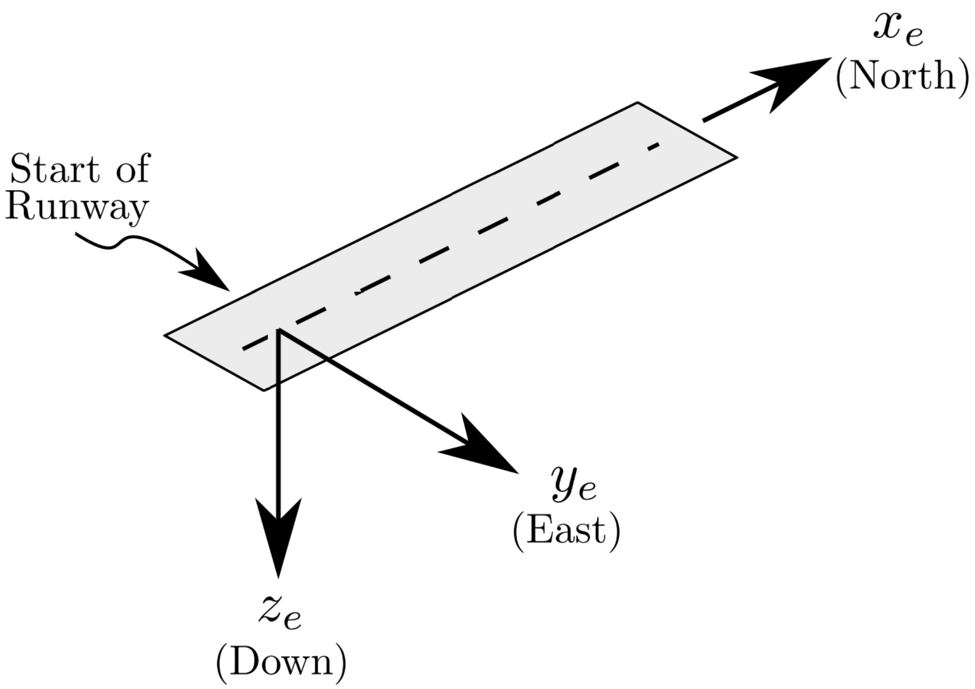


FIGURE 5.2: Earth Coordinate System

5.1.1.4 Wheels Coordinate Systems

It is a moving coordinate system associated with the UAV wheels. It originates at the contact point between the wheel and the ground, as shown in Figure 5.3. The x-axis is aligned to the direction of the wheel's motion, the y-axis is perpendicular to the wheel plan and positive to the right hand, and the z-axis is vertically downward. On a perfectly horizontal plane, a wheel coordinate system can be deduced from the earth's coordinate system by rotating around the vertical axis.

5.1.2 UAV Ground Dynamics

For the control design purpose, the UAV can be considered a rigid body with gravitational, engine, aerodynamic, and runway forces and moments acting upon it. This model has a three-wheel tricycle undercarriage, and its top view is shown

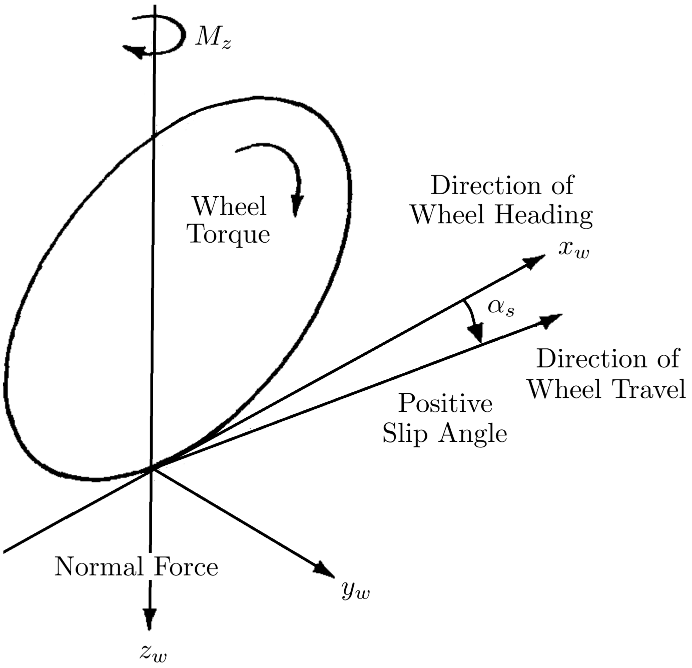


FIGURE 5.3: Wheel Coordinate System

in Figure 5.4, where l_n is the distance of the nose wheel from the CG, l_m is the distance of the main axle for CG, and l_w is the distance between the main wheels. The on-ground dynamics of a UAV are divided into three translational and three rotational components comprising 6-DoF. During the taxi or ground roll phase, it is assumed that there are no roll and pitch dynamics, so the motion is considered in the lateral plane only. The ground model presented in this paper has 3-DoF, two translational (along x and y-axes), and one rotational (yaw moment).

5.1.2.1 Equation of Motion

Newton's second law defines the translational motion of a UAV. The translational dynamic is described as [70] (p. 28):

$$\dot{v}_B = \frac{F_B}{m} - \omega_B \times v_B + Bg, \tag{5.1}$$

where v_B is the relative velocity of the UAV, ω_B is the angular velocity, $F_B = [F_x \ F_y \ F_z]^T$ is the force vector, m is the mass, g is gravitational acceleration, B is transformation matrix from NED to body coordinate and subscript B indicates

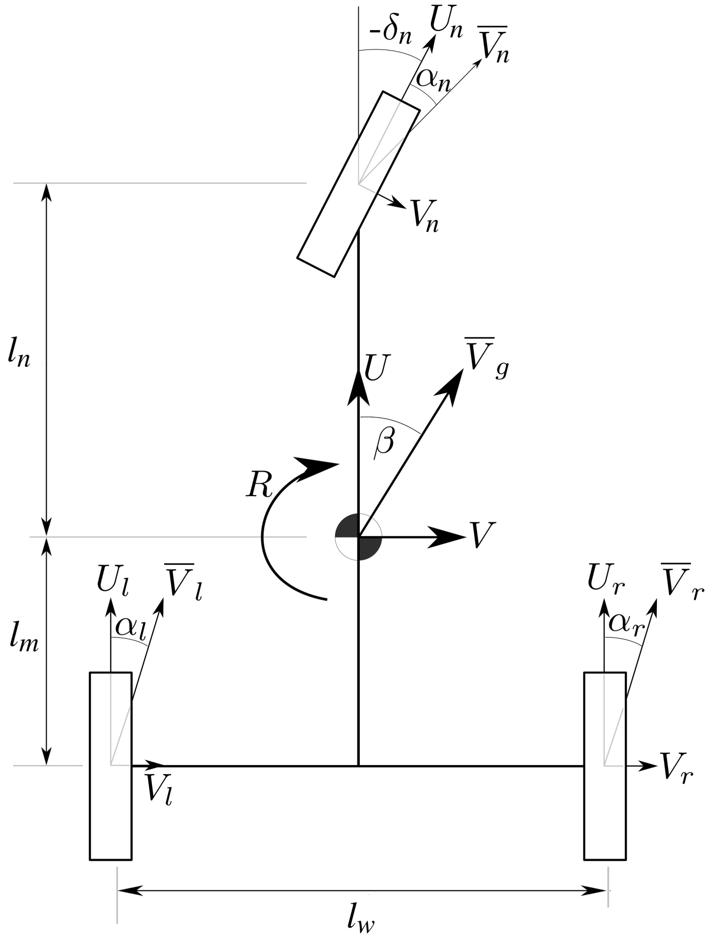


FIGURE 5.4: Three Wheel Tricycle Undercarriage

that all these vectors are expressed in body coordinate system. For definitions of body and NED frame, the reader is referred to [70]. The angular acceleration of the UAV is defined as [70] (p. 30):

$$\dot{\omega}_B = -J^{-1}\left(\omega_B \times J\omega_B\right) + J^{-1}T_B, \tag{5.2}$$

where $T_B = [T_x\ T_y\ T_z]^T$ is torque vector and J is the inertia matrix. Using the assumptions of constrained motion in the lateral plane, (5.1) and (5.2) can be written in component as:

$$\begin{aligned}\dot{U} &= RV + \frac{F_x}{m}\\ \dot{V} &= -RU + \frac{F_y}{m}\\ \dot{R} &= \frac{M_z}{J_z}\ ,\end{aligned} \tag{5.3}$$

where U and V are forward and lateral velocities expressed in body frame and R is yaw rate. The position of the UAV is determined in runway axes, in which x is aligned to the runway heading and y is in the lateral direction. The differential equations for the positions p_x, p_y, and yaw angle ψ are,

$$\begin{aligned}\dot{p}_x &= U\cos\psi - V\sin\psi\\ \dot{p}_y &= U\sin\psi + V\cos\psi\\ \dot{\psi} &= R.\end{aligned} \tag{5.4}$$

The next subsection defines the forces and moments acting upon the UAV.

5.1.3 Forces and Moments

The forces and moments that act on the UAV are aerodynamics, thrust, gravity, and ground reaction, which are described in this section. The control of the UAV is performed by manipulating these forces and moments.

5.1.3.1 Aerodynamic Forces and Moment

The aerodynamic forces and moments acting on the UAV are defined in terms of dimensionless aerodynamic coefficients:

$$\begin{aligned}F_x^a &= qS\left(C_{x_0} + C_x^{\alpha}|\alpha| + C_x^{\beta}|\beta| + C_x^{\delta_e}|\delta_e|\right)\\ F_y^a &= qS\left(C_y^{\beta}\beta + C_y^{\delta_r}\delta_r\right)\\ F_z^a &= qS\left(C_{z_0} + C_z^{\alpha}\alpha + C_z^{\delta_e}\delta_e\right)\\ M_z^a &= qS\left(C_z^{\beta}\beta + C_z^{\delta_r}\delta_r + \frac{b}{2V_0}C_z^{r}R\right)b,\end{aligned} \tag{5.5}$$

where F_x^a is the longitudinal aerodynamic drag depends upon angle of attack α, side-slip angle β and elevator deflection δ_e, F_y^a is the lateral aerodynamics force mainly created by rudder deflection δ_r, F_z^a is the lift force mainly generated by α

Table 5.1: Aerodynamic Coefficients of Test Vehicle

Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
C_{x_0}	-0.031	C_x^{α}	-0.088	C_x^{β}	-0.050
$C_x^{\delta_e}$	-0.010	C_y^{β}	-0.487	$C_y^{\delta_r}$	-0.097
C_{z_0}	-0.129	C_z^{α}	-3.368	$C_z^{\delta_e}$	-0.124
C_z^{β}	0.083	$C_z^{\delta_r}$	0.0225	C_z^{r}	-0.05

and δ_e, T_z^a is the aerodynamic yawing moment, $q = \frac{1}{2}\rho V_0^2$ is the dynamics pressure, S is the reference area of the UAV body, b is wing span, $V_0 = \sqrt{U^2 + V^2}$ is nominal flight speed and ρ is the air density. The aerodynamic coefficients of a test UAV are given in Table 5.1.

5.1.3.2 Thrust and Gravitational Forces

Engine thrust is the main force that produces acceleration in the forward direction. It is assumed that the thrust is acting along the body x-axis. The thrust force of a UAV can be adequately modeled as a first-order transfer function as [77]:

$$F_x^t = \frac{T_c}{\tau_T s + 1}, \tag{5.6}$$

where T_c is the commanded thrust and τ_T is the engine time constant.

The gravitational force at the CG of the UAV is along the z-axis is defined as:

$$F_z^G = \quad mg. \tag{5.7}$$

5.1.3.3 Ground Reaction Forces

These forces and moments are produced when the tires roll on the ground fraction between the tire and ground surface take the vehicle forward. The wheel coordinate system is used as reference frame. These forces have lateral and longitudinal

Table 5.2: Physical Parameters of Test Vehicle

Description	**Parameter**	**Value**
Mass	m	350 kg
z_B-axis moment of inertia	J_z	480 kg.m^2
Wing area	S	6.5 m^2
Wing span	b	6.6 m
Distance of nose wheel from CG	l_n	2.1 m
Distance of main wheels from CG	l_m	0.13 m
Distance of right wheel from the center of main wheel axle	l_r	0.7 m
Distance of left wheel from the center of main wheel axle	l_l	0.7 m

components, which merely depend on the normal load of the vehicle on the tires. The algebraic sum of the gravitational and lift forces are used to calculate the normal load on tires [77] as:

$$Z_n = (F_z^G + F_z^a)\left(\frac{l_m}{l_n + l_m}\right)$$

$$Z_l = Z_r = 0.5(F_z^G + F_z^a)\left(\frac{l_n}{l_n + l_m}\right),$$

where Z_n, Z_l, and Z_r are the normal load on the nose, left, and right wheels, respectively. These parameters for the test vehicle are presented in Table 5.2. During the movement on the ground, the forces between the tires and the ground are briefly described hereafter.

Cornering Forces

The cornering force appears when the velocity of the tire is no longer oriented to its plane (e.g while taking turn), and the wheel creates a side slip on the ground. It is due to the lateral velocity component (which is produced by the deformation of the tire) and friction force. The side-slip angles for the wheels of the UAV are defined as:

$$\begin{aligned}
\alpha_n = \tan^{-1}\left(\frac{V_n}{U_n}\right) &= \tan^{-1}\left(\frac{V + l_n R}{U}\right) - \delta_n \\
\alpha_l = \tan^{-1}\left(\frac{V_l}{U_l}\right) &= \tan^{-1}\left(\frac{V - l_m R}{U + l_l R}\right) \\
\alpha_r = \tan^{-1}\left(\frac{V_r}{U_r}\right) &= \tan^{-1}\left(\frac{V - l_m R}{U - l_r R}\right),
\end{aligned}$$

where V_n, V_l, and V_r are lateral, U_n, U_l, and U_r are longitudinal velocity components of the respective tire, $l_r = l_l = \frac{l_w}{2}$ are the distances of the left and right wheel from the center of the main axle, and δ_n is the nose wheel steering angle of the UAV. The mathematical functions that take into account the main characteristics of this force are based on side-slip angle, corner stiffness, and normal load on the tire [78] are,

$$\begin{aligned}
F_n^c &= -C_n^c \alpha_n Z_n, \\
F_l^c &= -C_l^c \alpha_l Z_l, \\
F_r^c &= -C_r^c \alpha_r Z_r \ ,
\end{aligned}$$

The corner stiffness coefficients C_n^c, C_l^c and C_r^c considered for the test UAV are defined in Table 5.3.

Ground Rolling Resistance

The major longitudinal force on a ground vehicle at level ground is the rolling resistance of the tires. It is the primary resistance force at low speeds and on hard pavement. The ground rolling resistance for the wheels of the UAV is defined as [78]:

$$\begin{aligned}
F_n^r &= -C_n^r Z_n, \\
F_l^r &= -C_l^r Z_l, \\
F_r^r &= -C_r^r Z_r \ .
\end{aligned}$$

The rolling resistance coefficients C_n^r, C_l^r and C_r^r considered for the test UAV are listed in Table 5.3.

Longitudinal Ground Forces

A friction force from the ground acts on the tires while accelerating (or decelerating). It appears when the tire's traveling speed is different from the tread velocity. It produces slip and friction between the tire and the ground. The slip ratio normally characterizes it. The slip ratio for the tires of UAVs is defined as:

$$\begin{aligned} s_n &= \frac{v_n^l - r_n\omega_n}{\max(v_n^l, r_n\omega_n)} \\ s_l &= \frac{v_l^l - r_l\omega_l}{\max(v_l^l, r_l\omega_l)} \\ s_r &= \frac{v_r^l - r_r\omega_r}{\max(v_r^l, r_r\omega_r)}, \end{aligned} \tag{5.8}$$

where v_n^l, v_l^l and v_r^l are longitudinal velocities at the axle of the wheels, ω_n, ω_l and ω_r are angular velocities and r_n, r_l and r_r are radius of respective tire. The experimental results have established in [78] that the longitudinal ground forces defined on the slip ratio, normal load on the tires, and the friction coefficient of the tire-road interface and model are described as:

$$\begin{aligned} F_n^l &= -C_n^l s_n Z_n, \\ F_l^l &= -C_l^l s_l Z_l, \\ F_r^l &= -C_r^l s_r Z_r \ , \end{aligned} \tag{5.9}$$

where C_n^l, C_l^l and C_r^l are the friction coefficients and defined in Table 5.3.

Braking Torque

During the taxi phase, differential braking is used to control the yawing moment of the UAV. Increasing the brake pressure at the rear right wheel compared to the left wheel creates a clockwise yawing moment and vise-versa. Braking torque is the function of brake friction coefficient C^b, brake pressure δ^b, breaking area of the wheel A_b, and braking radius r_b [78]. The braking torque on the main wheels is defined as:

$$\tau_l^b = r_b A_b C_l^b \delta_l^b, \qquad (5.10)$$
$$\tau_l^r = r_b A_b C_r^b \delta_r^b$$

For differential braking, inputs δ_l^b and δ_r^b are set as:

$$\begin{aligned} \text{if } \delta_b < 0 \text{ then } \delta_l^b &= 0, \quad \delta_r^b = -\delta_b \\ \text{else } \delta_l^b &= \delta_b, \quad \delta_r^b = 0. \end{aligned} \qquad (5.11)$$

for $\delta_b < 0$ right brake is applied, the left brake is kept at zero, and vice versa.

5.1.3.4 Wheels Dynamics

The longitudinal force and rolling resistance are directly related to the dynamics of the wheel through its rotational velocity. The dynamics for the wheels of a UAV can be defined as [78]:

$$\begin{aligned} \dot{\omega}_n &= -\frac{r_n}{J_n}\left(-F_n^r + F_n^l\right) \\ \dot{\omega}_l &= -\frac{1}{J_l}\left(-r_l F_l^r + r_l F_l^l + \tau_l^b\right) \\ \dot{\omega}_r &= -\frac{1}{J_r}\left(-r_r F_r^r + r_r F_r^l + \tau_r^b\right), \end{aligned} \qquad (5.12)$$

where J_n, J_l and J_r are moments of inertia of respective wheel and, τ_l^b and τ_r^b are the braking torques of the main left and right tires, respectively.

The overall ground forces and moment of the UAV are expressed along body axes as follows:

$$\begin{aligned} F_x^g &= (F_n^r + F_n^l)\cos\delta_n - F_n^c \sin\delta_n + F_l^r + F_r^r + F_l^l + F_r^l \\ F_y^g &= (F_n^r + F_n^l)\sin\delta_n + F_n^c \cos\delta_n + F_l^c + F_r^c \\ M_z^g &= l_n\left((F_n^r + F_n^l)\sin\delta_n + F_n^c\cos\delta_n\right) + l_r(F_l^r - F_r^r \\ &\quad + F_l^l - F_r^l) - l_m\left(F_l^c + F_r^c\right). \end{aligned} \qquad (5.13)$$

The total forces and moments acting upon the UAV are defined as:

$$\begin{aligned} F_x &= F_x^a + F_x^g + F_x^t \\ F_y &= F_y^a + F_y^g \\ F_z &= F_z^a + F_x^G \\ M_z &= M_z^a + M_z^g \ . \end{aligned} \tag{5.14}$$

A complete non-linear model of the UAV on the ground is obtained by combining the dynamics equations (5.3), kinematic equation (5.4), and wheel dynamics (5.12) with three inputs (nose wheel deflection δ_n, rudder deflection δ_r and differential braking of main wheels δ_b) and nine state variables as:

$$\begin{aligned} \dot{U} &= RV + \frac{F_x}{m} \\ \dot{V} &= -RU + \frac{F_y}{m} \\ \dot{R} &= \frac{M_z}{J_z} \\ \dot{p}_x &= U \cos\psi - V \sin\psi \\ \dot{p}_y &= U \sin\psi + V \cos\psi \\ \dot{\psi} &= R \\ \dot{\omega}_n &= -\frac{r_n}{J_n}\left(-F_n^r + F_n^l\right) \\ \dot{\omega}_l &= -\frac{1}{J_l}\left(-r_l F_l^r + r_l F_l^l + \tau_l^b\right) \\ \dot{\omega}_r &= -\frac{1}{J_r}\left(-r_r F_r^r + r_r F_r^l + \tau_r^b\right) , \end{aligned} \tag{5.15}$$

The next section derives an LPV model from the complete non-linear taxi model presented in 5.15.

5.2 Linear Parameter Variable Modelling

Another objective of this work is to develop an accurate but simplified LPV model of the UAV during the ground taxi phase. In this section, first the non linear model

TABLE 5.3: Cornering, Rolling and Friction Parameters of the Test Vehicle

Coefficient	**Value**	**Coefficient**	**Value**	**Coefficient**	**Value**
C_n^r	0.0141	C_n^c	5.6	r_n	0.080 m
C_l^r	0.0141	C_l^c	8	r_l	0.149 m
C_r^r	0.0141	C_r^c	8	r_r	0.149 m
C_n^l	4.2	J_n	0.006	C_l^b	9.25
C_l^l	12	J_l	0.051	C_r^b	9.25
C_r^l	12	J_r	0.051	r_b	0.04 m
				r_A	0.5 m^2

of the UAV is linearized analytically and shown that the linear model explicitly depends on the longitudinal velocity of the UAV. A full state space LPV model is constructed by a set of linear models which cover the whole taxi envelope (longitudinal velocities from state to take-off). Then, a reduced-ordered LPV model is presented for directional control of the UAV.

5.2.1 Full-State LPV Model

In the ground taxi phase, the UAV's velocity varies, unlike in the airborne phase, where the UAV flies at a constant speed most of the time. Therefore, no steady-state or equilibrium point exists in the longitudinal axis for linearisation. However, an equilibrium point in the lateral direction exists if lateral variables $V,\ R, \psi$ and inputs $\delta_r,\ \delta_n,\ \delta_b$ are kept zero. At each forward velocity U and corresponding steady-state wheel velocities $\frac{U}{R_n},\ \frac{U}{R_l},\ \frac{U}{R_r}$ the Jacobian matrices $A,\ B,\ C$ and D for the linearized model are obtained by differentiating the non-linear equation analytically. It gives explicit expressions for each term used in the Jacobian matrices and all non zero terms are given in Tables 5.4, 5.5 and 5.6 respectively. It comes out that each term in these matrices depends explicitly on longitudinal velocity as relations $U,\ U^2$ and/or $\frac{1}{U}$. These dependencies are used to define the explicit structure of the LPV model. In general, the full state parameter varying model of the taxi operation is written as:

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A\left(U\right) & B\left(U\right) \\ C\left(U\right) & D\left(U\right) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \tag{5.16}$$

$$\begin{aligned} A(U) &= A_0 + A_1 U + \frac{A_2}{U} \\ B(U) &= B_0 + B_1 U + B_2 U^2 \\ C(U) &= C_0 + \frac{C_1}{U} \\ D &= 0 \ , \end{aligned}$$

where $U \in [U_{\min}, U_{\max}]$, $U_{\min} > 0$, $U_{\max}$ is less than the take-off velocity, $x = [p_x \;\; p_y \;\; U \;\; V \;\; R \;\; \psi \;\; \omega_n \;\; \omega_l \;\; \omega_r]^T$ is the state vector, $u = [\delta_r \;\; \delta_n \;\; \delta_b]^T$ is the input vector, $y = [p_y \;\; \psi_e \;\; R]^T$ is the output vector of the system, and ψ_e is the effective yaw angle and is define as: $\psi_e = \psi + \beta$. As our control objective is to keep the UAV on centre line of runway (), the model is reduced to lateral dynamics in the next subsection.

5.2.2 Reduced Ordered Model

The full-state LPV model given above is reduced in this subsection to obtain a lateral model for the UAV for the directional control. It is evident from the structure of matrix B and C that the states corresponding to p_x, U, and ω_n can be removed to reduce the order of the system for control design purposes. For the differential input on the main wheels, we define $\omega := \omega_l - \omega_r$ and reduce one more state. Finally, (5.16) is transformed to a reduced-order linear model for direction control that consists of 5 states, $x = [p_y \;\; V \;\; R \;\; \psi \;\; \omega]^T$ with the same inputs and outputs as above. The state-space coefficient matrices of the reduced-ordered MIMO LPV model are given below:

Table 5.4: Coefficients of A Matrix

Matrix Element	**Expression**
$a_{13} = \dfrac{\partial \dot{p}_x}{\partial U}$	1
$a_{24} = \dfrac{\partial \dot{p}_y}{\partial V}$	1
$a_{26} = \dfrac{\partial \dot{p}_y}{\partial \psi}$	U_0
$a_{33} = \dfrac{\partial \dot{U}}{\partial U}$	$U_0(\frac{1}{m}(\rho S(C_{x_0} + C_x^{\delta_e}\|\delta_e\|) - \rho S(C_{z_0} + C_z^{\delta_e}\|\delta_e\|)(l_m C_n^r + 0.5 l_n(C_l^r + C_r^r))/l_L - \frac{1}{U_0}((C_n^l Z_n + C_l^l Z_l + C_r^l Z_r))$
$a_{37} = \dfrac{\partial \dot{U}}{\partial \omega_n}$	$\dfrac{1}{U_0}\left(C_n^l Z_n r_n/m\right)$
$a_{38} = \dfrac{\partial \dot{U}}{\partial \omega_l}$	$\dfrac{1}{U_0}\left(C_l^l Z_l r_l/m\right)$
$a_{39} = \dfrac{\partial \dot{U}}{\partial \omega_r}$	$\dfrac{1}{U_0}\left(C_r^l Z_r r_r/m\right)$
$a_{44} = \dfrac{\partial \dot{V}}{\partial V}$	$U_0(-0.5\rho S C_y^\beta)/m - \dfrac{1}{U_0}(C_n^c Z_n + C_l^c Z_l + C_r^c Z_r)/m$
$a_{45} = \dfrac{\partial \dot{V}}{\partial R}$	$-U_0$
$a_{54} = \dfrac{\partial \dot{R}}{\partial V}$	$U_0(0.5\rho S M_z^\beta S_\omega)/J_z + \dfrac{1}{U_0}(-l_n C_n^c Z_n + l_m C_l^c Z_l + l_m C_r^c Z_r)/J_z$
$a_{55} = \dfrac{\partial \dot{R}}{\partial R}$	$\dfrac{1}{U_0}(-l_n^2 C_n^c Z_n + l_r^2 C_l^l Z_l - l_r^2 C_r^l Z_r - l_m^2 C_l^c Z_l - l_m^2 C_r^c Z_r)/J_z$
$a_{58} = \dfrac{\partial \dot{R}}{\partial \omega_l}$	$\dfrac{1}{U_0}\left(C_l^l Z_l R_l l_r\right)/J_z$
$a_{59} = \dfrac{\partial \dot{R}}{\partial \omega_r}$	$-\dfrac{1}{U_0}\left(C_r^l Z_r R_r l_r\right)/J_z$
$a_{65} = \dfrac{\partial \dot{\psi}}{\partial R}$	1
$a_{73} = \dfrac{\partial \dot{\omega}_n}{\partial U}$	$U_0(r_n C_n^r \rho S l_m(C_{z_0} + C_z^{\delta_e}\|\delta_e\|)/(J_n l_L) - \dfrac{1}{U_0}(C_l^n Z_n r_n)/J_n$
$a_{77} = \dfrac{\partial \dot{\omega}_n}{\partial \omega_n}$	$-\dfrac{1}{U_0}\left(C_n^l Z_n r_n\right)/J_n$
$a_{83} = \dfrac{\partial \dot{\omega}_l}{\partial U}$	$U_0(r_l C_l^r \rho S l_n(C_{z_0} + C_z^{\delta_e}\|\delta_e\|)/(J_l l_L) - \dfrac{1}{U_0}(C_l^l Z_l r_l/J_l)$
$a_{85} = \dfrac{\partial \dot{\omega}_l}{\partial R}$	$\dfrac{1}{U_0}\left(r_l C_l^l Z_l l_r\right)/J_l$
$a_{88} = \dfrac{\partial \dot{\omega}_l}{\partial \omega_l}$	$-\dfrac{1}{U_0}\left(r_l^2 C_l^l Z_l\right)/J_l$
$a_{93} = \dfrac{\partial \dot{\omega}_r}{\partial U}$	$U_0(r_r C_r^r \rho S l_n(C_{z_0} + C_z^{\delta_e}\|\delta_e\|)/(J_r l_L) - \dfrac{1}{U_0}(C_l^r Z_r r_r/J_r)$
$a_{95} = \dfrac{\partial \dot{\omega}_r}{\partial R}$	$-\dfrac{1}{U_0}\left(r_r C_r^l Z_r l_r\right)/J_r$
$a_{99} = \dfrac{\partial \dot{\omega}_r}{\partial \omega_r}$	$-\dfrac{1}{J_r U_0}\left(r_r^2 C_r^l Z_r\right)/J_r$

Table 5.5: Coefficients of B Matrix

Matrix Element	**Expression**
$b_{41} = \dfrac{\partial \dot{V}}{\partial \delta_r}$	$U_0^2 \left(0.5 C_y^{\delta_r} \rho S/m\right)$
$b_{42} = \dfrac{\partial \dot{V}}{\partial \delta_n}$	$\dfrac{F_n^r + F_n^l + C_n^c Z_n}{m}$
$b_{51} = \dfrac{\partial \dot{R}}{\partial \delta_r}$	$\dfrac{C_N^{\delta_r} q S b}{J_z}$
$b_{52} = \dfrac{\partial \dot{R}}{\partial \delta_n}$	$U_0^2 \left(l_n (F_n^r + F_n^l + C_n^c Z_n)\right)/J_z$
$b_{83} = \dfrac{\partial \dot{\omega}_l}{\partial \delta_{lb}}$	$-\dfrac{C_l^b A_b r_b}{J_l}$
$b_{93} = \dfrac{\partial \dot{\omega}_r}{\partial \delta_{rb}}$	$-\dfrac{C_r^b A_b r_b}{J_r}$

Table 5.6: Coefficients of C Matrix

Matrix Element	**Expression**
$c_{12} = \dfrac{\partial y_p}{\partial y_p}$	1
$c_{24} = \dfrac{\partial \psi_e}{\partial V}$	$\dfrac{1}{U_0}$
$c_{26} = \dfrac{\partial \psi_e}{\partial \psi}$	1
$c_{35} = \dfrac{\partial R}{\partial R}$	1

$$A_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.055 & 0 \\ 0 & 0.085 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -3.6275 \end{bmatrix}$$

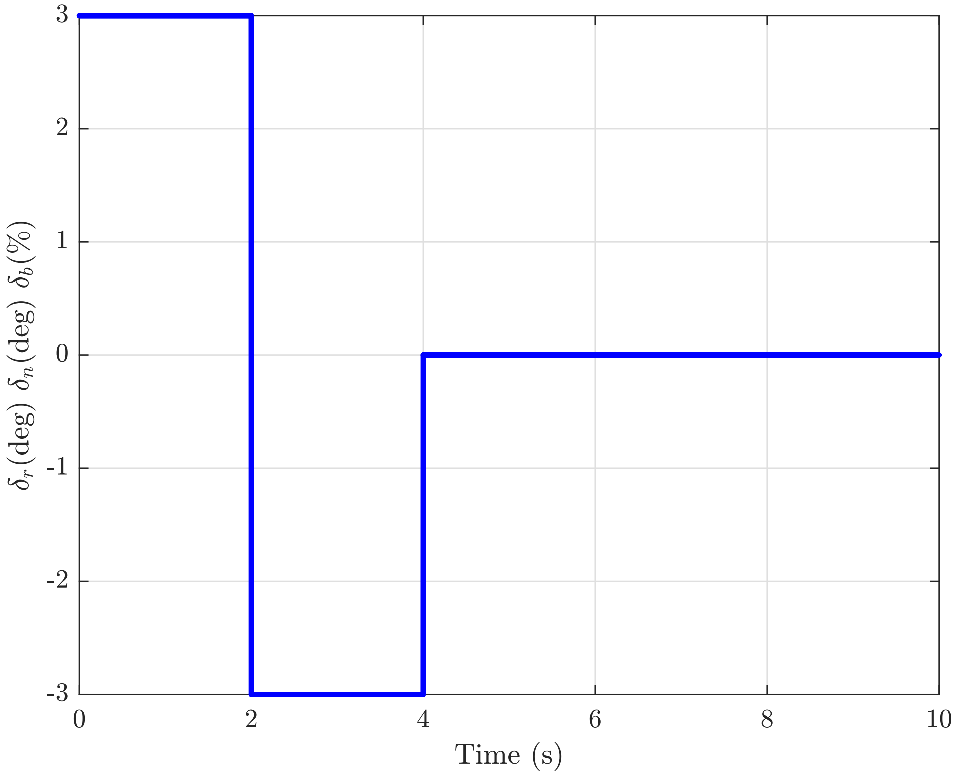
$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0.030 & -1 & 0 & 0 \\ 0 & 0.004 & 0.022 & 0 & -0.002 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -35.05 & 0 & 3.8748 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -77.1 & 0 & 0 & 0 \\ 0 & 2.06 & -47.8 & 0 & 4.073 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 76405 & 0 & -8445.3 \end{bmatrix}, \quad B_2 = 10^{-5} \begin{bmatrix} 0 & 0 & 0 \\ -2 & -3 & 0 \\ 2 & -4 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

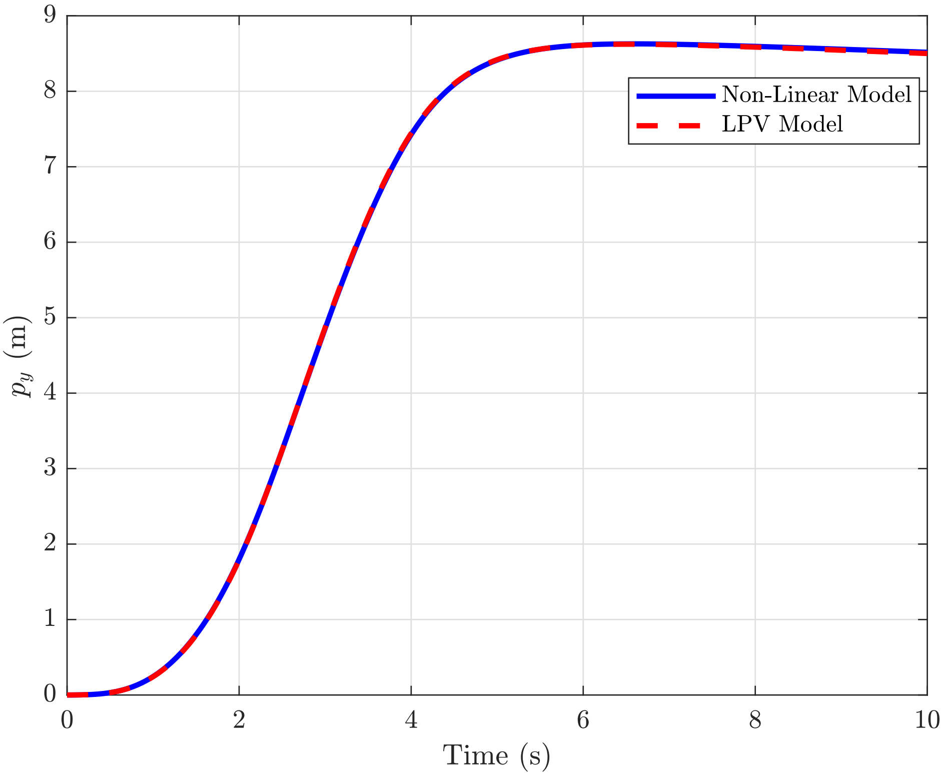
$$C_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 57.3 & 0 \\ 0 & 0 & 57.3 & 0 & 0 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 57.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In this model, the inputs $\delta_r, \delta_n, \delta_b$ have units deg, deg, and % respectively, and outputs p_y, ψ_e, R have meters, deg, and deg/sec respectively.

A simulation is developed to compare the simplified LPV model with the non-linear model to validate it. The same input signal is applied at δ_r, δ_n and δ_b, and is shown in Figure 5.5a. The simulation results in Figure 5.5 show that the constructed LPV model is in high fidelity with the established non-linear model because we have considered the non-linear dependencies of the model in the scheduling parameters of the LPV model. Moreover, the quantitative comparison of both (Non-linear and LPV) models is done by calculating the NRMSE between the outputs. The NRMSEs between the outputs are 0.9972, 0.9929, and 0.9865 (1 for 100% match and 0 for no match) for cross distance (p_y), yaw (ψ), and yaw rate (R) respectively. The error between both models is presented in 5.6.

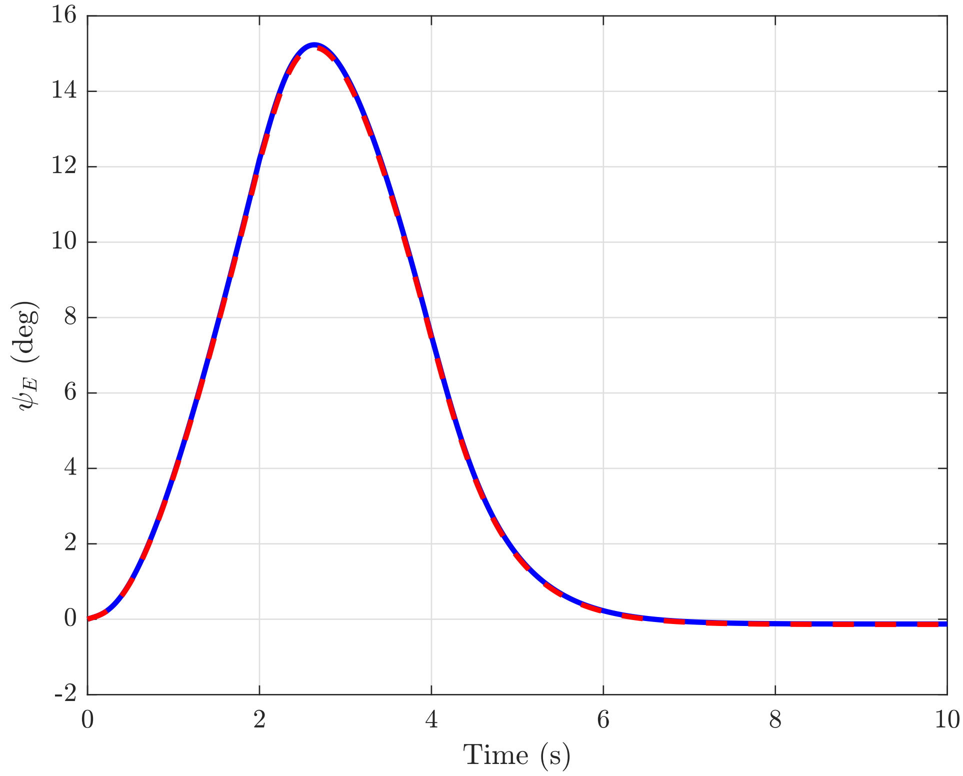


(A) Input Setting

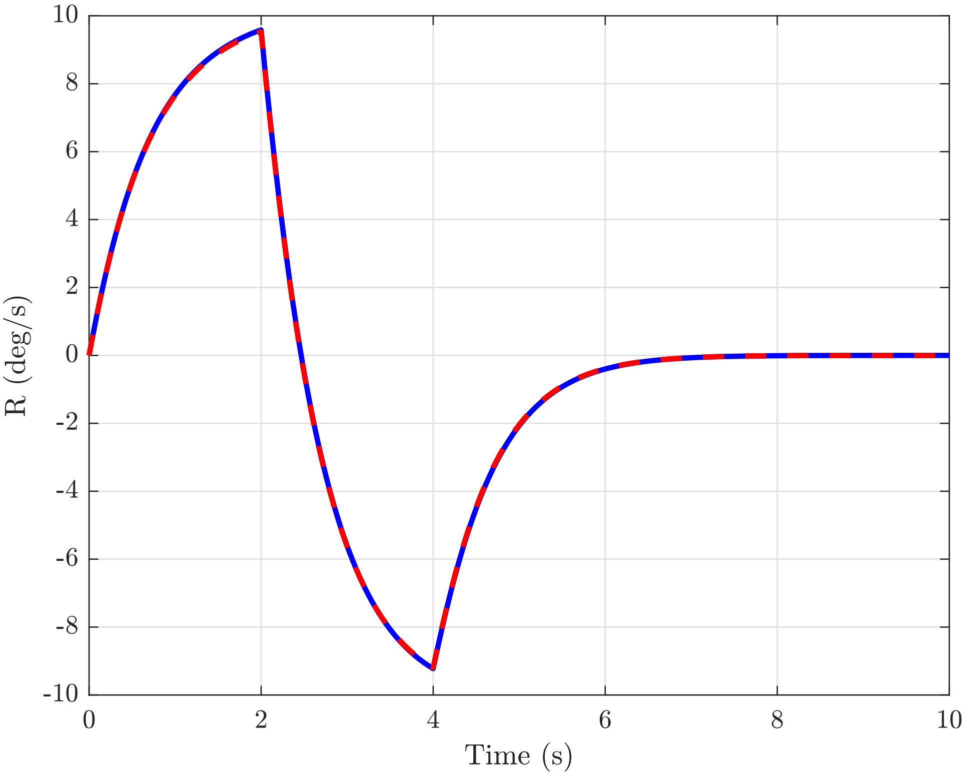


(B) Cross Distance (p_y)

FIGURE 5.5: Comparison of the Non-Linear and Reduced Ordered LPV Model

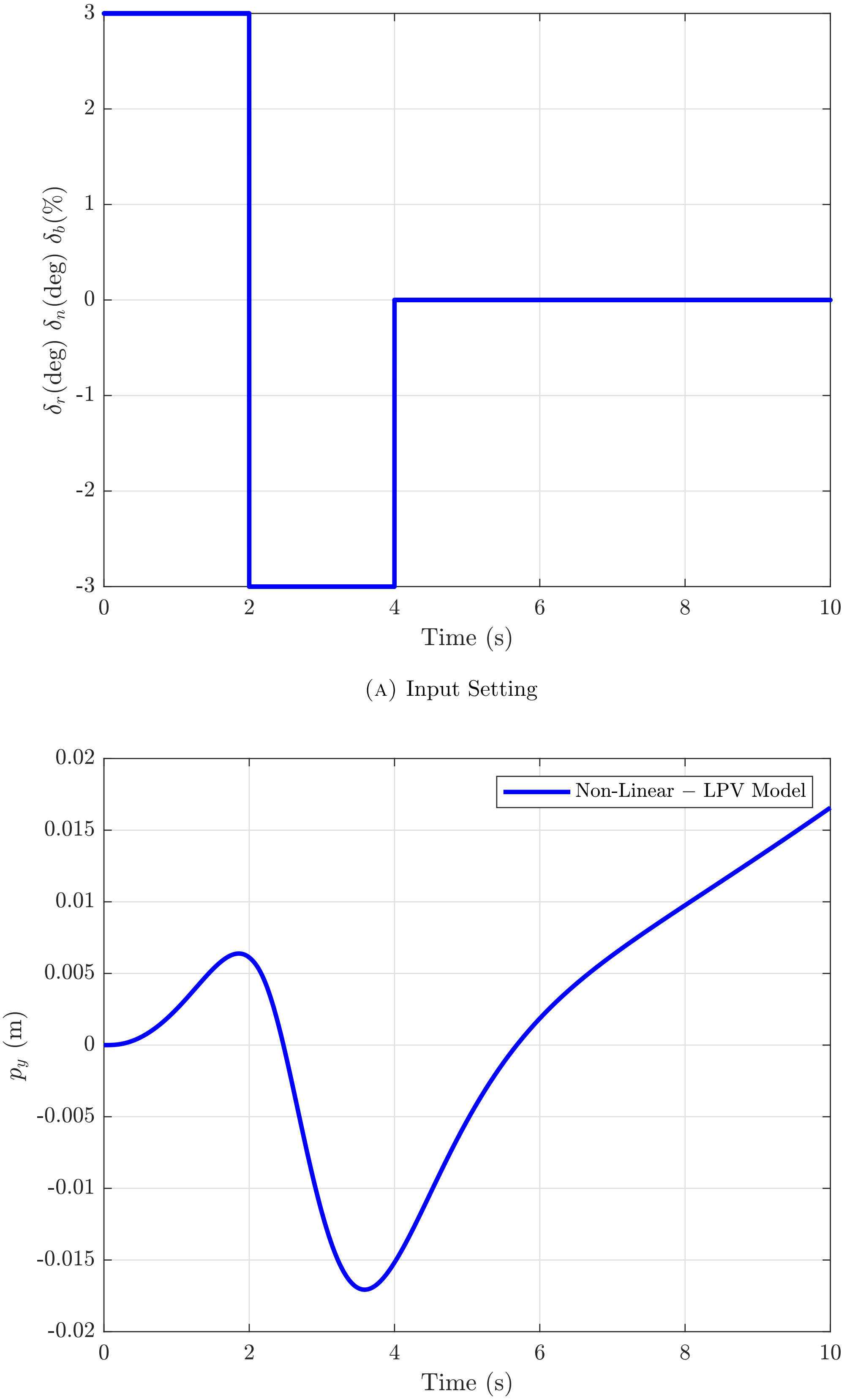


(c) Yaw (ψ)



(d) Yaw Rate (R)

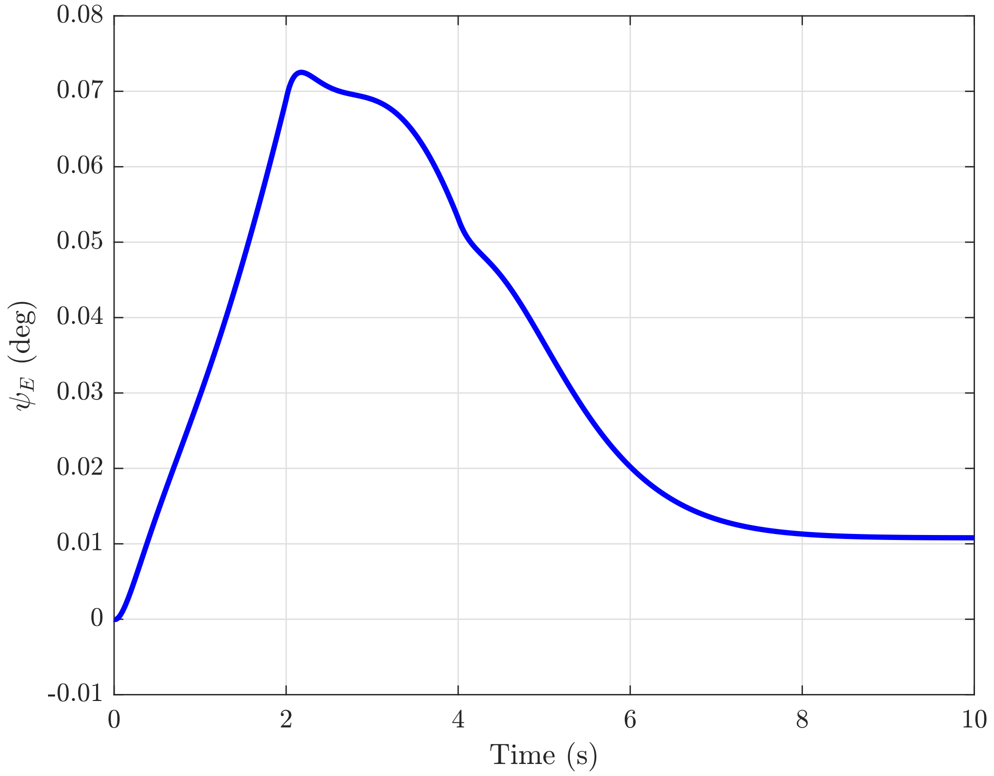
Figure 5.5: (Continued)



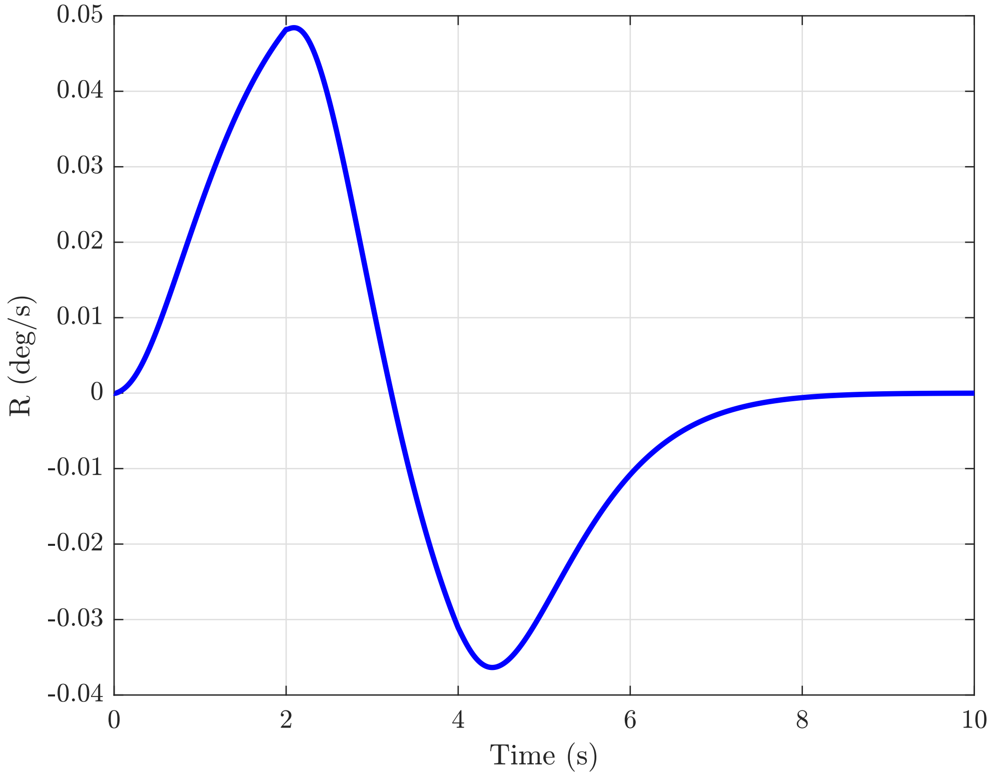
(A) Input Setting

(B) Error in Cross Distance

FIGURE 5.6: Error Between the Non-Linear and Reduced Ordered LPV Model



(c) Error in Yaw



(d) Error in Yaw Rate

Figure 5.6: (Continued)

An apparently obvious approach is to use LPV synthesis for the LPV model, but [79] shows that the computational complexities increase with the number of varying parameters. To avoid these complexities, the H_∞ based LPV-MPC design framework is proposed to control the directional offset in UAV's position on the runway. As a first step, the LPV H_∞ controller is designed and tested, and presented in the next section.

5.3 LPV H_∞ Control Design

In this section, the LPV H_∞ controller is designed for the directional taxi control of the UAV on ground. To illustrate the design methodology, first the H_∞ loop shaping controller is designed for fixed velocity point, e.g., $U = 30\text{m/s}$. Then, the procedure for obtaining parameter varying controller $K_s(U)$ and observer $H_s(U)$ gains is explained.

5.3.1 Design Specifications

The primary objective of the lateral taxi control is to keep the UAV on the central line of the runway by driving the cross distance p_y to zero smoothly by manipulating all three inputs δ_r, δ_n, and δ_b. The overall control design requirements are as follow:

- The response should have little or no overshoot.
- 90% of the desired cross-track ($p_y = 0$) should achieved in $7 \sim 8$ sec.
- The robustness index γ should be less than 4.

By keeping these design specifications, first, the controller was developed for a fixed velocity to explain the methodology and then provide steps and structure of LPV H_∞ controller.

5.3.2 Design Steps for a Fixed Velocity Plant

The reduced order vehicle model $G_{\text{v}} : C \mapsto C^{3\times3}$ at $U = 30$ m/s is used to describe the design methodology. To include the effects of the actuators and sensor dynamics, pre and post-multiply the vehicle model G_{v} as:

$$G_{\text{p}} = G_{\text{act}}\, G_{\text{v}}\, G_{\text{sen}}, \tag{5.17}$$

where $G_{\text{act}} \in C^{3\times3}$ is the diagonal matrix having actuator dynamics. Sensor lags are mainly due to the integration of the Inertial Navigation System (INS) and Global Positioning System (GPS) and are incorporated in $G_{\text{sen}} \in C^{3\times3}$. To meet the desired specifications following pre and post-compensators are selected,

$$\begin{aligned} W_1 &= \begin{bmatrix} \dfrac{1.5(s+2)}{s+0.1} & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & -9 \end{bmatrix}, \\ W_2 &= \begin{bmatrix} \dfrac{0.5(s+3)}{s+1.5} & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.18}$$

The resulting shaped plant is obtained as:

$$G_{\text{sp}} = W_1\, G_{\text{p}}\, W_2 \tag{5.19}$$

After including the effects of actuators, sensor delays, and shaping of the vehicle model, the dimension of the shaped state vector has increased to 14. The updated state-space model is defined as:

$$\begin{aligned} \dot{x}_s &= A_s x_s + B_s u_s \\ y &= C_s x_s \end{aligned} \tag{5.20}$$

where A_s, B_s, and C_s are shaped state-space matrices and given below.

$$A_s = \begin{bmatrix}
0.9704 & 0.1339 & 0 & 0 & -0.0240 & -0.0003 & 0 & -0.0078 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.8187 & 0 & 0 & 0.0725 & 0.0007 & 0 & 0.0225 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.8187 & 0 & 0 & 0.1361 & 0.0005 & 4.1544 & 0 & -0.0001 & -0.0002 & 0.0001 & 0.0003 & 0 \\
0 & 0 & 0 & 0.3679 & 0 & 0.0118 & 5.7585 & 0 & 0.0021 & 0.0030 & 0.0091 & 0.0076 & 0.0231 & 0.0003 \\
0 & 0 & 0 & 0 & 1 & 0.0197 & 0.0001 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9660 & -0.5868 & 0 & -0.0002 & -0.0012 & -0.0030 & 0.0011 & 0.0020 & -0.0001 \\
0 & 0 & 0 & 0 & 0 & 0.0035 & 0.9905 & 0 & 0.0005 & 0.0011 & 0.0025 & 0.0027 & 0.0063 & 0.0001 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0199 & 1 & 0 & 0 & 0 & 0 & 0.0001 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.0223 & 8.6494 & 0 & 0.0403 & 0.0061 & 0.0169 & 0.0156 & 0.0429 & 0.0006 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0543 & -0.4760 & 0 & 0 & 0.1258 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4874 & 0.5550 & 0 & 0 & 0.1177 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0543 & -0.4760 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4874 & 0.5550 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9980
\end{bmatrix}$$

$$B_s = \begin{bmatrix} 0 & 0 & 0 & 0.0002 & 0 & -0.0001 & 0.0001 & 0 & 0.0005 & 0.0939 & 0.0867 & 0 & 0 & 0.0275 \\ 0 & 0 & 0 & 0.0002 & 0 & 0 & 0.0001 & 0 & 0.0004 & 0 & 0 & 0.0305 & 0.0285 & 0 \\ 0 & 0 & 0 & 0.0006 & 0 & -0.0001 & 0.0002 & 0 & 0.1916 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_s = \begin{bmatrix} 0.5076 & 2.5381 & 0 & 0 & -0.5076 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.5 & 0 & 0 & -2.8648 & 0 & -85.9437 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12.5 & 0 & 0 & -57.2958 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$K_s = \begin{bmatrix} 0.3662 & 0.4562 & 1.3835 & 0.2725 & 10.5550 & 18.5143 & 218.72 & 942.31 & 0.1040 & 0.4686 & 0.6035 & 1.3558 & 1.7432 & 2.4651 \\ 0.2378 & 0.2954 & 0.8826 & 0.1714 & 6.8192 & 11.6712 & 141.34 & 604.32 & 0.0672 & 0.3034 & 0.3909 & 0.8746 & 1.1254 & 1.3179 \\ 0.1485 & 0.1846 & 0.5526 & 0.1113 & 4.2645 & 7.2774 & 89.28 & 378.17 & 0.0425 & 0.1911 & 0.2471 & 0.5502 & 0.7101 & 0.8199 \end{bmatrix}$$

$$H_s = \begin{bmatrix} 0.0059 & 0.0028 & 0.0012 & 0 & 0.0072 & 0 & 0 & 0.0001 & 0 & 0 & 0.0001 & 0 & 0 & 0.0002 \\ 0.0029 & 0.0027 & 0.0056 & 0.0003 & 0.0075 & -0.0019 & 0 & 0.0003 & 0.0003 & -0.0001 & 0.0020 & 0 & 0 & 0.0074 \\ 0.00007 & 0.0001 & 0.0011 & 0.0069 & 0.0005 & -0.0059 & 0.0008 & 0.0003 & 0.0068 & 0.0003 & 0.0057 & 0.0001 & -0.0002 & 0.0238 \end{bmatrix}^T$$

The singular values of the plant and shaped plant are shown in Figure 5.7a. Following the procedure of H_∞ loop shaping control, as explained in Section 3.2.1, the controller K_s and observer H_s are synthesized and given above. The shaped plant's singular values and the controller are also shown in Figure 5.7a. The step response given in Figure 5.7b shows that the controller also meets the desired rise time specification. The normalized coprime factor robust optimization gives $\gamma = 3.7 < 4$ which meets the desired criteria.

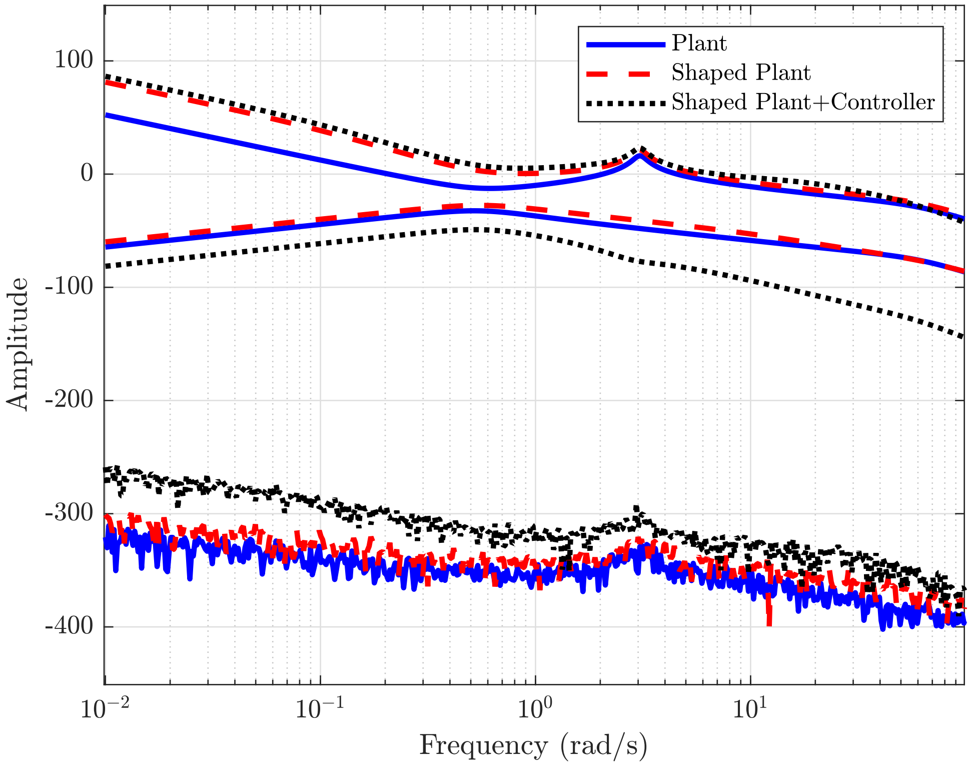
The Equation (5.16) shows that the taxi model is LPV and explicitly depends on the forward velocity of the UAV. The controller designed for fixed velocity point (e.g., 30 m/s) is used for the other velocities of the taxi envelope (e.g., $1,\ 5, ...45$ m/s). Figure 5.7c shows that the cross-over frequency changes, eventually affecting the system's performance specifications. Figure 5.7d shows that the rise time of the system varies from 6 to 300. Thus, a varying parameter controller (K_s) and observer (H_s) are required to obtain the desired performance requirements for the taxi envelope.

5.3.3 Design Steps for Parameter Varying Controller and Observers

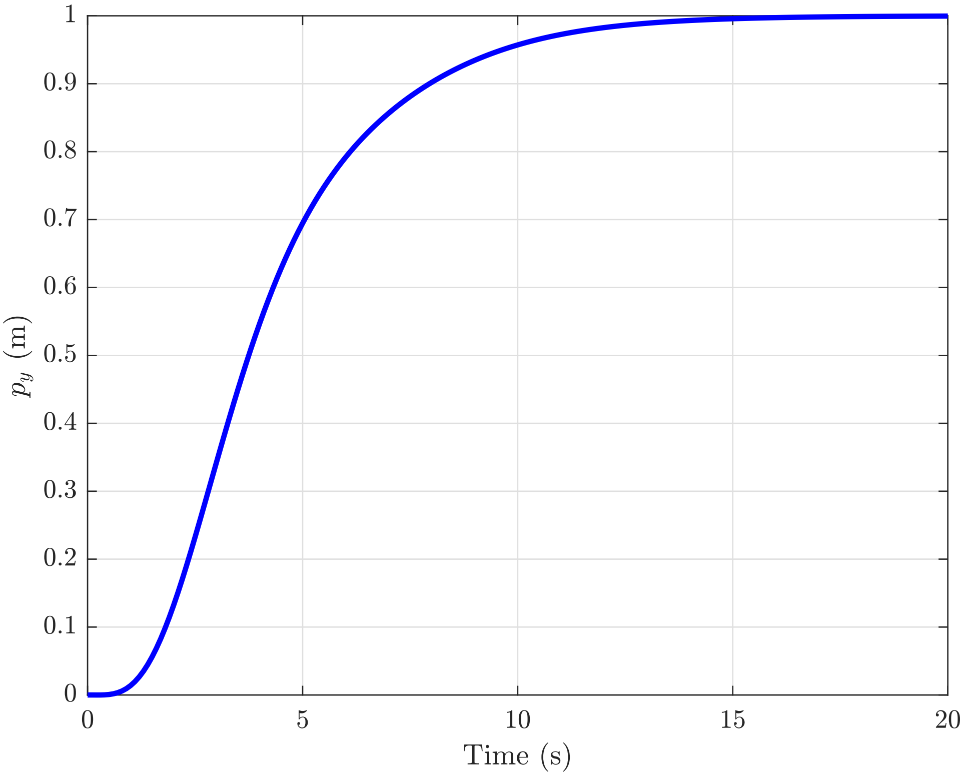
In this subsection, the parameter-dependent set of controllers $K_s(U)$ and observers $H_s(U)$ are designed. Let $\mathcal{U}$ be a set of finite number of points of $U \in [U_{\min}, U_{\max}]$ with an incremental step of 1 m/s. To achieve the desired performance throughout the taxi envelope, the controller is designed for every $U \in \mathcal{U}$. As a first step, the plant $G(U)$ is shaped for every $U \in \mathcal{U}$ with fixed pre-compensator W_1 and parameter-dependent post-compensator $W_2(U)$. In $W_2(U)$, only its first component $W_2^{(1,1)}(U)$ is scheduled as:

$$W_2^{(1,1)}(U) = k_{gs}(U) * W_2^{(1,1)}(U_0), \tag{5.21}$$

where $U_0 = 30$ m/s. This approach is general and is not limited to one parameter

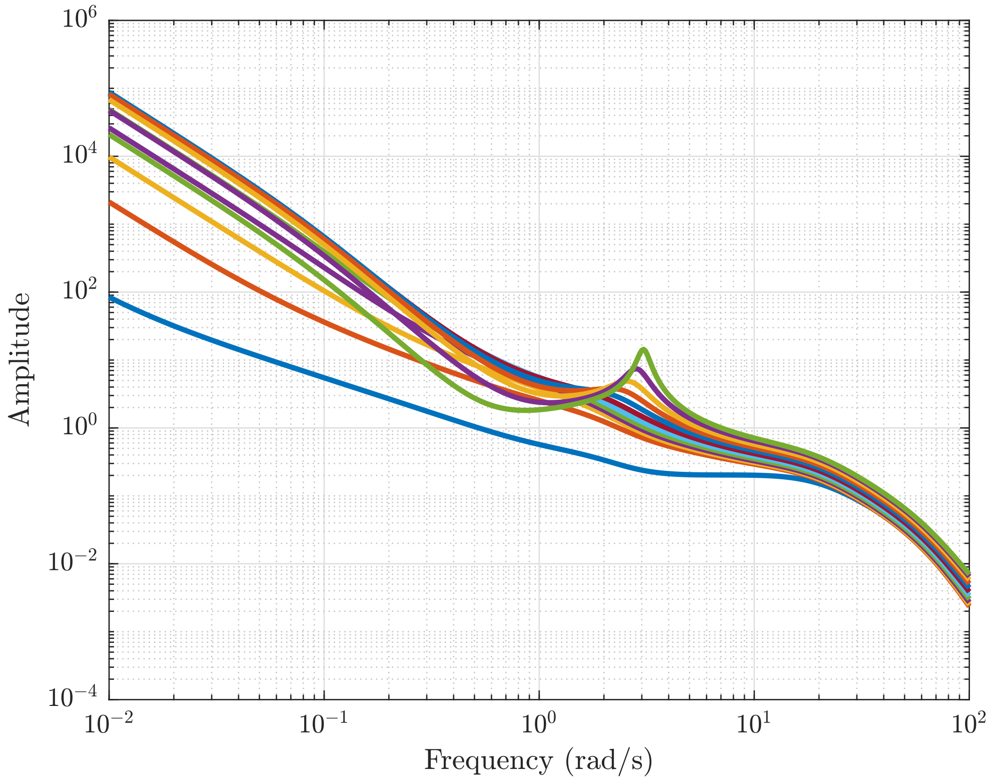


(A) Singular Values of Plant, Shaped Plant, and Shaped Plant+Controller

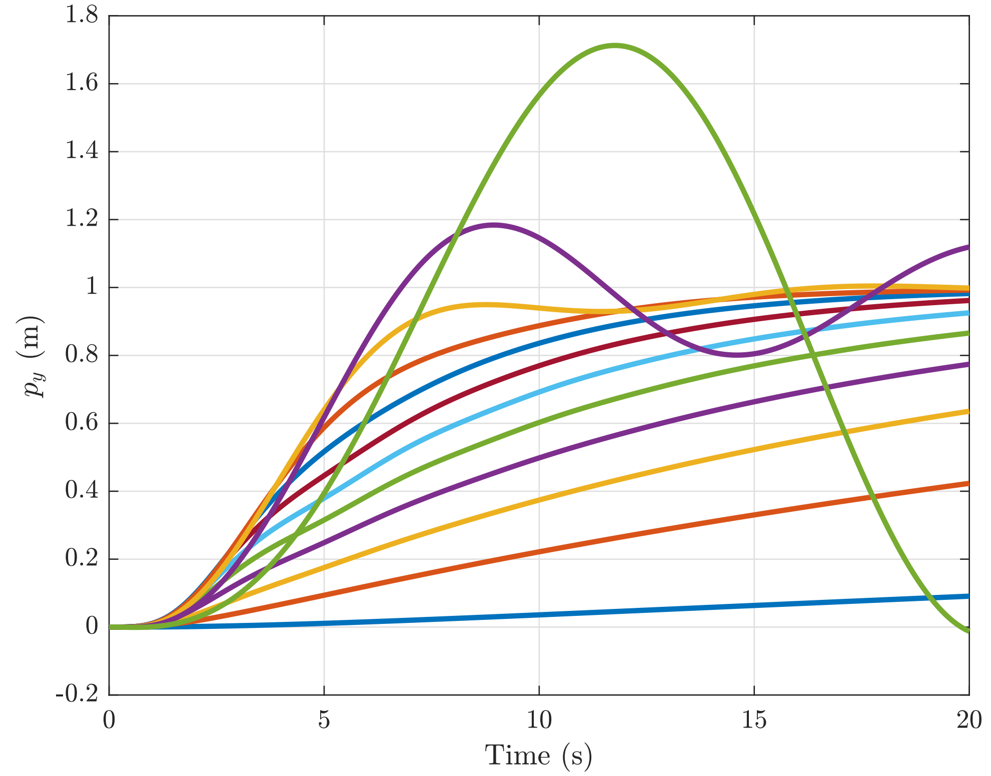


(B) Step Response in Cross Distance

Figure 5.7: Results for Fixed Velocity Point Design



(c) Maximum Singular Values on other Velocities



(d) Step Response on other Velocities

Figure 5.7: (Continued)

variation. One may change other coefficients of W_1 and W_2. The gain scheduling parameter $k_{gs}(U)$ at each velocity point is computed by solving H_∞ control problem iteratively to meet the desired rise time and other performance specifications. In Figure 5.8a, the maximum singular values of the loop transfer function are shown for every $U \in \mathcal{U}$. The cross-over frequency does not change much with the change in longitudinal velocity. The step response of the closed-loop system at different velocities is shown in Figure 5.8b. It shows that the system's performance requirement (rise time) is fairly good for the whole envelope of the taxi phase. The scheduled gains to achieve this performance are given in Figure 5.8c as a function of forward velocity, and it is modeled as:

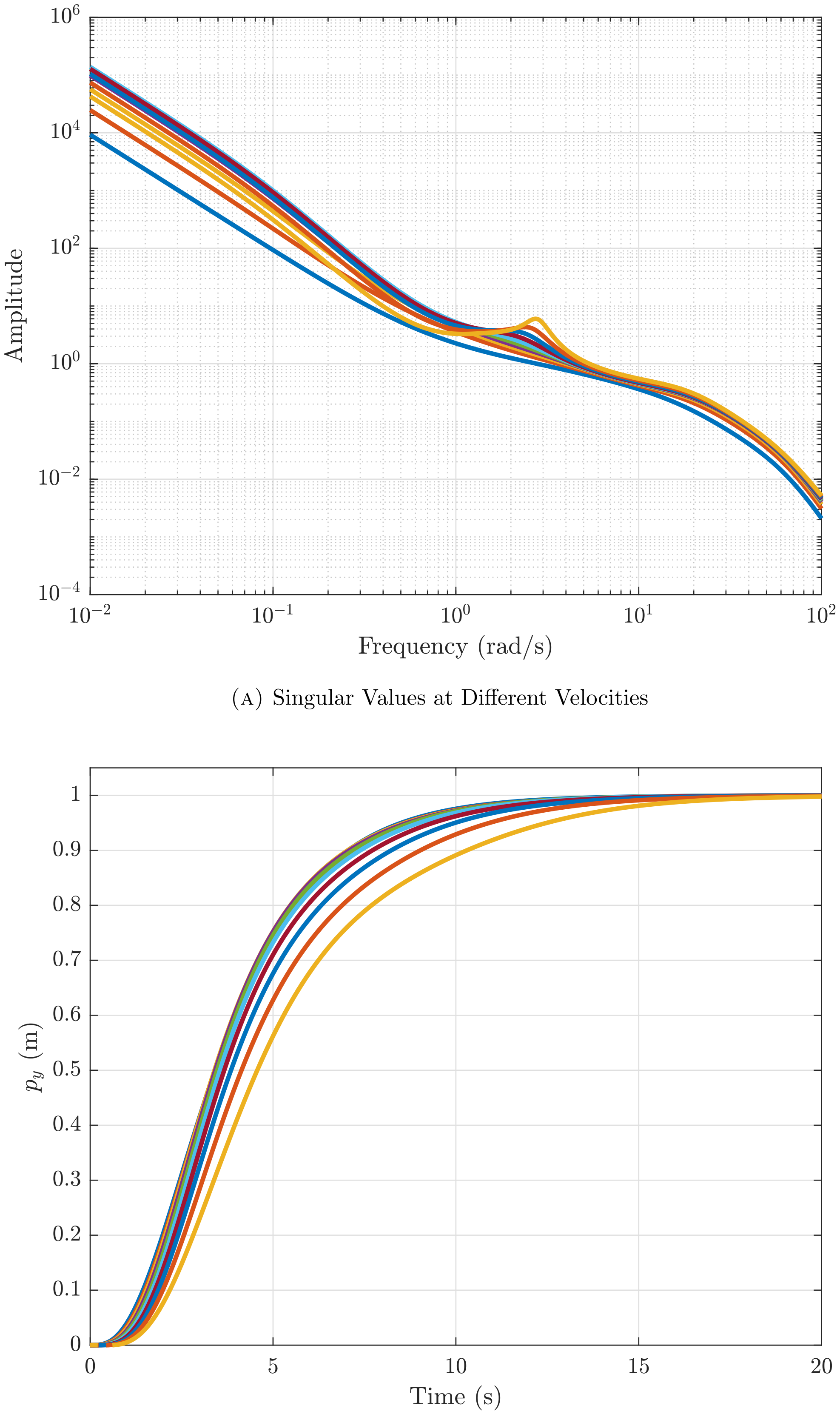
$$k_{gs}(U) = k_0 + k_1 U + k_2 U^2 + \frac{k_3}{U} \tag{5.22}$$

where $k_0,\ k_1,\ k_2$ and k_3 are constants. Figure 5.8d shows that the robustness of the system is also not degraded with the increase in rolling velocity and remains within the limit for the higher velocities as well.

The parameter varying weights are used to design the set of controllers (K_s) and observers (H_s) matrices for the taxi envelope. The knowledge of the physical system is used in order to define the structure of $K_s(U)$ and $H_s(U)$. Since shaped plant matrices $A_s(U),\ B_s(U)$ and $C_s(U)$ vary smoothly with operating point U and have explicit structure, then the controller $K_s(U)$ and observer $H_s(U)$ gains will vary smoothly if the Riccati solution $X_s(U)$ and $Z_s(U)$ vary smoothly ([67]). The following structure of observer and controller gain matrices is considered for the given problem.

$$\begin{aligned} H_s(U) &= H_0 + H_1 U + H_2 U^2 + \frac{H_3}{U} \\ K_s(U) &= K_0 + K_1 U + K_2 U^2 + \frac{K_3}{U}, \end{aligned} \tag{5.23}$$

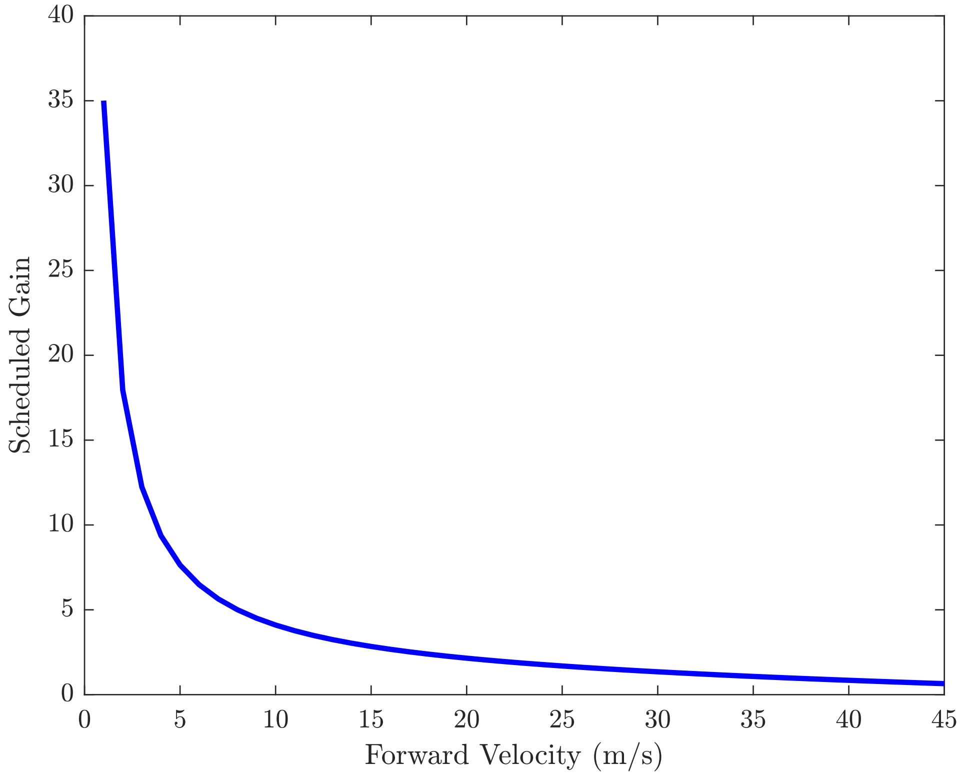
where K_i and H_i $(i \in 0, \cdots, 3)$ are constant matrices. The model in Equation (5.23) is a regression fit, and the coefficient matrices are obtained by minimizing the least square error.



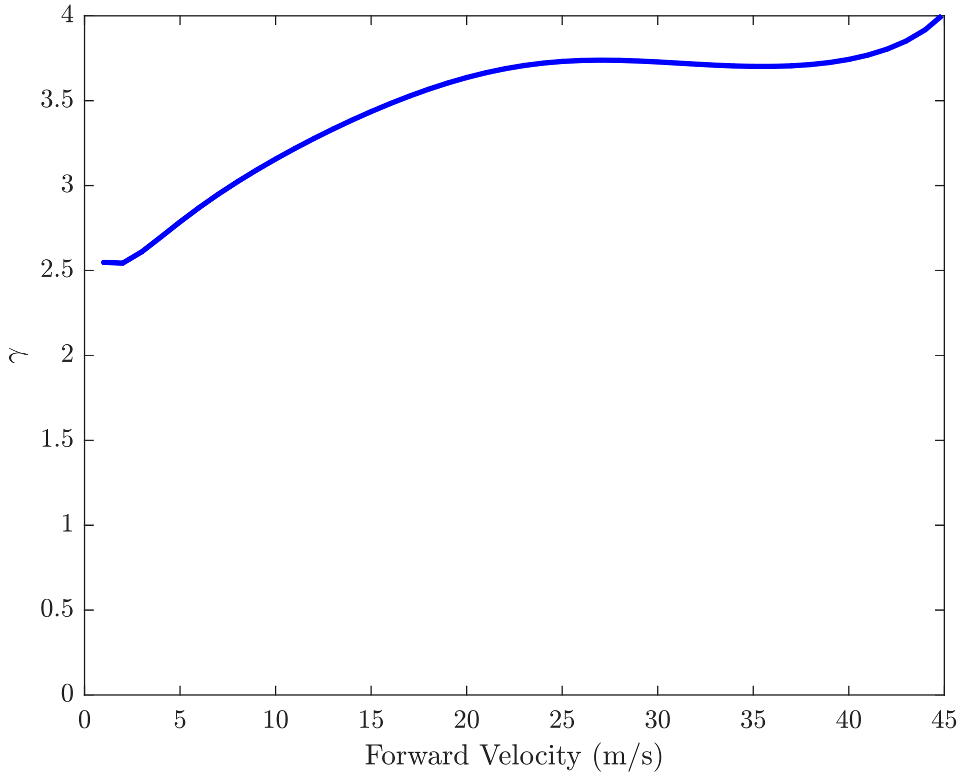
(A) Singular Values at Different Velocities

(B) Step Response at Different Velocities

FIGURE 5.8: Results for Fixed Velocity Point Design



(C) Scheduled gain



(D) γ at Different Velocities

FIGURE 5.8: (Continued)

5.3.4 Stability Analysis

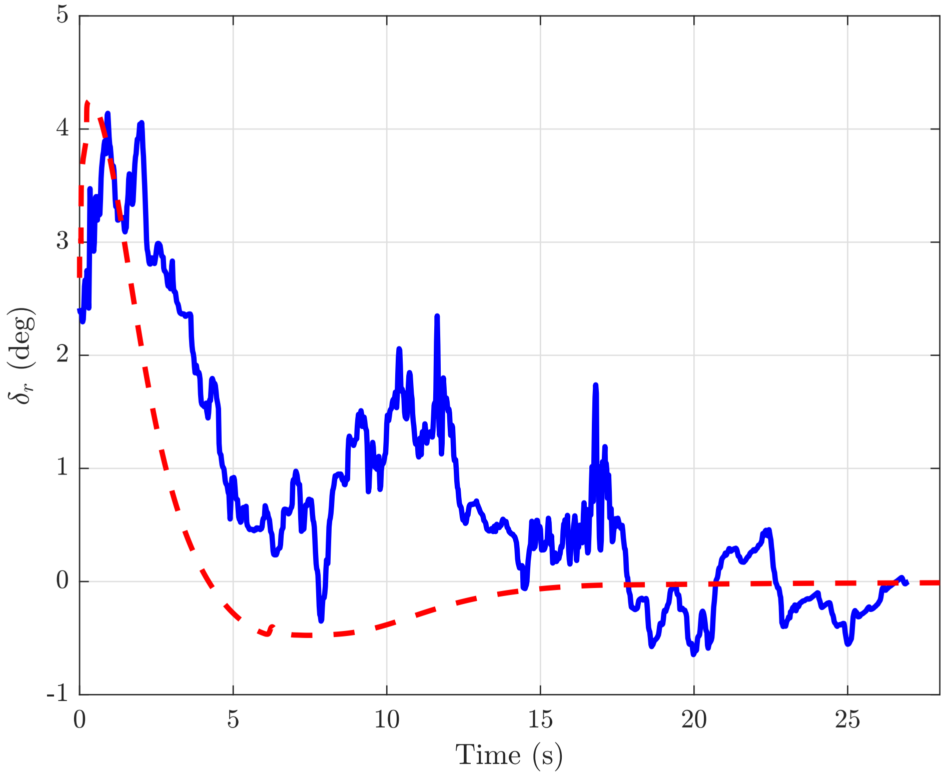
The stability of the system at designed points is assured by the $\gamma_{\min}$ but can not be guaranteed between the design points, e.g., infinite velocity points between 5 m/s to 6 m/s. In [80], the Lyapunov approach is presented to ensure the stability of LPV systems. The longitudinal velocity varies over a wide range (from start to take-off), so it may not be possible to have a single constant Lyapunov matrix P for the whole range. As argued in [81], it is also not necessary to have a globally valid single common Lyapunov matrix. Consequently, a piecewise Lyapunov function is sufficient to guarantee the stability of an LPV system. The whole taxi envelope is divided into 45 regions. The following LMI can state the quadratic stability of the LPV system for each region:

$$\begin{gathered} P_i = P_i' > 0, \quad \forall\, i = 1, 2, \ldots, 45 \\ \bar{A}_i(U)'P_i + P_i\bar{A}_i(U)' < 0, \quad U \in [U_i, U_{i+1}], . \end{gathered} \tag{5.24}$$

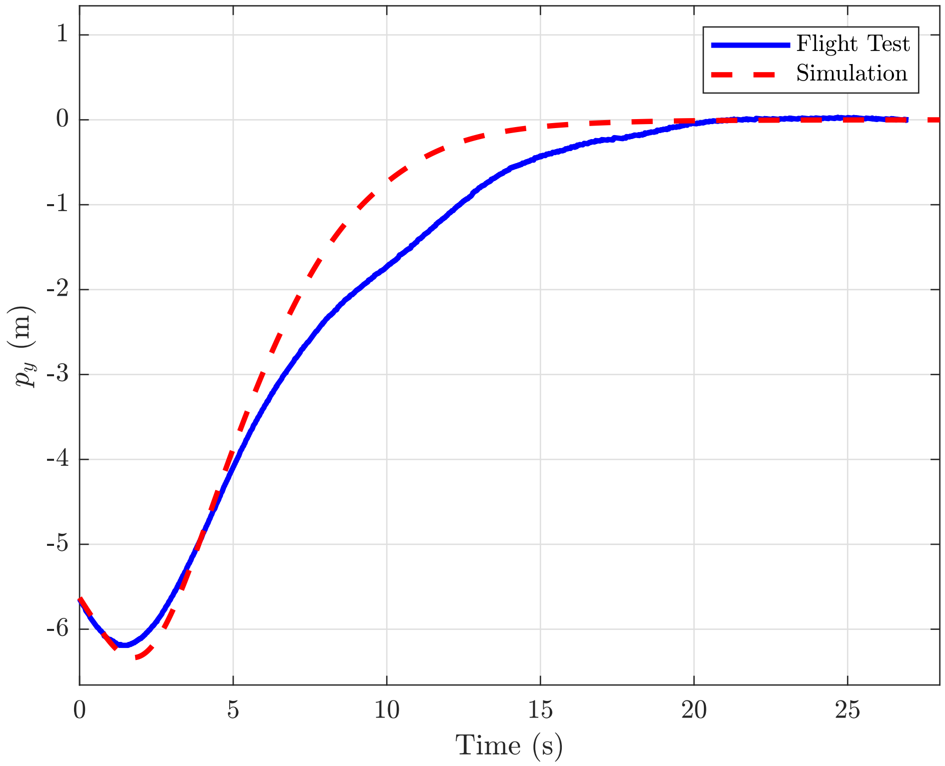
where $\bar{A}_i(U)$ is closed-loop matrix for i^{th} region. The algorithm given in [80] is used to compute a symmetric positive definite matrix P_i for each region, which satisfies the LMIs (5.24). It ensures a positive-definite matrix P_i for each interval which guarantees the stability between design points. This not only proves the stability at each U_i but also $U \in [U_i, U_{i+1}]$, $\forall\, i = 1, 2, \ldots, U_{45}$.

5.3.5 Field Test Results of LPV H_∞ Controller

To validate the LPV H_∞ controller, it is implemented on a test UAV, and experimental results are compared with the simulation. The test results are in good comparison with the simulation results as shown in Figure 5.9. In this case, the UAV touches the runway at a cross distance of 6 m, as shown in Figure 5.13b. The control actions δ_r, δ_n, and δ_b are generated by the controller to steer the UAV on the center line and are shown in Figures 5.13a, 5.13c and 5.13e respectively. The outputs (cross distance, yaw, and yaw rate) of the UAV are in Figures 5.13b, 5.13d and 5.13f respectively.

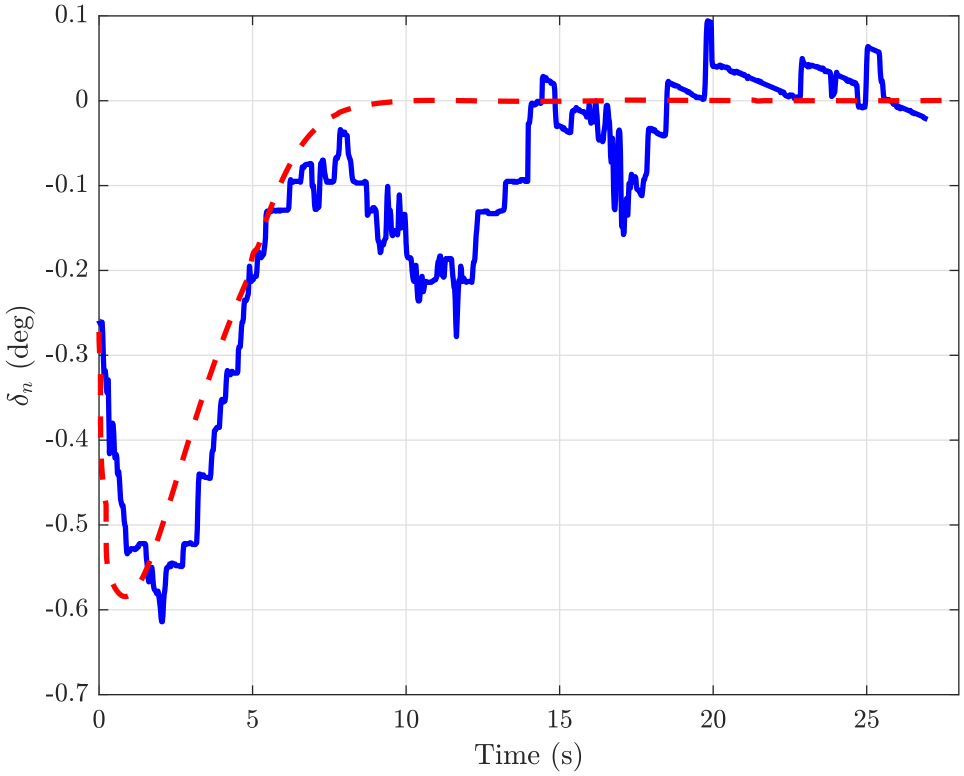


(A) Rudder Deflection

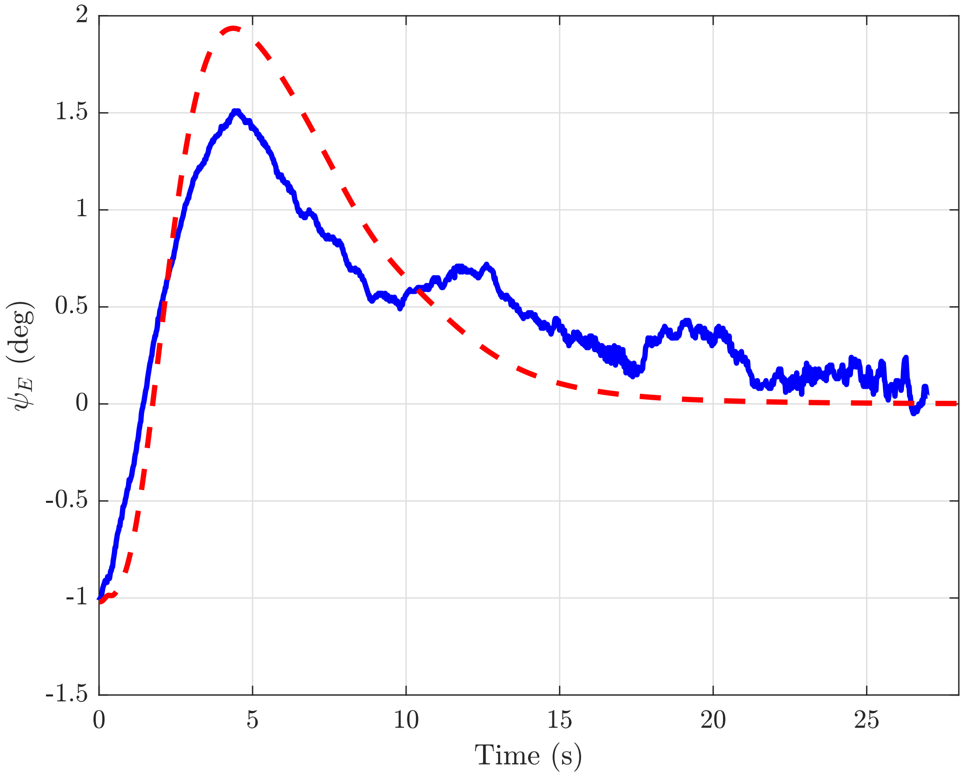


(B) Cross Distance of the UAV from Central Line

FIGURE 5.9: Test Results of LPV H_∞ for Lateral Taxi Control

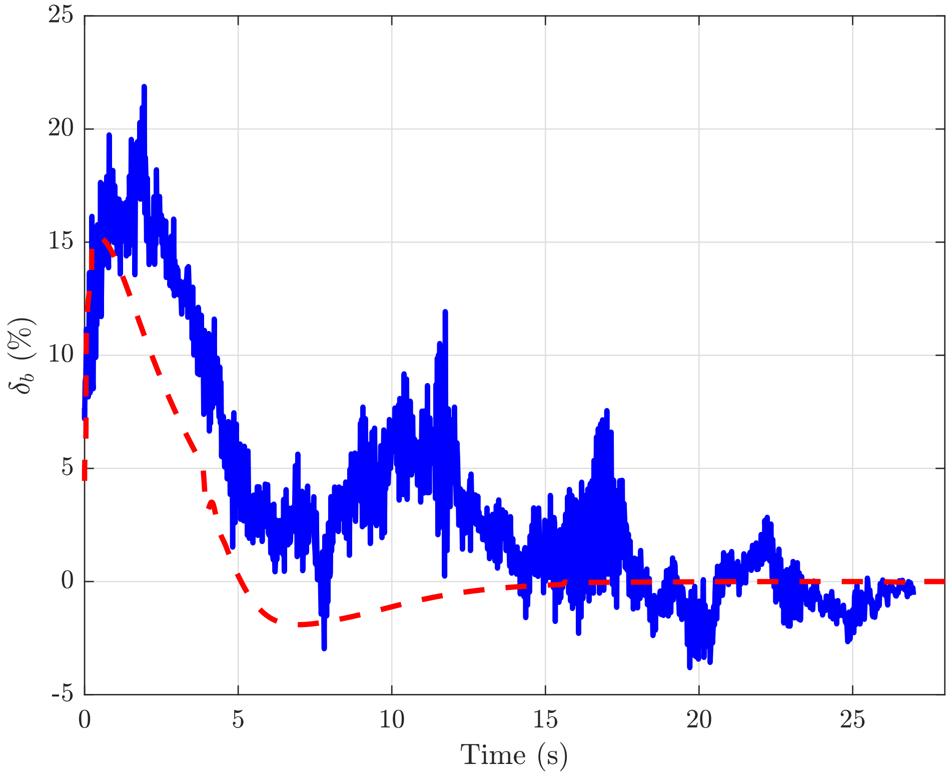


(c) Nose Wheel Deflection

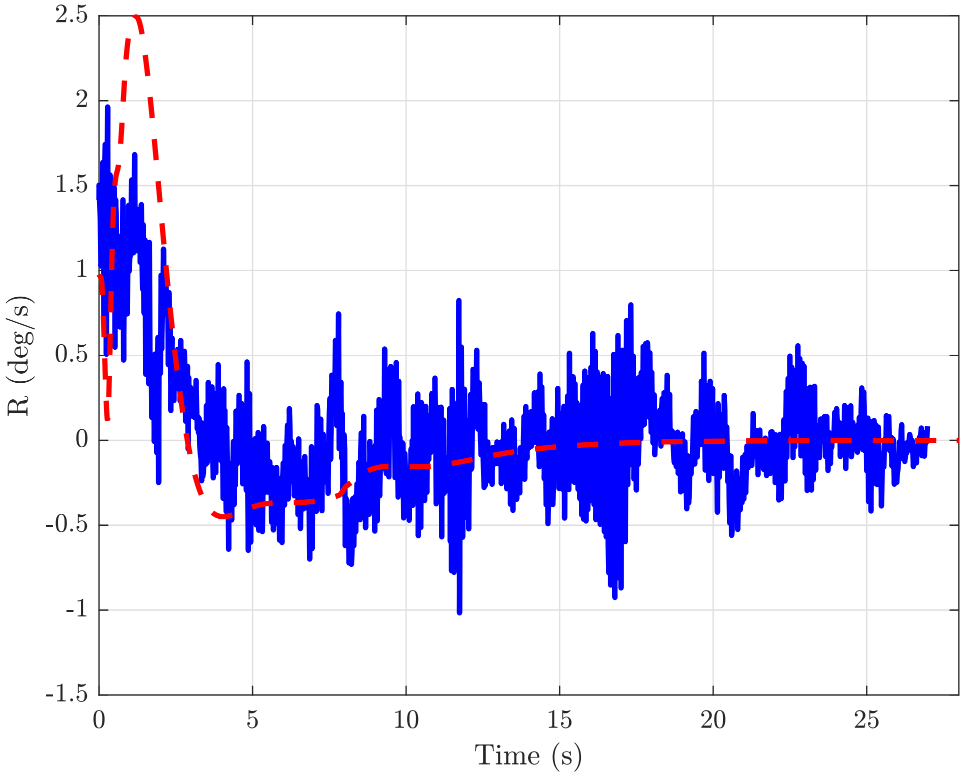


(d) Effective Yaw

Figure 5.9: (Continued)



(E) Differential Braking on Main Wheels



(F) Yaw Rate

FIGURE 5.9: (Continued)

To minimize the distance traveled by the UAV on the runway, a fix 50% brakes are also applied on the main wheels when its speed reaches 30 m/s. The longitudinal velocity of the UAV on the ground and the applied brakes to slow down the UAV are shown in Figure 5.10. After successfully designing and testing the LPV H_∞ controller, the next step is to design H_∞ based LPV-MPC, which is explained in the coming section.

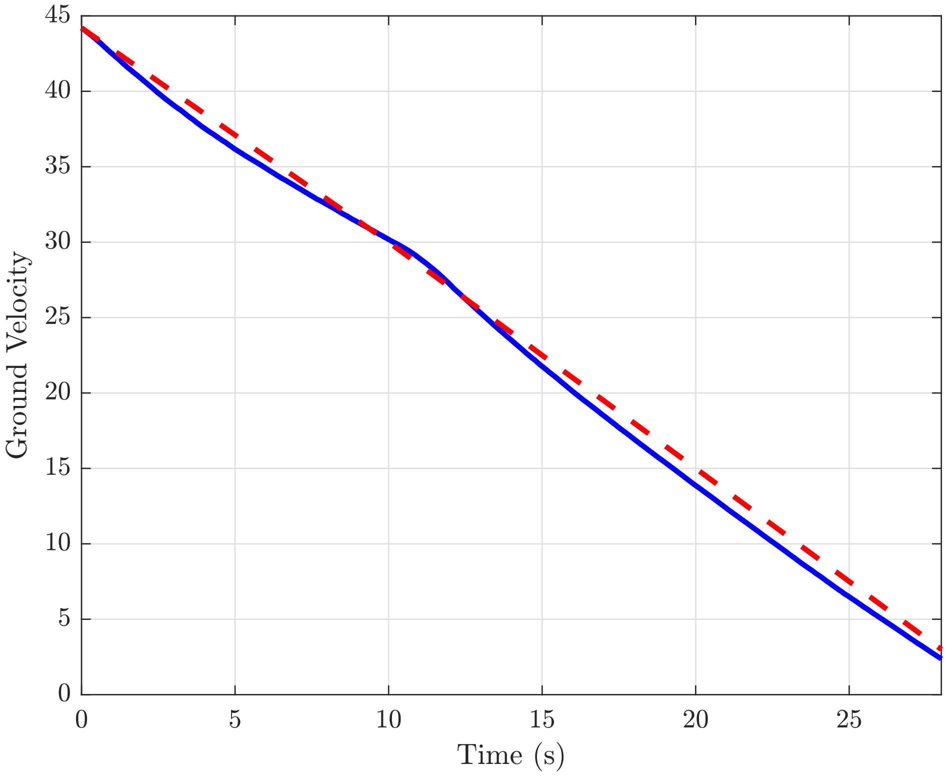
5.4 H_∞ Based MPC Design

The MPC is one of the most popular modern control techniques used frequently in the process industries. It is a model-based technique in which a cost function (usually quadratic) is minimized online along with a set of constraints over a prediction horizon at each time step. The major challenge in designing the MPC is the selection of cost function matrices which is explained in the next subsection.

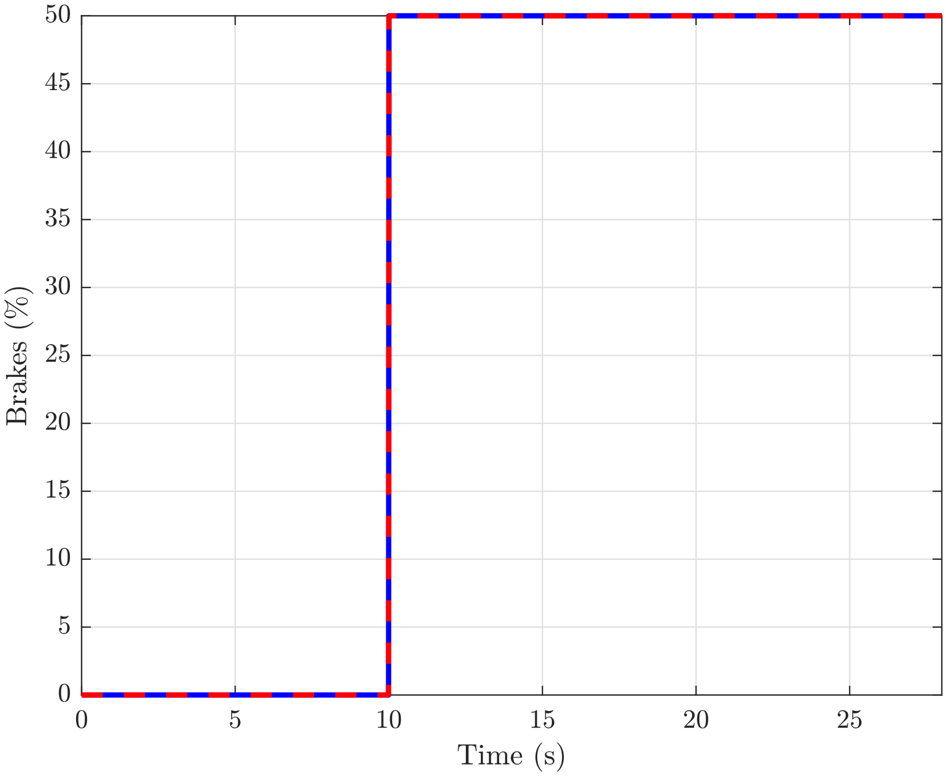
5.4.1 Design of Cost Function Matrices

A major challenge in the MPC design is the selection of cost-function weighting matrices. Generally, the tuning of MPC has based on heuristic rules and often does not provides the required performance or robust stability guarantee, even for the unconstrained problems. In this work, we used the method given in Section 3.3 to design the cost function matrices for the taxi problem. The cost function matrices P_s, Q_s, and R_s for the fixed velocity point ($U = 30$ m/s) are mentioned below. Then, the steps used in Section 5.3.3 are followed to design LPV cost function matrices as:

$$\begin{aligned} P_s(U) &= P_0 + P_1 U + P_2 U^2 + \frac{P_3}{U} \\ Q_s(U) &= Q_0 + Q_1 U + Q_2 U^2 + \frac{Q_3}{U} \\ R_s(U) &= R_0 + R_1 U + R_2 U^2 + \frac{R_3}{U}, \end{aligned} \tag{5.25}$$



(A) Ground Velocity of the UAV



(B) Brakes on Main Wheels

FIGURE 5.10: Ground Velocity and Brakes on Main Wheels

$$
P_s = \begin{bmatrix}
0.1313 & 0.0504 & 0.0230 & 0.0052 & -0.1240 & 0.0146 & -0.0318 & 0.2233 & 0.0000 & -0.0001 & -0.0001 & 0.0000 & 0.0001 & 0.0006 \\
0.0504 & 2.2648 & 0.1146 & -0.0020 & -0.9399 & 0.0003 & 0.0121 & 0.0286 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0230 & 0.1146 & 0.0236 & 0.0036 & -0.0651 & 0.0015 & -0.0104 & 0.1661 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0002 & 0.0002 \\
0.0052 & -0.0020 & 0.0036 & 0.0138 & -0.0037 & 0.0035 & -0.0969 & 0.0497 & 0.0001 & -0.0001 & 0.0001 & -0.0002 & 0.0005 & 0.0003 \\
-0.1240 & -0.9399 & -0.0651 & -0.0037 & 0.5281 & 0.0016 & 0.0804 & 0.6321 & 0.0000 & 0.0001 & 0.0002 & 0.0004 & 0.0004 & -0.0005 \\
0.0146 & 0.0003 & 0.0015 & 0.0035 & 0.0016 & 0.0353 & 0.0472 & 1.0367 & 0.0000 & 0.0000 & 0.0001 & 0.0005 & 0.0007 & 0.0012 \\
-0.0318 & 0.0121 & -0.0104 & -0.0969 & 0.0804 & 0.0472 & 1.9389 & 4.3362 & -0.0002 & 0.0031 & 0.0017 & 0.0087 & 0.0048 & -0.0083 \\
0.2233 & 0.0286 & 0.1661 & 0.0497 & 0.6321 & 1.0367 & 4.3362 & 56.5370 & 0.0007 & 0.0065 & 0.0093 & 0.0282 & 0.0391 & 0.0037 \\
0.0000 & 0.0000 & 0.0000 & 0.0001 & 0.0000 & 0.0000 & -0.0002 & 0.0007 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
-0.0001 & 0.0000 & 0.0000 & -0.0001 & 0.0001 & 0.0000 & 0.0031 & 0.0065 & -0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
-0.0001 & 0.0000 & 0.0000 & 0.0001 & 0.0002 & 0.0001 & 0.0017 & 0.0093 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & -0.0002 & 0.0004 & 0.0005 & 0.0087 & 0.0282 & 0.0000 & 0.0000 & 0.0000 & 0.0001 & 0.0000 & 0.0000 \\
0.0001 & 0.0000 & 0.0002 & 0.0005 & 0.0004 & 0.0007 & 0.0048 & 0.0391 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0001 & 0.0000 \\
0.0006 & 0.0000 & 0.0002 & 0.0003 & -0.0005 & 0.0012 & -0.0083 & 0.0037 & -0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0007
\end{bmatrix}
$$

$$Q_s = \begin{bmatrix}
0.0076 & -0.0067 & 0.0047 & 0.0033 & -0.0041 & 0.0003 & -0.0260 & -0.0125 & 0.0000 & 0.0000 & 0.0000 & -0.0001 & 0.0001 & 0.0000 \\
-0.0067 & 0.7333 & 0.0353 & -0.0017 & -0.2873 & -0.0009 & 0.0127 & 0.0050 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0001 \\
0.0047 & 0.0353 & 0.0078 & 0.0025 & -0.0181 & -0.0012 & -0.0195 & -0.0147 & 0.0000 & 0.0000 & 0.0000 & -0.0001 & 0.0001 & 0.0001 \\
0.0033 & -0.0017 & 0.0025 & 0.0120 & -0.0022 & 0.0022 & -0.0905 & 0.0279 & 0.0001 & -0.0001 & 0.0001 & -0.0003 & 0.0005 & 0.0002 \\
-0.0041 & -0.2873 & -0.0181 & -0.0022 & 0.1190 & 0.0002 & 0.0200 & 0.0222 & 0.0000 & 0.0000 & 0.0000 & 0.0001 & 0.0000 & 0.0001 \\
0.0003 & -0.0009 & -0.0012 & 0.0022 & 0.0002 & 0.0025 & -0.0087 & 0.0397 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0001 & 0.0002 \\
-0.0260 & 0.0127 & -0.0195 & -0.0905 & 0.0200 & -0.0087 & 0.7676 & 0.0104 & -0.0007 & 0.0010 & -0.0012 & 0.0025 & -0.0036 & -0.0001 \\
-0.0125 & 0.0050 & -0.0147 & 0.0279 & 0.0222 & 0.0397 & 0.0104 & 1.3350 & -0.0013 & 0.0001 & -0.0002 & 0.0008 & 0.0023 & 0.0059 \\
0.0000 & 0.0000 & 0.0000 & 0.0001 & 0.0000 & 0.0000 & -0.0007 & -0.0013 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & -0.0001 & 0.0000 & 0.0000 & 0.0010 & 0.0001 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0001 & 0.0000 & 0.0000 & -0.0012 & -0.0002 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
-0.0001 & 0.0000 & -0.0001 & -0.0003 & 0.0001 & 0.0000 & 0.0025 & 0.0008 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0001 & 0.0000 & 0.0001 & 0.0005 & 0.0000 & 0.0001 & -0.0036 & 0.0023 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & -0.0001 & 0.0001 & 0.0002 & 0.0001 & 0.0002 & -0.0001 & 0.0059 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000
\end{bmatrix}$$

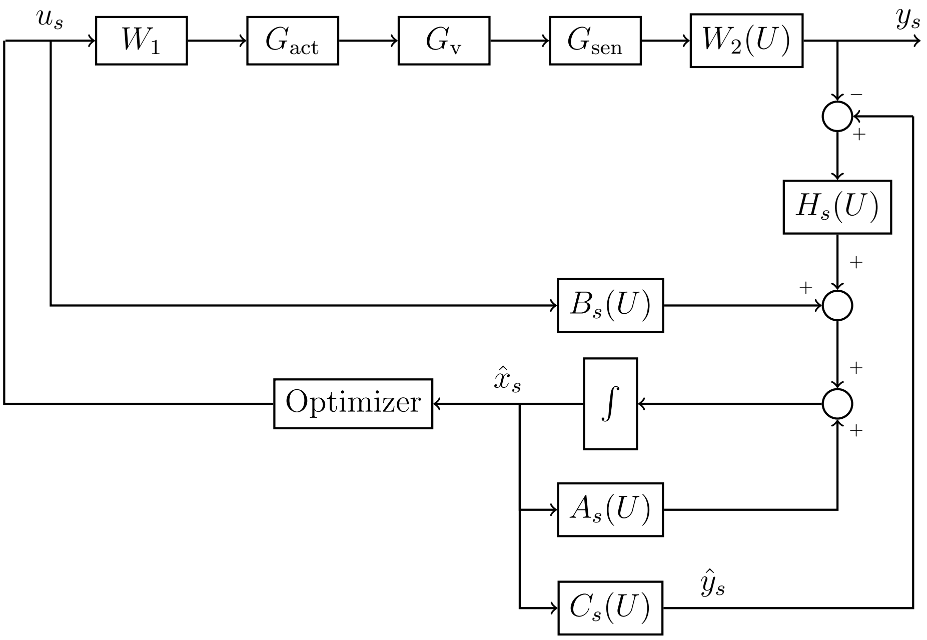


FIGURE 5.11: An Observer-based Implementation of LPV-MPC

Figure 5.11 shows the block diagram of the H_∞ based LPV-MPC implementation in observer form.

5.4.2 LPV-MPC Design

The MPC is designed for the shaped LPV model having input constraints ±10°, ± 3° and ±20% for rudder deflection, nose wheel deflection, and percentage braking, respectively, along with the LPV observer and cost function matrices from (5.23) and (5.25) respectively.

The MPC formulation in Section 3.4 can be posed as a quadratic programming (QP) problem. Generally, the dense approach given in [73] is used for LTI systems. The computationally expensive task, matrix multiplication (e.g., A^N), is done offline, and the QP problem is solved online at each sample step. For LPV systems (i.e., taxi problem) where the system matrix A changes at each time step, one cannot use the dense approach because it also requires matrix multiplication online. Another approach, sparse formulation, can be preferred for the LPV system and is given below.

$$V_N(\theta) = \frac{1}{2}\theta^T H(U)\theta,$$

$$\text{s.t.} \qquad F(U)\theta = f(\hat{x})$$

$$G\theta \leq g$$

where,

$$\theta = [\ x_0^T \quad u_0^T \quad x_1^T \quad u_1^T \ \cdots \ u_{N-1}^T \ \ x_N^T]^T$$

$$F(U) = \begin{bmatrix} -I & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ A(U) & B(U) & -I & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & A(U) & B(U) & -I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & A(U) & B(U) & -I \end{bmatrix}, \quad f = \begin{bmatrix} \text{-}\hat{x} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$G = \begin{bmatrix} J & E & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & J & E & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & J & E & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & J \end{bmatrix}, \quad g = \begin{bmatrix} d \\ d \\ d \\ \vdots \\ d \end{bmatrix}$$

$$H(U) = \begin{bmatrix} Q(U) & 0 & 0 & \cdots & 0 & 0 \\ 0 & R(U) & 0 & \cdots & 0 & 0 \\ 0 & 0 & Q(U) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & R(U) & 0 \\ 0 & 0 & 0 & \cdots & 0 & P(U) \end{bmatrix}$$

In this approach, one can avoid online matrix multiplication, but the number of unknowns in the QP problem has increased from mN to $(n+m)N+n$, where n and m are states and inputs of the system, respectively. The increased number of decision variables greatly impacts computational and memory requirements.

This issue has been dealt with in [82] with the efficient interior point method by exploiting the structure of the sparse approach and discussed in coming subsection.

5.4.3 Efficient Interior Point Method

The Interior-Point Method (IPM) is an optimization strategy to solve convex problems. It starts from the feasible region and searches for the best solution. The Primal-Dual Interior-Point Method (PDIPM) employing Mehrotra's predictor-corrector scheme is used in this work [83]. At each sampling step, the MPC cost function matrices are computed by (5.25) for corresponding longitudinal velocity U and problem 5.26 is optimized with the help of IPM. The performance bottleneck and most expensive step in IPM is the computation of the search direction in the Karush–Kuhn–Tucker (KKT) conditions [84]. To formulate the KKT conditions, a slack variable s is added to the inequality constraint, and Lagrangian is formed as follows:

$$\begin{aligned} L(\theta, v, \lambda, s) = \frac{1}{2}\theta^T H\theta + h^T\theta + v^T(F\theta - f) \\ + \lambda^T(G\theta - g + s) \end{aligned} \tag{5.26}$$

The vector $v \in \mathbf{R}^{n_e}$ and $\lambda \in \mathbf{R}^{ni}$ are called Lagrange multipliers vectors or dual variables, and n_e and n_i are the number of equality and inequality constraints. The KKT conditions are as follows:

$$\begin{aligned} H\theta + F^T v + G^T\lambda + h &= 0 \\ -F\theta + f &= 0 \\ -G\theta + g - s &= 0 \\ \Lambda S e &= 0 \\ \lambda,\ s &\geq 0 \end{aligned}$$

where $s \in \mathbf{R}^{n_i}$ is the vector of slack variables, $e \in \mathbf{R}^{n_i}$ is a vector of ones and $\Lambda,\ S \in \mathbf{R}^{n_i \times n_i}$ are diagonal matrices define as:

$$\Lambda = \text{diag}(\lambda_1,\ \lambda_2,\ \cdots,\ \lambda_m), \quad S = \text{diag}(s_1,\ s_2,\ \cdots,\ s_m)$$

The equation $\Lambda S e = 0$ is called the complementarity condition. The PDIPM generate the iterates $(\theta^i,\ v^i,\ \lambda^i,\ s^i),\ i = 1, 2, \cdots$ that approach feasibility with respect to the KKT conditions. The KKT conditions given in (5.27) can be written as a linear system to be solved for the search direction as:

$$\begin{bmatrix} H & F^T & G^T & 0 \\ -F & 0 & 0 & 0 \\ -G & 0 & 0 & -I \\ 0 & 0 & S & \Lambda \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta v \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_H \\ r_F \\ r_G \\ \Lambda S e \end{bmatrix} \tag{5.27}$$

where,

$$\begin{aligned} r_H &= H\theta + F^T v + G^T \lambda + h \\ r_F &= F\theta - f \\ r_G &= G\theta - g + s \\ r_s &= -\Lambda S e \end{aligned} \tag{5.28}$$

The block elimination method can be applied on (5.27) to obtain the reduced system. After the elimination of ΔS and $\Delta\lambda$ as:

$$\begin{aligned} \Delta S &= \Lambda^{-1}(r_s - S\Delta\lambda) \\ \Delta\lambda &= S^{-1}(\Lambda r_G + r_S + \Lambda G \Delta\theta) \end{aligned} \tag{5.29}$$

The so-called augmented system is obtained as follows:

$$\begin{bmatrix} \Phi & F^T \\ -F & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta v \end{bmatrix} = \begin{bmatrix} r_T \\ r_F \end{bmatrix} \tag{5.30}$$

where,

$$\begin{aligned} \Phi &= H + G^T \Lambda S^{-1} G \\ r_T &= -r_H - G^T S^{-1}(\Lambda r_G + r_S) \end{aligned}$$

Solution of (5.30) gives the primal $\Delta\theta$ and dual Δv search step, and by substitution in 5.29, one can find complete search direction.

Solving (5.30) without exploiting the structure of sparse formulation, the computation cost is $\frac{1}{3}N^3(2n+m)^3$ flops [85]. The methods presented below can improve the speed of solving the problem. The search direction at each step is updated by (5.31) that is the Schur complement of the (5.30) [86]:

$$Y\Delta v = \beta, \tag{5.31}$$

where

$$\begin{aligned} Y &= F\Phi^{-1}F^T \\ \beta &= r_F + F\Phi^{-1}r_T \end{aligned}$$

and

$$\Delta\theta = \phi^{-1}\left(r_T - F^T\Delta v\right) \tag{5.32}$$

The resulting matrix Y yields a block-banded structure as shown in (5.34) and can be efficiently factored by block-wise Cholesky factorization. The matrix Φ also has a block diagonal structure just like the H matrix, and one can take Φ^{-1} block-wise to reduce the computations, and its structure is shown in (5.33).

$$\Phi^{-1} = \begin{bmatrix} \bar{Q}_0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \bar{R}_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \bar{Q}_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{R}_1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \bar{Q}_{N-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \bar{R}_{N-1} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \bar{P}_N \end{bmatrix} \tag{5.33}$$

In (5.31), instead of direct multiplying ($Y = F\Phi^{-1}F^T$), the elements of the Y

matrix can be computed more effectively by exploiting the structure of Φ^{-1} and F. Each element of Y is $n \times n$, and corresponding elements of tridiagonal are symmetric, meaning $Y_{ij} = Y_{ji}^T$. The Algorithm 1 is used to find matrix elements Y. To solve (5.31) for Δv, we need to take the inverse of matrix Y. Rather than inverting Y directly, it is also possible to solve (5.31) by Cholesky decomposition into a lower triangular matrix and its transpose, $Y = LL^T$. Then, the forward and backward substitution is used to find the search direction Δv. The Cholesky decomposition can be time-consuming and computationally expensive for the dense matrix, but we can take advantage of the known tridiagonal structure of Y in a sparse approach. The Algorithm 2 is used to calculate the elements of L and is given in (5.35).

Algorithm 1 An Algorithm for Calculating the Elements of Y

```
function Elements of Y
    $Y_{11} = \bar{Q}_0$
    for $i = 2, \cdots, N$ do
        $Y_{i-1,i} = -\bar{Q}_{i-2}A^T$
        $Y_{i,i} = A\bar{Q}_{i-2}A^T + B\bar{R}_{i-2}B^T + \bar{Q}_{i-1}$
    end for
end function
```

$$Y = \begin{bmatrix} Y_{11} & Y_{12} & 0 & \cdots & 0 & 0 \\ Y_{21} & Y_{22} & Y_{23} & \cdots & 0 & 0 \\ 0 & Y_{32} & Y_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & Y_{N-1,\ N-1} & Y_{N-1,\ N} \\ 0 & 0 & 0 & \cdots & Y_{N,\ N-1} & Y_{N,\ N} \end{bmatrix} \tag{5.34}$$

Algorithm 2 An Algorithm for Cholesky Decomposition of Matrix Y

```
function Cholesky Decomposition of Y
    $L_1 = chol(Y_{11})$
    for $i = 1, \cdots, N - 1$ do
        $M_i = Y_{i,i+1}\ inv(L_i)$
        $L_{i+1} = chol(Y_{i+1,i+1} - M_i M_i^T)$
    end for
end function
```

$$L = \begin{bmatrix} L_{11} & 0 & 0 & \cdots & 0 & 0 \\ M_{21} & L_{22} & 0 & \cdots & 0 & 0 \\ 0 & M_{32} & L_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & L_{N-1,\ N-1} & 0 \\ 0 & 0 & 0 & \cdots & M_{N,\ N-1} & L_{N,\ N} \end{bmatrix} \tag{5.35}$$

By this method, one can avoid many flops while calculating the search direction. The Algorithms 1 and 2 required an order of $N(n+m)^3$ and Nn^3 flops, respectively.

5.4.4 Results

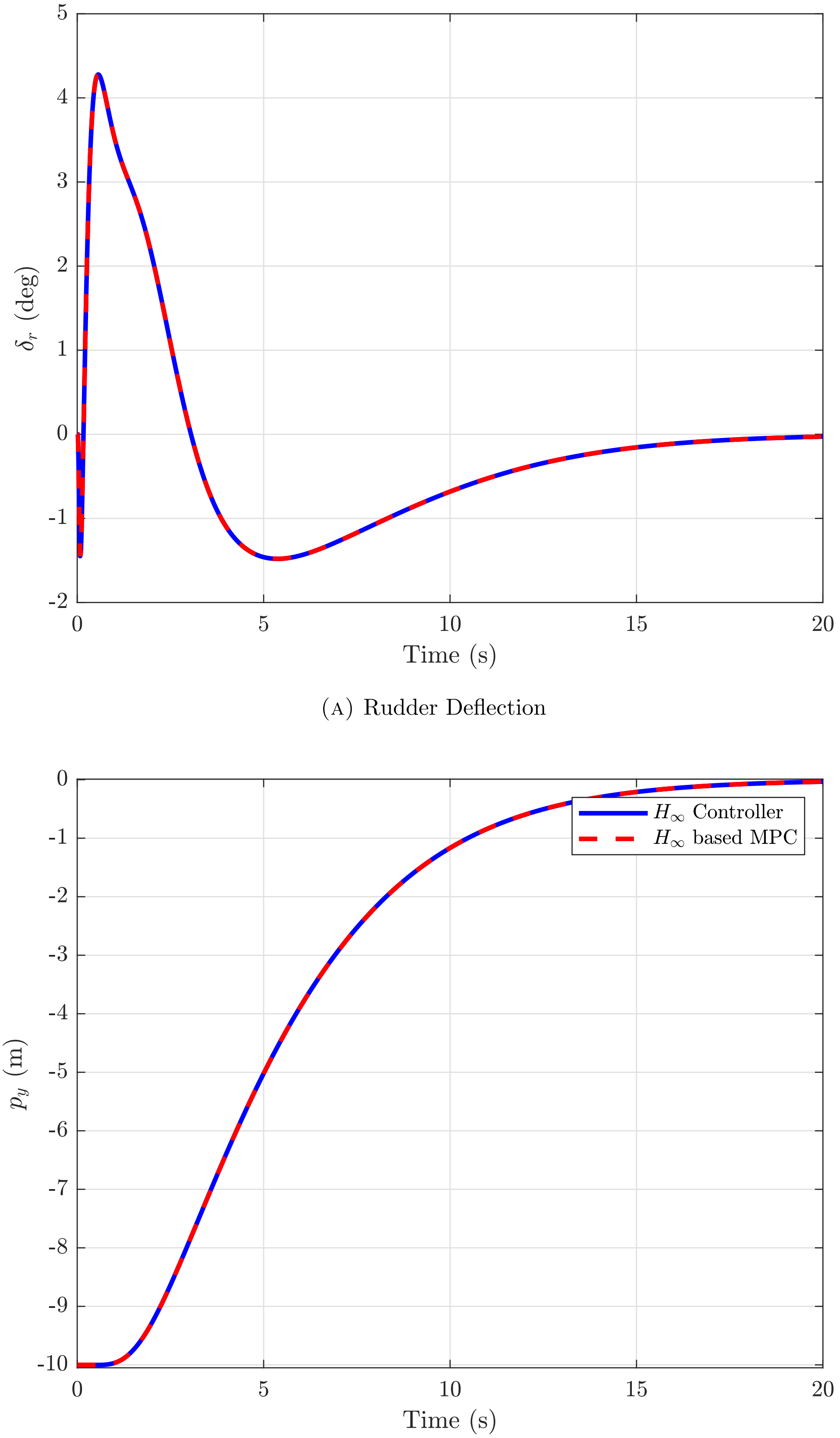
The proposed control law is implemented in a non-linear simulation environment to validate the designed algorithm. The H_∞ based LPV-MPC strategy presented in section 3.4 with $N = 10$ and cost function matrices $P_s(U)$, $Q_s(U)$ and $R_s(U)$ designed by (5.25) is implemented. The simulations were conducted for two different scenarios. In the first case, the design is tested when the constraints are inactive, and the second illustrates the constraints management of the control technique.

5.4.4.1 Case-1

In the first case, it is shown that $u_{\text{MPC}} = u_{\text{H}_\infty}$, whenever, the constraints are inactive. The input saturation limits are $\pm 10°$, $\pm 3°$ and $\pm 20\%$ for rudder deflection, nose wheel deflection, and percentage of differential braking. A cross distance from the centerline of the runway is taken as 10m. Figure 5.12 shows that the input demand does not exceed the saturation limits, and the NRMSE between the outputs shows that both controllers are in high fidelity.

5.4.4.2 Case-2

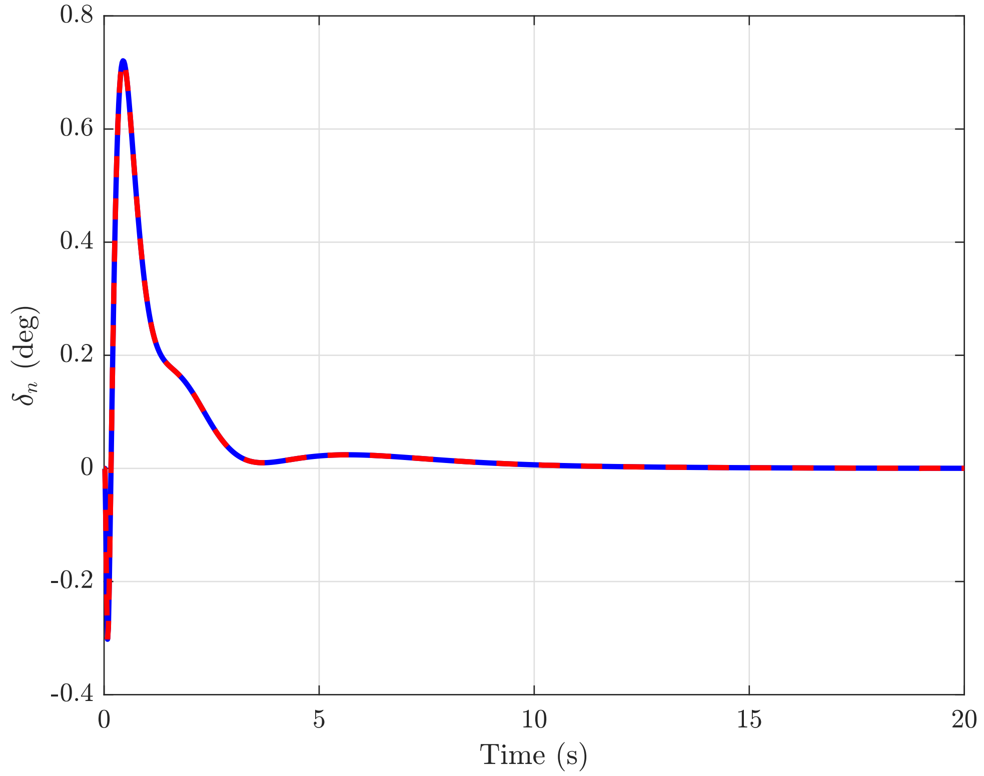
The second case has been formulated by reducing the input saturation limits to $\pm 3°$, $\pm 0.5°$ and $\pm 5\%$ for rudder deflection, nose wheel deflection, and percentage



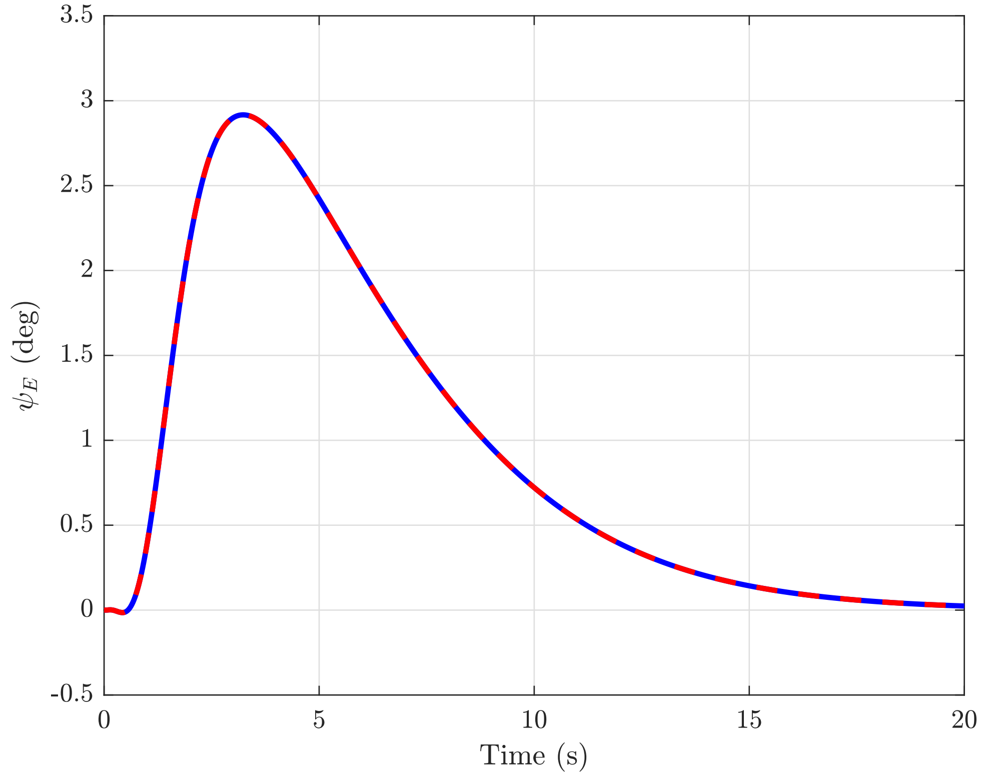
(A) Rudder Deflection

(B) Cross Distance of the UAV from Central Line

FIGURE 5.12: Comparison of H_∞ based LPV-MPC and LPV H_∞ when Constraints are Inactive

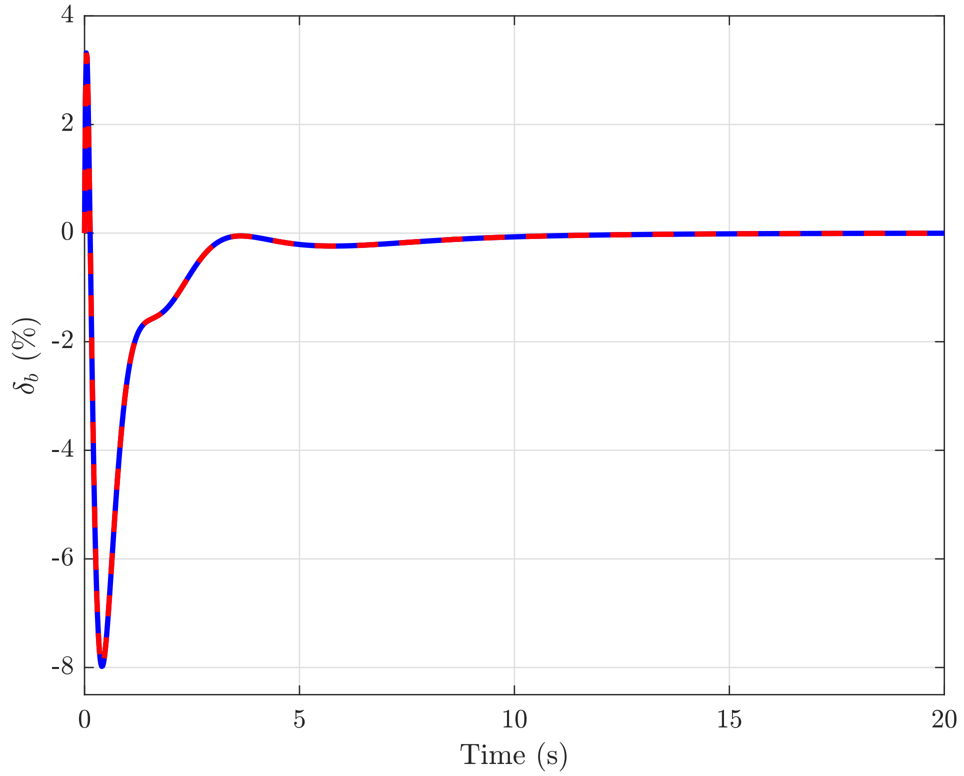


(c) Nose Wheel Deflection

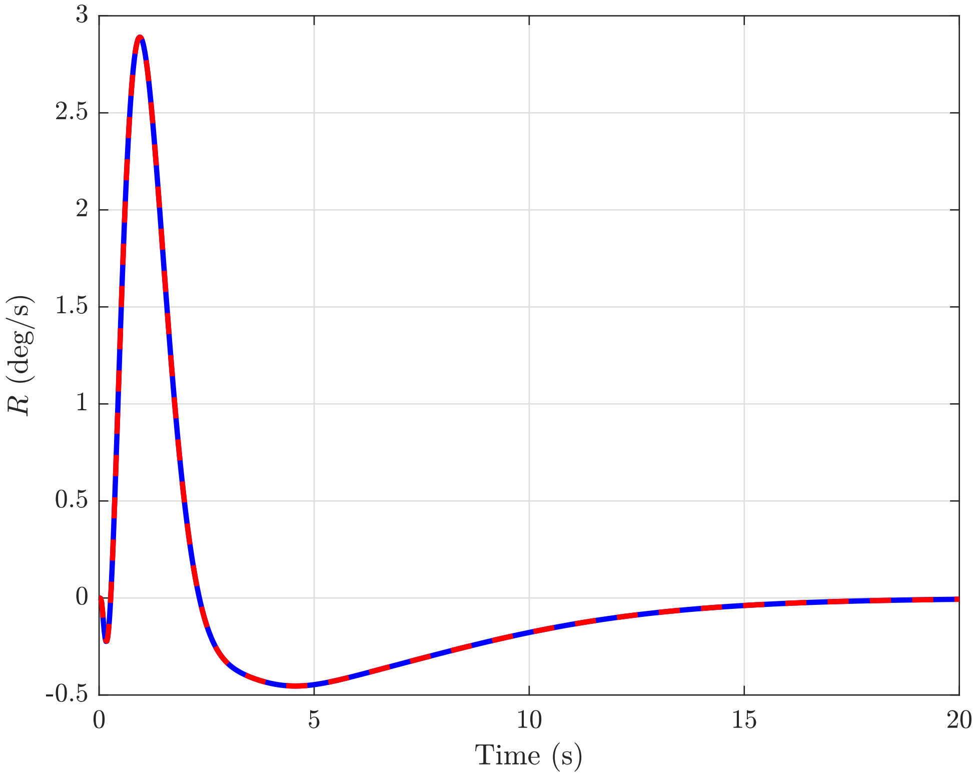


(d) Effective Yaw

Figure 5.12: (Continued)

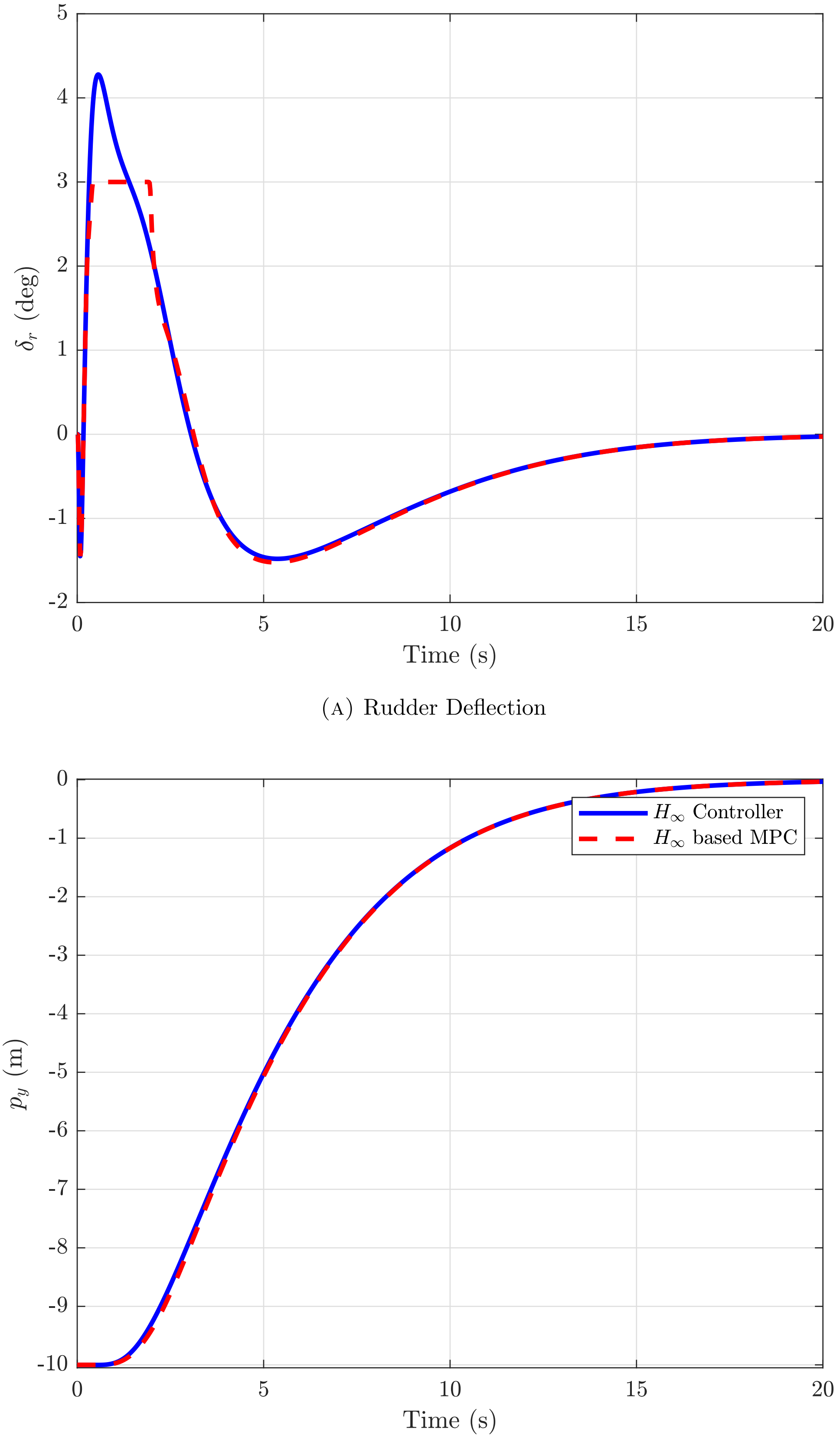


(E) Differential Braking on Main Wheels



(F) Yaw Rate

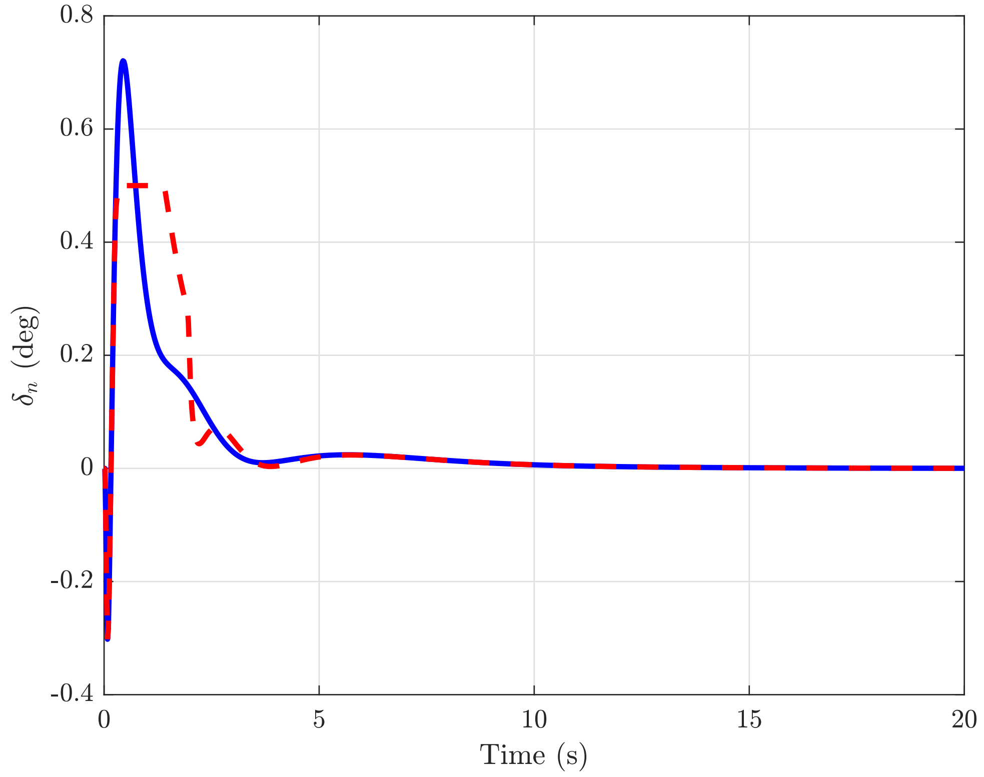
FIGURE 5.12: (Continued)



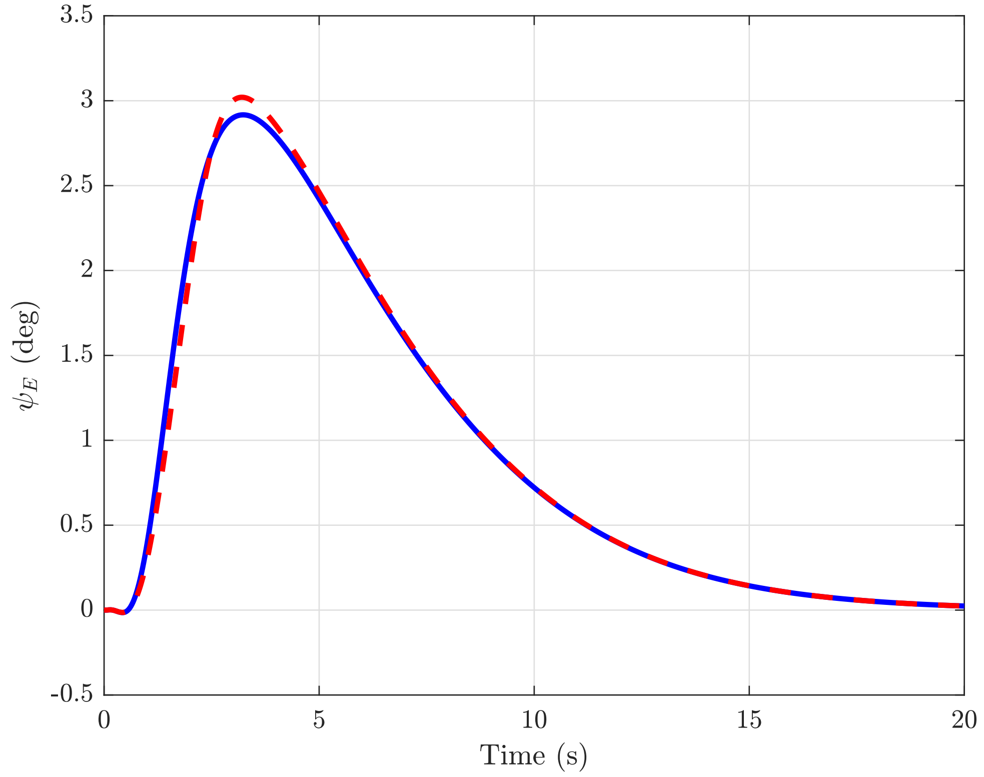
(A) Rudder Deflection

(B) Cross Distance of the UAV from Central Line

FIGURE 5.13: Comparison of H_∞ based LPV-MPC and LPV H_∞ when Constraints are Active

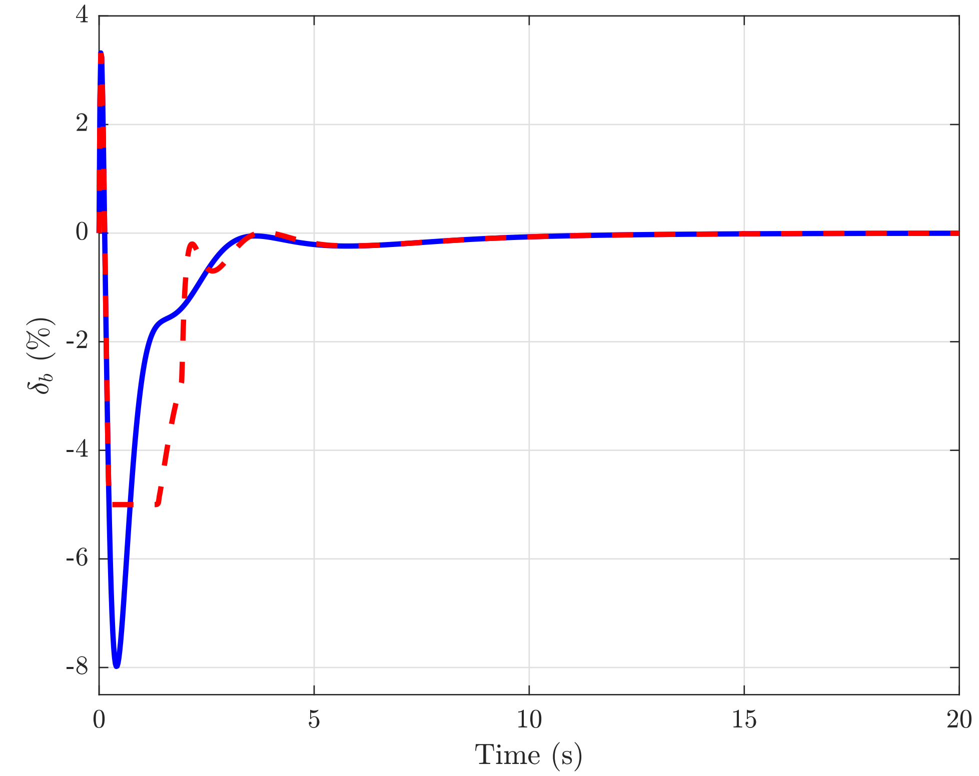


(c) Nose Wheel Deflection

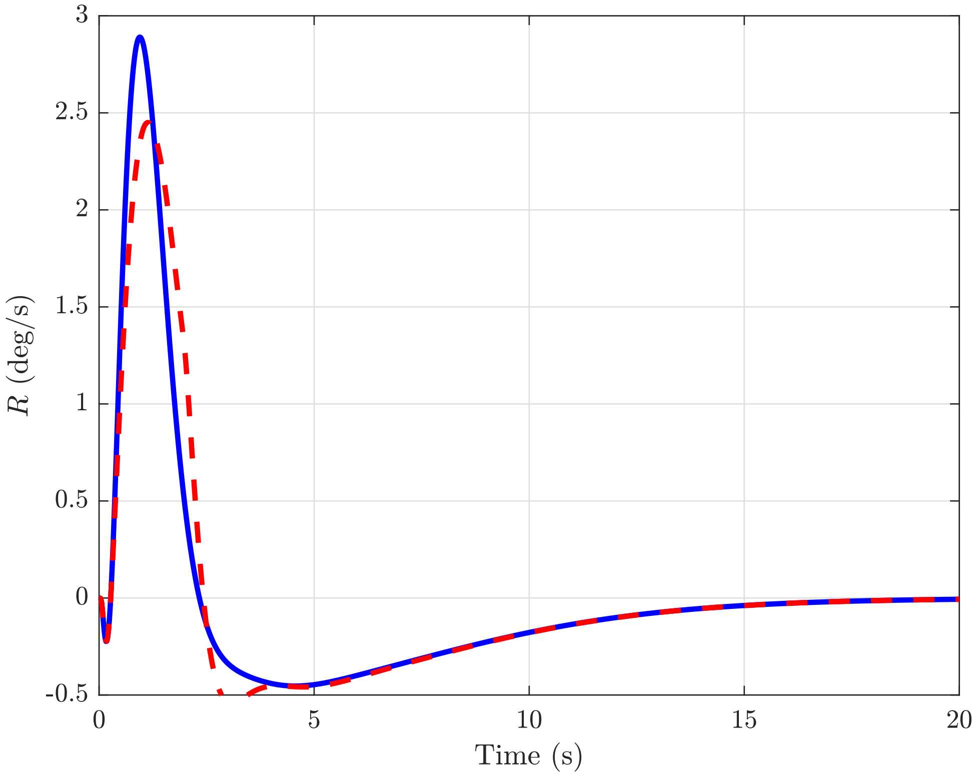


(d) Effective Yaw

Figure 5.13: (Continued)



(E) Differential Braking on Main Wheels



(F) Yaw Rate

FIGURE 5.13: (Continued)

of differential braking respectively. All three constraints become active with these limits, and the results are shown in Figure 5.13. The H_∞ controller violated the maximum limit of the actuators, but the MPC handled the constraints properly. When the constraints violation region has passed, the MPC behaves exactly like the H_∞.

5.5 Summary

This chapter establishes a non-linear mathematical model of a UAV during the taxi phase. The model is linearised analytically and shown that its lateral dynamics depend on the forward speed. Then the LPV model is constructed by a set of linear models covering the taxi envelope. It is shown that the constructed LPV model is in high fidelity with the established non-linear model. Simulation results show that the error between both models is less than 1 %. The H_∞ based LPV-MPC is proposed. The model is robustly stabilized by the normalized coprime factorization method to design controller K_s and observer H_s. Then, the LQR-based inverse optimal problem is formulated and solved for a set of plants that cover the whole taxi envelope to design the parameter-varying cost function matrices, which are further utilized to realize LPV-MPC. The designed MPC inherits the small-signal properties (stability margin and closed-loop performance) of the H_∞ LSDP when constraints are inactive (e.g., the perturbation around the equilibrium point that system remains within limits). The simulation results show that the designed MPC is in high fidelity with the H_∞ controller. Moreover, the quantitative comparison of both techniques is done by taking the normalized root mean square error (NRMSE) between the outputs. The NRMSEs between the outputs are 1, 1, and 0.9999 (1 for 100% match and 0 for no match) for cross distance (p_y), yaw (ψ), and yaw rate (R) respectively.

Chapter 6

Conclusion and Future Work

This chapter concludes the research work presented in this dissertation, along with some future avenues that can be explored for further study.

6.1 Conclusions

The challenges involved in the landing of fixed-wing UAVs have been highlighted in this work. The landing problem is divided into phases, and the contribution of each phase is concluded in the subsequent sections.

6.1.1 Airborne Phase

A different approach for landing fixed-wing UAVs is presented in this work. Glide and flare reference trajectories are predefined, and the UAV is forced to follow the altitude of the reference trajectory. The UAV encountered a wind disturbance during the landing. The vortex-ring model is used to simulate the moderate and severe wind disturbances in the system. Various scenarios are simulated and studied. The results demonstrate the effectiveness and correctness of the proposed technique under both wind shear effects. A qualitative analysis is also performed to compare the results with the benchmark work. The landing results demonstrated a significant improvement.

6.1.2 Ground Taxi Phase

The on-ground dynamics and control are complex due to the coupling between aerodynamics and tire-ground friction forces. Moreover, the lateral control during the deceleration phase becomes more challenging due to the high initial longitudinal velocity at touch-down and uncertainty in the tire-ground friction model. These effects are significantly amplified in high wind or gust conditions. This work establishes a non-linear mathematical model of a UAV during the taxi phase. The model is linearised analytically and shown that its lateral dynamics depend on the forward speed. Then the LPV model is constructed by a set of linear models covering the taxi envelope. It is shown that the constructed LPV model is in high fidelity with the established non-linear model. Simulation results show that the error between both models is less than 1 %.

The H_∞ based LPV-MPC is proposed for the lateral control of the UAV. The model is robustly stabilized by the normalized coprime factorization method to design the controller and observer gain matrices. Then, an LQR-based inverse optimisation problem is formulated and solved to design a set of cost function matrices (P, Q and R) for plants that cover the whole taxi envelope. This set of matrices used to design the parameter varying cost function matrices LPV-MPC. The proposed MPC inherits the small-signal properties (stability margin and closed-loop performance) of the H_∞ LSDP when constraints are inactive (i.e the perturbation around the equilibrium point that system remains within limits). The results show that the designed MPC is in high fidelity with the H_∞ controller. Moreover, the quantitative comparison of both techniques is done by taking the NRMSE between the outputs. The NRMSEs between the outputs are 1, 1, and 0.9999 (1 for 100% matching and 0 for no matching) for cross distance (p_y), yaw (ψ), and yaw rate (R) respectively.

6.2 Future Work

The potential research avenues in landing UAVs based on the current work that can effectively contribute are as follows.

6.2.1 Airborne Phase

A challenging task in the autonomous landing of fixed wind UAVs is to maintain baseline control performances in the presence of wind shear. An unknown and transient wind disturbance in the system can cause significant deviation in the UAV's position. One can estimate the wind by using onboard wind measurement sensors. In the aerospace industry, LiDAR's (Light Detection and Ranging) remote sensing capabilities have been used to detect wind disturbances. This wind preview can be used in a feed-forward controller to compensate for the effects of wind disturbance more efficiently.

6.2.2 Ground Taxi Phase

There are several parameters that can vary during ground taxi control of UAVs. Some of the important parameters are:

- Ground surface: The type and condition of the ground surface can affect the movement of the UAV during taxiing. Different surfaces such as concrete, grass, or gravel can affect the UAV's speed, stability, and handling.
- Wind speed and direction: Wind conditions can significantly affect the UAV's movement during ground taxi control. Strong winds can cause the UAV to drift or lose control, especially during takeoff and landing.
- Weight and balance: The weight and balance of the UAV can affect its handling characteristics during taxiing. If the UAV is not balanced correctly, it may be difficult to control or may tip over.

- Temperature and humidity: Temperature and humidity can affect the UAV's battery performance, motor performance, and other systems, which can affect its handling and control during ground taxi.
- Obstacles and terrain: The presence of obstacles such as trees, buildings, or other UAVs can affect the UAV's movement during ground taxi. The terrain can also affect the UAV's handling and stability, especially if it is uneven or sloped.

To ensure safe and efficient ground taxi control of UAVs, one can explore the taxi problem by considering and adjusting these parameters accordingly.

Bibliography

[1] J. B. Jeangene Vilmer, "Not so remote drone warfare," *International Politics*, vol. 1, no. 2, pp. 1–22, 2021.

[2] M. Stewart, S. Martin, and N. Barrera, "Unmanned aerial vehicles: fundamentals, components, mechanics, and regulations," *Unmanned Aerial Vehicles*, vol. 3, no. 1, pp. 1–70, 2021.

[3] A. K. Cronin, "US counter-terrorism: Moving beyond global counterinsurgency to strongpoint defence," *Survival*, vol. 63, no. 5, pp. 97–120, 2021.

[4] R. Hossain, "A short review of the drone technology," *International Journal of Mechatronics and Manufacturing Technology*, vol. 7, no. 2, pp. 53–68, 2022.

[5] M. Ghamari, P. Rangel, M. Mehrubeoglu, G. Tewolde, and R. S. Sherratt, "Unmanned aerial vehicle communications for civil applications: a review," *IEEE Access*, vol. 3, no. 2, pp. 1–8, 2022.

[6] S. Haugse, "Investigation of UAV related incidents and accidents," Master's thesis, UiT The Arctic University of Norway, 2022.

[7] G. Singhal, B. Bansod, and L. Mathew, "Unmanned aerial vehicle classification, applications and challenges: A review," vol. 1, no. 2, pp. 18–28, 2018.

[8] C. Lee, S. Kim, and B. Chu, "A survey: flight mechanism and mechanical structure of the UAV," *International Journal of Precision Engineering and Manufacturing*, vol. 22, no. 4, pp. 719–743, 2021.

[9] A. Gardi, R. Sabatini, and S. Ramasamy, “Multi-objective optimisation of aircraft flight trajectories in the ATM and avionics context,” *Progress in Aerospace Sciences*, vol. 83, no. 1, pp. 1–36, 2016.

[10] Crash, “Commercial aviation accidents statistics,” https://www.1001crash.com/index-page-statistique-lg-2-numpage-3.html, 2013, [Online; accessed July-05-2023].

[11] Z. Latif, A. Shahzad, A. I. Bhatti, J. F. Whidborne, and R. Samar, “Autonomous landing of an UAV using H_∞ based model predictive control,” *Drones*, vol. 6, no. 12, p. 416, 2022.

[12] Z. Latif, A. Shahzad, R. Samar, and A. I. Bhatti, “Lateral parameter-varying modelling and control of a UAV on-ground,” *4th IFAC Workshop on Linear Parameter Varying Systems*, vol. 54, no. 8, pp. 130–135, 2021.

[13] H. Mohandas and T. K. Weng, “Human factors analysis for aviation accidents and incidents in singapore,” in *International Conference on Human-Computer Interaction*, vol. 3, no. 1. Springer, 2021, pp. 308–323.

[14] Z. Zhen, S. Jiang, and K. Ma, “Automatic carrier landing control for unmanned aerial vehicles based on preview control and particle filtering,” *Aerospace Science and Technology*, vol. 81, no. 1, pp. 99–107, 2018.

[15] N. Sedlmair, J. Theis, and F. Thielecke, “Flight testing automatic landing control for unmanned aircraft including curved approaches,” *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 4, pp. 726–739, 2022.

[16] L. Wang, X. Jiang, Z. Zhang, and Z. Wen, “Lateral automatic landing guidance law based on risk-state model predictive control,” *ISA Transactions*, vol. 128, no. 1, pp. 611–623, 2022.

[17] L. Yang, C. Wang, and L. Wang, “Autonomous UAVs landing site selection from point cloud in unknown environments,” *ISA Transactions*, vol. 130, no. 1, pp. 610–628, 2022.

[18] Y. Yuan, H. Duan, and Z. Zeng, "Automatic carrier landing control with external disturbance and input constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 3, no. 1, pp. 100–108, 2022.

[19] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems.* John Wiley & Sons, 2015.

[20] S. Mathisen, K. Gryte, S. Gros, and T. A. Johansen, "Precision deep-stall landing of fixed-wing UAVs using nonlinear model predictive control," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 1, pp. 1–15, 2021.

[21] W. Qu, H. Zhang, and Y. Dong, "Optimization of UAVs landing longitudinal control under wind disturbance," in *IOP Conference Series: Earth and Environmental Science*, vol. 693, no. 1. IOP Publishing, 2021, p. 012106.

[22] R. Lungu and M. Lungu, "Application of $\mathrm{H}_2\,/\,\mathrm{H}_\infty$ and dynamic inversion techniques to aircraft landing control," *Aerospace Science and Technology*, vol. 46, pp. 146–158, 2015.

[23] R. Lungu, M. Lungu, and L. T. Grigorie, "Automatic control of aircraft in longitudinal plane during landing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1338–1350, 2013.

[24] D. V. Rao and T. H. Go, "Automatic landing system design using sliding mode control," *Aerospace Science and Technology*, vol. 32, no. 1, pp. 180–187, 2014.

[25] D. Zhang and X. Wang, "Autonomous landing control of fixed-wing UAVs: from theory to field experiment," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2-4, pp. 619–634, 2017.

[26] S. A. Adibi, S. Forer, J. Fries, and L. Yliniemi, "Autonomous unmanned aerial vehicle landing in windy conditions with map-elites," *The Knowledge Engineering Review*, vol. 32, no. 1, pp. 100–120, 2017.

[27] S. Kurnaz and O. Çetin, "Autonomous navigation and landing tasks for fixed wing small unmanned aerial vehicles," *Acta Polytechnica Hungarica*, vol. 7, no. 1, pp. 87–102, 2010.

[28] J.-G. Juang and S.-T. Yu, "Disturbance encountered landing system design based on sliding mode control with evolutionary computation and cerebellar model articulation controller," *Applied Mathematical Modelling*, vol. 39, no. 19, pp. 5862–5881, 2015.

[29] J. G. Juang, C. J. Cheng, and T. C. Yang, "Wind shear encountered landing control based on CMACs," in *Applied Mechanics and Materials*, vol. 284, no. 1. Trans Tech Publ, 2013, pp. 2351–2355.

[30] B. Huang, B. Lu, Q. Li, and Y. Tong, "Average dwell time based smooth switching linear parameter-varying proportional-integral-derivative control for an F$-$16 aircraft," *IEEE Access*, vol. 9, no. 1, pp. 30 979–30 992, 2021.

[31] Z. Jiao, H. Zhang, Y. Shang, X. Liu, and S. Wu, "A power-by-wire aircraft brake system based on high-speed on-off valves," *Aerospace Science and Technology*, vol. 106, no. 1, pp. 106–117, 2020.

[32] X. Li, X. Wei, Y. Wang, and Q. Yin, "Nonlinear tracking deviation correction control of airplanes in high-speed landing taxiing," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 12, pp. 2413–2424, 2020.

[33] Y. Dai, J. Song, L. Yu, Z. Lu, S. Zheng, and F. Li, "The lateral control during aircraft-on-ground deceleration phases," *Aerospace Science and Technology*, vol. 95, p. 105482, 2019.

[34] C. Roos, J.-M. Biannic, S. Tarbouriech, C. Prieur, and M. Jeanneau, "On-ground aircraft control design using a parameter-varying anti-windup approach," *Aerospace Science and Technology*, vol. 14, no. 7, pp. 459–471, 2010.

[35] Y. Zhang and H. Duan, "A directional control system for UCAV automatic takeoff roll," *Aircraft Engineering and Aerospace Technology*, vol. 85, no. 1, pp. 48–61, 2013.

[36] Q. Yin, H. Nie, and X. Wei, “Dynamics and directional stability of high-speed unmanned aerial vehicle ground taxiing process,” *Journal of Aircraft*, vol. 57, no. 4, pp. 689–701, 2020.

[37] J.-C. Roosa and I. K. Peddleb, “Autonomous take-off and landing of a low cost unmanned aerial vehicle,” *Journal of the South African Institution of Mechanical Engineering*, vol. 18, no. 1, pp. 59–71, 2009.

[38] M. Essuri, K. Alkurmaji, and A. Ghmmam, “Developing a dynamic model for unmanned aerial vehicle motion on ground during takeoff phase,” in *Applied Mechanics and Materials*, vol. 232, no. 1, 2012, pp. 561–567.

[39] Q. Yin, H. Nie, X. Wei, and K. Xu, “Design and analysis of high-speed unmanned aerial vehicle ground directional rectifying control system,” *International Journal of Aeronautical and Space Sciences*, vol. 18, no. 4, pp. 623–640, 2017.

[40] H. Song and X. Chen, “Model and auto take-off control law design of unmanned aerial vehicle,” in *Advanced Materials Research*, vol. 383, no. 1, 2012, pp. 1524–1530.

[41] J. Gao and H. Jia, “Control research for a small fixed-wing UAV during ground taxiing,” *Journal of Harbin Institute of Technology*, vol. 2, no. 1, pp. 8–13, 2017.

[42] D. Seiferth, R. Kuchar, and M. Heller, “Model-based design and real live on-runway testing of a ground controller for a novel diamond-shaped unmanned air vehicle,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC).* IEEE, 2017, pp. 3934–3941.

[43] S. Dong, Z. Jiao, X. Sun, and X. Liu, “Dynamic allocation algorithm for the gain of UAV nose wheel steering and differential braking based on decomposition control,” in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, vol. 4, no. 1. IEEE, 2016, pp. 831–835.

[44] F. Re, "Modelica landing gear modelling and on-ground trajectory tracking with sliding mode control," in *Advances in Aerospace Guidance, Navigation and Control*. Springer, 2011, vol. 2, no. 1, pp. 103–115.

[45] J. Duprez, F. Mora-Camino, and F. Villaume, "Control of the aircraft-on-ground lateral motion during low speed roll and manoeuvers," in *2004 IEEE Aerospace Conference Proceedings*, vol. 4, no. 1, 2004, pp. 2656–2666.

[46] C. Rowe and J. Maciejowski, "Tuning MPC using H_∞ loop shaping," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, vol. 2. IEEE, 2000, pp. 1332–1336.

[47] S. A. Bortoff, P. Schwerdtner, C. Danielson, and S. Di Cairano, "H_∞ loop-shaped model predictive control with heat pump application," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2386–2393.

[48] E. N. Hartley and J. M. Maciejowski, "Designing output-feedback predictive controllers by reverse-engineering existing LTI controllers," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2934–2939, 2013.

[49] S. Di Cairano and A. Bemporad, "Model predictive control tuning by controller matching," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 185–190, 2009.

[50] M. C. Priess, R. Conway, J. Choi, J. M. Popovich, and C. Radcliffe, "Solutions to the inverse lqr problem with application to biological systems analysis," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 770–777, 2014.

[51] G. Shah and S. Engell, "Tuning MPC for desired closed-loop performance for MIMO systems," *Proceedings of the 2011 American Control Conference*, pp. 4404–4409, 2011.

[52] Q. N. Tran, L. Özkan, and A. Backx, "Generalized predictive control tuning by controller matching," *Journal of Process Control*, vol. 25, pp. 1–18, 2015.

[53] M. Ryberg Wahlgreen, J. Bagterp Jørgensen, and M. Zanon, “Model predictive control tuning by monte carlo simulation and controller matching,” *arXiv e-prints*, pp. arXiv–2212, 2022.

[54] W. Lim, S. Lee, J. Yang, M. Sunwoo, Y. Na, and K. Jo, “Automatic weight determination in model predictive control for personalized car-following control,” *IEEE Access*, vol. 10, pp. 19 812–19 824, 2022.

[55] S. A. Giraldo, P. A. Melo, and A. R. Secchi, “Tuning of model predictive controllers based on hybrid optimization,” *Processes*, vol. 10, no. 2, p. 351, 2022.

[56] H. Cho, G. Bacelli, and R. G. Coe, “Model predictive control tuning by inverse matching for a wave energy converter,” *Energies*, vol. 12, no. 21, p. 4158, 2019.

[57] K. Tamkaya, L. Ucun, and I. Ustoglu, “H-infinity based model following method in autolanding systems,” *Aerospace Science and Technology*, vol. 94, p. 105379, 2019.

[58] P. D. Domański and M. Ławryńczuk, “Impact of MPC embedded performance index on control quality,” *IEEE Access*, vol. 9, pp. 24 787–24 795, 2021.

[59] B. E. Sedhom, M. M. El-Saadawi, A. Y. Hatata, and E.-H. E. Abd-Raboh, “H-infinity versus model predictive control methods for seamless transition between islanded-and grid-connected modes of microgrids,” *IET Renewable Power Generation*, vol. 14, no. 5, pp. 856–870, 2020.

[60] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[61] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[62] J. C. Doyle, “Guaranteed margins for LQG regulators,” *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 756–757, 1978.

[63] K. Glover and D. McFarlane, "Robust stabilization of normalized coprime factor plant descriptions with H_∞ bounded uncertainty," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 821–830, 1989.

[64] D. McFarlane and K. Glover, "A loop-shaping design procedure using H_∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 759–769, 1992.

[65] S. S. Nair, "Automatic weight selection algorithm for designing H_∞ controller for active magnetic bearing," *International Journal of Engineering Science and Technology*, vol. 3, no. 1, pp. 122–138, 2011.

[66] J. Sefton and K. Glover, "Pole/zero cancellations in the general H_∞ problem with reference to a two block design," *Systems & Control Letters*, vol. 14, no. 4, pp. 295–306, 1990.

[67] R. A. Hyde and K. Glover, "The application of scheduled H_∞ controllers to a VSTOL aircraft," *IEEE Transactions on Automatic Control*, vol. 38, no. 7, pp. 1021–1039, 1993.

[68] "Jos F. Sturm sedumi home page," https://sedumi.ie.lehigh.edu/, accessed: July-05-2023.

[69] "Johan Löfberg yalmip home page," http://yalmip.github.io/, accessed: July-05-2023.

[70] B. L. Stevens and F. L. Lewis, "Aircraft control and simulation," 1992.

[71] B. H. Wang, D. B. Wang, Z. A. Ali, B. Ting Ting, and H. Wang, "An overview of various kinds of wind effects on unmanned aerial vehicle," *Measurement and Control*, vol. 52, no. 7-8, pp. 731–739, 2019.

[72] A. A. Woodfield and J. F. Woods, "Worldwide experience of wind shear during 1981-1982," Royal Aircraft Establishment Bedford (England), Tech. Rep., 1983.

[73] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, "A condensed and sparse QP formulation for predictive control," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 5217–5222.

[74] Y. Du, Z. Chen, J. Hao, and Y. Guo, "Joint optimization of trajectory and communication in multi-UAV assisted backscatter communication networks," *IEEE Access*, vol. 10, pp. 40 861–40 871, 2022.

[75] L. Sultan, M. Anjum, M. Rehman, S. Murawwat, and H. Kosar, "Communication among heterogeneous unmanned aerial vehicles (uavs): Classification, trends, and analysis," *IEEE Access*, vol. 9, pp. 118 815–118 836, 2021.

[76] G. Cai, B. M. Chen, and T. H. Lee, "Coordinate systems and transformations," in *Unmanned rotorcraft systems*. Springer, 2011, vol. 9, pp. 23–34.

[77] M. Sidoryuk, M. Goman, S. Kendrick, D. Walker, and P. Perfect, "Nonlinear analysis and synthesis techniques for aircraft control," vol. 1, no. 1, pp. 15–36, 2001.

[78] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[79] C. Hoffmann and H. Werner, "Complexity of implementation and synthesis in linear parameter-varying control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 749–11 760, 2014.

[80] R. A. Lobo, J. M. Palma, C. F. Morais, L. d. P. Carvalho, M. E. Valle, and C. R. Oliveira, "A brief tutorial on quadratic stability of linear parameter-varying model for biomathematical systems," in *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. IEEE, 2019, pp. 1–6.

[81] M. Johansson and A. Rantzer, "Computation of piecewise quadratic lyapunov functions for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 555–559, 1998.

[82] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control," in *2012 IEEE 51st IEEE conference on Decision and Control (CDC).* IEEE, 2012, pp. 668–674.

[83] I. J. Lustig, R. E. Marsten, and D. F. Shanno, "On implementing mehrotra's predictor–corrector interior-point method for linear programming," *SIAM journal on Optimization*, vol. 2, no. 3, pp. 435–449, 1992.

[84] L. Li and K.-C. Toh, "An inexact interior point method for l 1-regularized sparse covariance selection," *Mathematical Programming Computation*, vol. 2, no. 3, pp. 291–315, 2010.

[85] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 267–278, 2009.

[86] R. A. Horn and F. Zhang, "Basic properties of the schur complement," in *The Schur Complement and Its Applications.* Springer, 2005, vol. 5, no. 1, pp. 17–46.